

Heuristic Planning for Decentralized MDPs with Sparse Interactions

Francisco S. Melo and Manuela Veloso

Abstract. In this work, we explore how local interactions can simplify the process of decision-making in multiagent systems, particularly in multirobot problems. We review a recent decision-theoretic model for multiagent systems, the decentralized sparse-interaction Markov decision process (Dec-SIMDP), that explicitly distinguishes the situations in which the agents in the team must coordinate from those in which they can act independently. We situate this class of problems within different multiagent models, such as MMDPs and transition independent Dec-MDPs. We then contribute a new general approach that leverages the particular structure of Dec-SIMDPs to efficiently plan in this class of problems, and propose two algorithms based on this underlying approach. We pinpoint the main properties of our approach through illustrative examples in multirobot navigation domains with partial observability, and provide empirical comparisons between our algorithms and other existing algorithms for this class of problems. We show that our approach allows the robots to look ahead for possible interactions, planning to accommodate such interactions and thus overcome some of the limitations of previous methods.

1 Introduction

Recent years have witnessed a profusion of work on multiagent models that capture some of the fundamental features of Dec-(PO)MDPs (such as partial observability) without incurring in the associated computational cost. In this paper, we contribute to this extensive literature, and investigate a recent model for cooperative multiagent decision-making in the presence of global partial observability [17]. This model is

Francisco S. Melo
INESC-ID/Instituto Superior Técnico, UTL, Av. Prof. Dr. Cavaco Silva,
2780-990 Porto Salvo, Portugal
e-mail: fmelo@inesc-id.pt

Manuela Veloso
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
e-mail: veloso@cs.cmu.edu

motivated by the observation that, in many real-world scenarios involving multiple decision makers (*e.g.*, robots), the tasks of the different agents/robots are not coupled at every decision-step but only in relatively infrequent situations. We dub such problems as having *sparse interaction*. *Multi-robot systems* provide our primary motivation and constitute natural examples for the class of problems considered herein. In multi-robot systems, the interaction among the different robots is naturally limited by each robot’s physical boundaries (workspace, communication range, etc.) and limited perception capabilities. Therefore, when dealing with multi-robot systems, one natural approach is to subdivide the overall task into smaller tasks that each robot can then execute autonomously or as part of a smaller group [5, 15, 18].

Several previous works have exploited simplified models of interaction in multi-agent settings. For example, a hierarchical learning algorithm can consider only the interaction between the different agents at a higher control level, while allowing the agents to learn lower level tasks independently [6]. Also, coordination graphs can represent compactly the dependencies between the actions of different agents, thus capturing the local interaction between them [8, 10]. Local interactions have also been exploited to minimize communication during policy execution [16] and in the game-theoretic literature to attain compact game representations [9, 20].

In this paper we consider Dec-MDPs with parse interactions (henceforth Dec-SIMDPs). Dec-SIMDPs have been proposed in [17] under the designation of *interaction-driven Markov games* and are closely related to *distributed POMDPs with coordination locales* [19] and *Dec-MDPs with event-driven interactions and complex rewards* [14]. Dec-SIMDPs leverage the independence between agents to decouple the decision process in significant portions of the joint state space, allowing the agents to base their decisions in their local perception of state and alleviating the difficulties arising from global partial observability. On those situations in which the agents interact, Dec-SIMDPs rely on communication to bring down the the computational complexity of the joint decision process. Dec-SIMDPs “balance” the independence assumptions with communication: in any given state, the agents are either independent or can communicate.¹

The contributions in this paper are two-fold. On one hand, we build on [17], providing a precise formalization of Dec-SIMDPs and discussing in some detail the relation with well-established decision-theoretic models such as Dec-MDPs, MMDPs and MDPs. On the other hand, we contribute two new algorithms that exhibit significant computational savings when compared to existing algorithms for Dec-SIMDPs. We illustrate the application of our algorithms in several simple navigation tasks.

2 Decision Theoretic Models

We start by reviewing *decentralized partially observable Markov decision processes* (Dec-POMDPs) and related decision theoretic models. A N -agent Dec-POMDP \mathcal{M}

¹ We note that both independence assumptions and communication can significantly bring down the computational complexity in Dec-(PO)MDP related models [1, 7].

is specified as a tuple $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), (\mathcal{Z}_k), P, (O_k), r, \gamma)$, where \mathcal{X} is the joint state-space, $\mathcal{A} = \times_{k=1}^N \mathcal{A}_k$ is the set of joint actions, with each \mathcal{A}_k the individual action set for agent k , each \mathcal{Z}_k represents the set of possible local observation for agent k , $P(x, a, y)$ represents the transition probabilities from joint state x to joint state y when the joint action a is taken, each $O_k(x, a, z_k)$ represents the probability of agent k making the local observation z_k when the joint state is x and the last joint action taken was a , and $r(x, a)$ represents the expected reward received by all agents for taking the joint action a in joint state x . The scalar γ is a discount factor.

A *N-agent Decentralized Markov decision process* (Dec-MDP) is a particular class of Dec-POMDP in which the state is *jointly fully observable*. Formally this can be translated into the following condition: for every joint observation $z \in \mathcal{Z}$, with $\mathcal{Z} = \times_{k=1}^N \mathcal{Z}_k$, there is a state $x \in \mathcal{X}$ such that $\mathbb{P}[X(t) = x \mid Z(t) = z] = 1$, where $X(t)$ is the joint state of the process at time t and $Z(t)$ the corresponding joint observation. Similarly, a *partially observable Markov decision process* (POMDP) is a 1-agent Dec-POMDP and a *Markov decision process* (MDP) is a 1-agent Dec-MDP. Finally, a *N-agent multiagent MDP* (MMDP) is a *N-agent Dec-MDP* that is *fully observable*, i.e., for every individual observation $z_k \in \mathcal{Z}_k$ there is a state $x \in \mathcal{X}$ such that $\mathbb{P}[X(t) = x \mid Z_k(t) = z_k] = 1$.

In the remainder of the paper we focus on Dec-MDPs, particularly in Dec-MDPs for which the state-space \mathcal{X} can be factorized as $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_N$. Although more general Dec-MDP models exist [3], we adhere to this simplified version, as this is sufficient for our purposes and makes the presentation both clearer and simpler. Indeed, since multirobot navigation scenarios constitute the main motivation behind our work, the sensible approach is, in fact, to consider a factored joint state-space, where each \mathcal{X}_k denotes the individual state-space for robot k . For future reference, let $\mathcal{X}_{-k} = \mathcal{X}_0 \times \dots \times \mathcal{X}_{k-1} \times \mathcal{X}_{k+1} \times \dots \times \mathcal{X}_N$ and denote by x_{-k} a general element of \mathcal{X}_{-k} . We also write $x = (x_{-k}, x_k)$ to denote the fact that the k th component of x takes the value x_k . We use a similar notation for actions.

In this partially observable multiagent setting, an individual (non-Markov) policy for agent k is a mapping $\pi_k : \mathcal{H}_k \rightarrow \Delta(\mathcal{A}_k)$, where $\Delta(\mathcal{A}_k)$ is the space of probability distributions over \mathcal{A}_k and \mathcal{H}_k is the set of all possible finite histories (finite sequences of actions and observations) for agent k .

In a Dec-MDP, the purpose of all agents is to determine a joint policy π so as to maximize the total sum of discounted rewards. In order to write this in terms of a function, we consider a distinguished initial state, $x^0 \in \mathcal{X}$, that is assumed common knowledge among all agents. The purpose of the agents is then to maximize

$$V^\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(X(t), A(t)) \mid X(0) = x^0 \right].$$

Transition-independent Dec-MDPs [2] constitute a particular subclass of Dec-MDPs in which, for all $(x, a) \in \mathcal{X} \times \mathcal{A}$,

$$\mathbb{P}[X_k(t+1) = y_k \mid X(t) = x, A(t) = a] = \mathbb{P}[X_k(t+1) = y_k \mid X_k(t) = x_k, A_k(t) = a_k]. \quad (1)$$

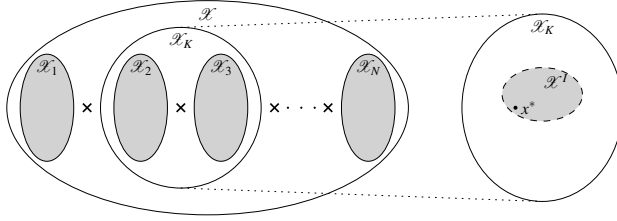


Fig. 1 Diagram representing the relation between individual state-spaces, \mathcal{X}_k , the joint state-space \mathcal{X} , and the set \mathcal{X}_K for a set of agents $K = \{2, 3\}$. We also represent an interaction area \mathcal{X}^I associated with an interaction state $x^* \in \mathcal{X}_K$ (see main text).

The transition probabilities can thus be factorized as

$$P(x, a, y) = \prod_{k=1}^N P_k(x_k, a_k, y_k), \quad (2)$$

where $P_k(x_k, a_k, y_k)$ represents the transition probabilities from local state x_k to local state y_k when the individual action a_k was taken. This particular class of Dec-MDPs has been shown to be NP-complete in finite-horizon settings, versus the NEXP-completeness of general Dec-MDPs [7, 2].

Similarly, *reward independent Dec-MDPs* correspond to a subclass of Dec-MDPs in which, for all $(x, a) \in \mathcal{X} \times \mathcal{A}$, $r(x, a) = f(r_k(x_k, a_k), k = 1, \dots, N)$, i.e., the global reward function r can be obtained from local reward functions $r_k, k = 1, \dots, N$, and each individual reward is consistent with the global reward [7]. One typical example is

$$r(x, a) = \sum_{k=1}^N r_k(x_k, a_k). \quad (3)$$

Interestingly, it was recently shown that reward independent Dec-MDPs retain NEXP-complete complexity [1]. However, when associated with transition independence, reward independence implies that a Dec-MDP can be decomposed into N independent MDPs, each of which can be solved separately. The complexity of this class of problems thus reduces to that of standard MDPs (P-complete). For a summary of complexity results for Dec-POMDP related models, we refer to [1, 7].

3 Local Interactions in Dec-MDPs

In this paper we exploit sparse interactions among the different agents in a Dec-MDP. In particular, we are interested in Dec-MDPs in which there is some level of both transition and reward dependency, but this dependency is limited to specific regions of the state space. We introduce decentralized sparse-interaction MDPs

² In this paper we are interested in infinite horizon problems. Complexity results for infinite-horizon problems with partial observability are even more discouraging—even single-agent POMDPs have been shown undecidable in infinite-horizon settings [12].

(Dec-SIMDPs). Dec-SIMDPs essentially correspond to the model previously proposed in [17] under the designation of *interaction-driven Markov games*. However, we revisit several aspects of this model that were not properly formalized in the original work, and provide a more extensive discussion on the relation between this work and the models surveyed in the previous section. We postpone to the following section the introduction of two novel algorithms for this class of problems.

We start by introducing some auxiliary notation. Given an N -agent Dec-MDP $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), P, r, \gamma)$, let K be a subset of the N agents in \mathcal{M} . Extending the notation in Section 2, we denote by $\mathcal{X}_K = \times_{k \in K} \mathcal{X}_k$ the joint state-space of all agents in K . Similarly, we write \mathcal{X}_{-K} to denote the joint state-space of the agents *not* in K . We write x_K to denote a general element of \mathcal{X}_K and x_{-K} to denote a general element of \mathcal{X}_{-K} . We write $x = (x_{-K}, x_K)$ to distinguish the components of x corresponding to agents in K and those corresponding to agents not in K (see Fig. 1 for an illustration).

Also, for any given a Dec-MDP, we write the reward $r(x, a)$ as

$$r(x, a) = \sum_{k=1}^N r_k(x_k, a_k) + \sum_{i=1}^M r_i^I(x_{K_i}, a_{K_i}), \quad (4)$$

where each r_k corresponds to an individual component of the reward function that depends only on agent k and there are M agent sets, $K_i, i = 1, \dots, M$, and M reward components, r_i^I (the interaction components), each depending on all the agents in K_i and only on these. We note that this decomposition can be performed at no loss of generality, since any reward r can be trivially written in that form by setting $M = 1$, $r_k \equiv 0$, $K_1 = \{1, \dots, N\}$, and $r_1^I = r$. The scenarios that we are interested in are those in which the support of $\sum_{i=1}^M r_i^I$ – the subset of $\mathcal{X} \times \mathcal{A}$ in which this sum is non-zero – is small when compared with $\mathcal{X} \times \mathcal{A}$.

We say that an agent k_0 in a Dec-MDP is *independent* of an agent k_1 in a state $x \in \mathcal{X}$ if the transition probabilities for the individual state of agent k_0 at x do not depend on the state/action of agent k_1 , *i.e.*,

$$\mathbb{P}[X_{k_0}(t+1) = y_{k_0} \mid X(t) = x, A(t) = a] = \mathbb{P}[X_{k_0}(t+1) = y_{k_0} \mid X_{-k_1}(t) = x_{-k_1}, A_{-k_1}(t) = a_{-k_1}].$$

and it is possible to decompose the global reward function $r(x, a)$ as in (4) in such a way that no set K_i contains both k_0 and k_1 . When any of the above does not hold, we say that agent k_0 *depends* on k_1 at state x . This notion of dependence extends trivially to sets of agents by interpreting the agents in each set as a single centralized agent. Intuitively, two agents are dependent if either the rewards or the transitions of one of the agents depend on the state or action of the other.

The agents in a set K *interact* at state $x \in \mathcal{X}$ if the following conditions hold:

- If $k_0 \in K$ and agent k_0 depends on agent k_1 in state x , then $k_1 \in K$.
- If $k_1 \in K$ and there is an agent k_0 that depends on agent k_1 in state x , then $k_0 \in K$.
- There is no strict subset $K' \subset K$ such that the above conditions hold for K' .

If the agents in a set K interact in a state x , then we refer to x_K as an *interaction state* for the agents in K . Interactions capture all dependencies between the agents in K : if the agents in K interact in state x_K , no agent in K is independent of all others in x_K and no agent outside K depends on any agent in K .

In a general Dec-MDP, all agents interact in all states, since generally there are no transition or reward independences. On the other hand, transition and reward independent Dec-MDPs have no interactions at all – as expected, such problems can be decomposed into N independent single-agent models and solved in a straightforward manner. An interaction occurs whenever a group of agents is coupled in terms of either transitions or rewards and either the transition probabilities cannot be factorized as in (2) or the reward function cannot be decomposed as in (3).

In a general N -agent Dec-MDP, we define an *interaction area* \mathcal{X}^I as follows:

- $\mathcal{X}^I \subset \mathcal{X}_K$ for some set of agents K ;
- $\exists x^* \in \mathcal{X}^I$ such that x^* is an interaction state for the agents in K ;
- The set \mathcal{X}^I is connected³

An agent k is involved in an interaction at time t if there is one interaction area \mathcal{X}^I involving a set of agents K such that $k \in K$ and $X(t) = (x_K, x_{-K})$ with $x_K \in \mathcal{X}^I$. We represent the concept of interaction area in the diagram of Fig. 1.

The purpose of defining/identifying the interaction areas in a Dec-MDP is to single out situations in which the actions of one agent depend on other agents. An agent that is not involved in any interaction should be able to choose its individual actions independently of the other agents and thus be unaffected by partial (global) state observability. In contrast, we focus on those problems for which each of the agents involved in an interaction in a particular interaction area $\mathcal{X}^I \subset \mathcal{X}_K$ at time t has full access to the state $X_K(t)$. We refer to such a Dec-MDP as having *observable interactions*. Our focus on Dec-MDPs with observable interactions, although apparently restrictive, actually translates a property often observed in real-world scenarios. For example, when interacting, robots are often able to observe/communicate relevant information for coordination. In a sense, interaction areas encapsulate the need for information sharing in a general multiagent decision problem.

We are now in position to introduce our model. A N -agent Dec-MDP \mathcal{M} has *sparse interactions* if all agents are independent except in a set of M interaction areas, $\{\mathcal{X}_1^I, \dots, \mathcal{X}_M^I\}$, with $\mathcal{X}_i^I \subset \mathcal{X}_{K_i}$ for some set of agents K_i , and such that $|\mathcal{X}_i^I| \ll |\mathcal{X}_{K_i}|$. We refer to a Dec-MDP with sparse observable interactions as a Dec-SIMDP (decentralized sparse-interaction MDP). For all agents outside interaction areas, the joint transition probabilities and reward function for a Dec-SIMDP can be factorized as in (2) and (3), and it is possible to model these agents using “individual MDPs”. On the other hand, the agents involved in an interaction can be modeled using a “local” MMDP. We represent such a Dec-SIMDP as a tuple

$$\Gamma = \{\mathcal{M}_k, k = 1, \dots, N\}, \{(\mathcal{X}_i^I, \mathcal{M}_i^I), i = 1, \dots, M\},$$

where

- Each \mathcal{M}_k is an MDP $\mathcal{M}_k = (\mathcal{X}_k, \mathcal{A}_k, P_k, r_k, \gamma)$ that individually models agent k in the absense of other agents, where r_k is the component of the joint reward function associated with agent k in the decomposition in (3);

³ In this context we say that a set $U \subset \mathcal{X}$ is *connected* if, for any pair of states $x, y \in U$, there is a sequence of actions that, with positive probability, yields a trajectory $\{x(0), \dots, x(T)\}$ such that $x(t) \in U, t = 0, \dots, T$, and either $x(0) = x$ and $x(T) = y$ or vice-versa.

- Each \mathcal{M}_i^I is an MMDP that captures a *local interaction* between K_i agents in the states in \mathcal{X}_i^I and is given by $\mathcal{M}_i^I = (K_i, \mathcal{X}_{K_i}, (\mathcal{A}_k), \mathbf{P}_i^I, \mathbf{r}_i^I, \gamma)$, with $\mathcal{X}_i^I \subset \mathcal{X}_{K_i}$.

Each MMDP \mathcal{M}_i describes the interaction between a subset K_i of the N agents, and the corresponding state-space, \mathcal{X}_{K_i} , is a superset of the respective interaction area.

A Dec-SIMDP is an alternative way of representing a Dec-MDP with observable interactions. In the states of each interaction area in a Dec-SIMDP (and only in these), the agents involved in the associated MMDP are able to observe their joint state. This can be interpreted as having the agents in this area use communication to overcome local state perception and decide jointly on their action. Outside these areas, the agents have only a local perception of the state and should, therefore, choose the actions independently of the other agents.

Note that, in the absence of any interaction areas, the Dec-SIMDP reduces to a set of independent MDPs that can be solved separately. This captures the situation in which the agents are completely independent. On the other hand, a Dec-SIMDP is a Dec-MDP model with joint state observability in the interaction areas. In those situations in which all agents interact in all states, as assumed in the general Dec-MDP model, the whole state-space is an interaction area and, as such, our assumption of full state observability in the interaction areas renders our model equivalent to an MMDP. Nevertheless, the appeal of the Dec-SIMDP model is that many practical situations do not fall in either of the two extreme cases (*i.e.*, independent MDPs vs. fully observable MMDP). It is in these situations that the Dec-SIMDP model may bring an advantage over more general (but potentially intractable) models.

4 Planning in Dec-SIMDPs

We now introduce two novel Dec-SIMDP algorithms that leverage the particular structure of this class of problems and avoid the computational complexity of more general Dec-MDP models. Our approach relies on a simple heuristic that provides interesting insights into the structure of Dec-SIMDP and on how should the interaction areas be chosen for a particular problem. As in most planning problems, the underlying Dec-MDP/Dec-SIMDP model is assumed known.

4.1 MPSI and LAPSI

Let us start by considering a Dec-SIMDP in which all except agent k have full state observability. Let us further suppose that the agents with full state observability follow some fixed known policy π_{-k} . Then, from the perspective of agent k , the environment behaves as a POMDP, since the other agents can be collectively regarded as part of the environment. In this particular situation, we can use any POMDP solution method to compute the policy for agent k .

Our heuristic departs from the simplified situation just described. For each agent $k = 1, \dots, N$, we assume all other agents to follow some (hypothesized) policy $\hat{\pi}_{-k}$

Algorithm 1. General outline of the proposed heuristic planning algorithms.

Require: Dec-SIMDP model $\mathcal{M} = (\{\mathcal{M}_k, k = 1, \dots, N\}, \{(\mathcal{X}_i^I, \mathcal{M}_i^I), i = 1, \dots, M\})$

- 1: **for all** $k = 1, \dots, N$ **do**
- 2: Build hypothetical policy $\hat{\pi}_{-k}$ for other agents
- 3: From \mathcal{M} and $\hat{\pi}_{-k}$ build POMDP model for agent k , $(\mathcal{X}, \mathcal{A}_k, \mathcal{Z}_k, P_{\hat{\pi}_{-k}}, r_{\hat{\pi}_{-k}}, \gamma)$
- 4: Use preferred POMDP solution technique to compute $\pi_k : \Delta(\mathcal{X}) \rightarrow \mathcal{A}_k$
- 5: **end for**

that depends only on the state X_t . Given this policy $\hat{\pi}_{-k}$, we derive the POMDP model for agent k and use the corresponding solution as the policy π_k . Algorithm 1 summarizes this approach.

This heuristic rests on the assumption that the hypothesized policy, $\hat{\pi}_{-k}$, will allow agent k to approximately “track” the other agents and hence choose its actions accordingly. The closer $\hat{\pi}_{-k}$ is to the actual policy of the other agents, the better agent k will be able to track them, and the better he will decide.

The two algorithms proposed in this paper, dubbed MPSI (Myopic Planning for Sparse Interactions) and LAPSI (Look-Ahead Planning for Sparse Interactions), share this underlying structure but consider different hypothetical policies $\hat{\pi}_{-k}$ in Step 2. In MPSI, agent k models the other agents as self-centered and oblivious to the interactions. In other words, agent k acts as if each agent j , $j \neq k$, is following the optimal policy for the corresponding MDP \mathcal{M}_j in the Dec-SIMDP model. In environments with almost no interaction, MPSI actually provides a good approximation to the policy of the other agents outside the interaction areas.

In contrast, in LAPSI, agent k considers that all other agents jointly adopt the optimal policy for the underlying MMDP. LAPSI is, in a sense, the counterpart to MPSI, as it provides a good approximation to the policy of the other agents in scenarios where the interactions are not so sparse.

Clearly, the idea in Algorithm 1 can be used in general Dec-POMDPs. However, the hypothetical policy $\hat{\pi}_{-k}$ will seldom correspond to the actual policy followed by the other agents, and it is only natural that this method will not allow each agent k to properly “track” the other agents and decide accordingly, this leading to poor results in general Dec-POMDPs. The particular structure of Dec-SIMDPs, however, renders this approach more appealing for two reasons: on one hand, outside interaction areas the policy of agent k (ideally) exhibits minimum dependence on the state/policy of the other agents. As such, poor tracking in these areas has little impact on the policy of agent k . In interaction areas, on the other hand, local full observability allows agent k to perfectly track the other agents involved in the interaction and choose its actions accordingly.

In the following subsection, we describe a specific instance of both MPSI and LAPSI that is closely related to the Q_{MDP} heuristic for POMDPs [11] and rests on the concept of *generalized α -vectors*. As will soon become apparent, even using such a simple POMDP solver such as Q_{MDP} , LAPSI is able to attain near-optimal performance in all test scenarios while incurring in a computational cost much lower than alternative methods.

4.2 Generalized α -Vectors

We now propose particular instances of both MPSI and LAPSIs that are closely related with the Q -MDP heuristic for POMDPs [11], although exploiting the structure of the Dec-SIMDPs model.

To this purpose, we note that each agent k in a Dec-SIMDP has *full local state observability*, implying that, at each time-step t , the k th component of the state, $X_k(t)$, is always unambiguously determined. Furthermore, given our assumption of observable interactions, at each time step t only those state-components corresponding to agents *not interacting* with agent k will be unobservable. By definition, these state-components do not depend on the state/action of agent k at time t , and instead depend only on $\hat{\pi}_{-k}$. We take advantage of this fact and modify the Q -MDP heuristic as our POMDP solution method.⁴ To this purpose we introduce the concept of *generalized α -vectors* for Dec-SIMDPs. Due to space limitations, we omit some of the details involved in the derivation of these vectors as well as the analysis of its properties. Instead, we refer to [13] for further details.

Let us denote by \mathcal{X}_I the set of all (joint) states in interaction areas, and define a *generalized α -vector* for agent k , α_k , recursively as follows:

$$\alpha_k(x) = r_{\pi_{-k}}(x, a_k) + \gamma \sum_{y \in \mathcal{X}_I} P_{\pi_{-k}}(x, a_k, y) \max_{u_k} \alpha_k(y, u_k) + \gamma \max_{u_k} \sum_{y \notin \mathcal{X}_I} P_{\pi_{-k}}(x, a_k, y) \alpha_k(y, u_k), \quad (5)$$

where

$$\begin{aligned} r_{\pi_{-k}}(x, a_k) &= \sum_{a_{-k}} \pi_{-k}(x_{-k}, a_{-k}) r(x, (a_{-k}, a_k)) \\ P_{\pi_{-k}}(x, a_k, y) &= \sum_{a_{-k}} \pi_{-k}(x_{-k}, a_{-k}) P(x, (a_{-k}, a_k), y). \end{aligned}$$

The generalized α -vector α_k is the fixed-point of the expression (5) and are well-defined and unique. Furthermore, they can be computed iteratively using a dynamic-programming-like approach that, essentially, iterates through the recursion in (5). It is also possible to show that α_k corresponds to the optimal Q -function of an associated MDP whose dimension grows linearly with the dimension of the original Dec-SIMDP. Recalling that the decision process for agent k can be modeled using a standard POMDP, we adopt the approximation

$$Q^*(x_k, \mathbf{b}_{-k}, a_k) \approx \sum_{x_{-k}} \mathbf{b}_{x_{-k}} \alpha_k(x, a_k). \quad (6)$$

This solution can now be used to choose the actions of agent k by maximizing the above expression.

⁴ In the continuation, and to avoid unnecessarily complicating the presentation, we focus on a 2-agent scenario. The development presented extends trivially to more than two agents at the cost of more cumbersome expressions.

5 Results

In this section we describe the results obtained from applying both MPSI and LAPSI to a range of problems of different dimensions, and analyze the performance of our methods in each of the test scenarios. We compare the performance of both MPSI and LAPSI to that of the optimal fully observable MMDP policy and that of the IDMG algorithm from [17]. In the IDMG algorithm, each agent k in a Dec-SIMDP $(\{\mathcal{M}_k, k = 1, \dots, N\}, \{(\mathcal{X}_i^I, \mathcal{M}_i^I), i = 1, \dots, M\})$ follows the optimal individual policy π_k for the MDP \mathcal{M}_k outside the interaction areas. In the interaction areas, the agents engage in a sequence of local matrix games in which they jointly adopt the equilibrium policy.

We used several robot navigation scenarios to test our algorithms (see Fig. 2), since the Dec-SIMDP model is particularly appealing for modeling multi-robot problems. Furthermore, in this class of problems, the results can be easily visualized and interpreted. In each of the test scenarios, each robot in a set of two/four robots must reach one specific state. In the smaller environments (Maps 1 through 4), the goal state is marked with a number, corresponding to the number of the robot. The cells with a boxed number correspond to the initial states for the robots. In the larger environments, the goal for each robot is marked with a cross, \times , and the robots each depart from the other robot's goal state, in an attempt to increase the possibility of interaction. Each robot has 4 possible actions that move the robot in the

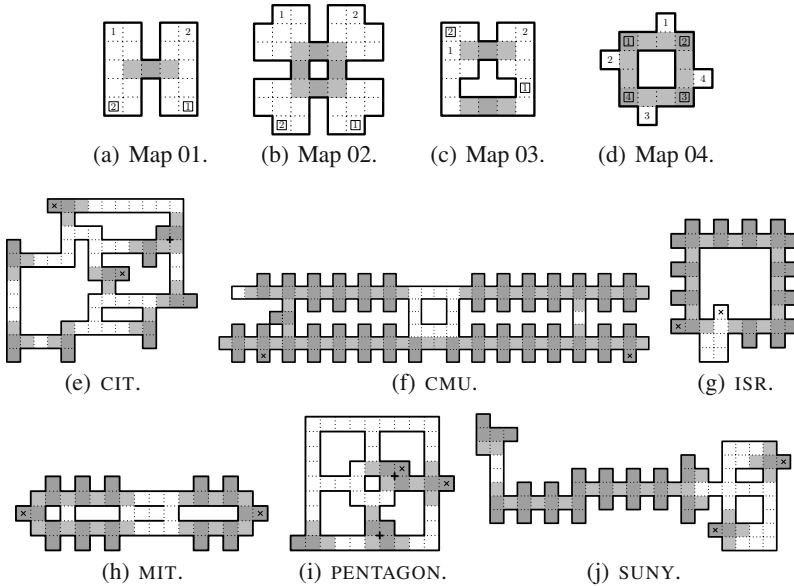


Fig. 2 Environments used in the experiments. The dark gray areas correspond to interaction states and the light gray areas to the corresponding interaction areas. We refer to the main text for details.

corresponding direction with probability 0.8 and fail with probability 0.2. The shaded regions correspond to interaction areas, inside of which the darker cells correspond to interaction states, in which the robots get a penalty of -20 if they stand in the same cell simultaneously. Also, in these interaction states, the rate of action failure is increased to 0.4.⁵ Upon reaching the corresponding goal, each agent receives a reward of $+1$ and its position is reset to the initial state. The dimension of the state-space for the different Dec-MDPs is summarized in Table 1.

Table 1 Total discounted reward for each of the four different algorithms in each of the test-scenarios. The results are averaged over 1,000 independent Monte-Carlo runs. Entries in **bold** correspond to guaranteed optimal performance. Entries in *italic* in the same line are not statistically different.

Environment	# States	IDMG	MPSI	LAPSI	MMDP
		Disc. Rew.	Disc. Rew.	Disc. Rew.	Disc. Rew.
Map 1	441	<i>12.035</i>	11.130	<i>11.992</i>	12.588
Map 2	1,296	10.672	10.159	10.947	11.069
Map 3	400	<i>13.722</i>	13.249	<i>13.701</i>	14.380
Map 4	65,536	—	15.384	15.564	16.447
CIT	4,900	11.178	11.105	11.126	11.151
CMU	17,689	2.839	2.688	2.824	2.906
ISR	1,849	14.168	<i>13.947</i>	<i>13.997</i>	14.335
MIT	2,401	6.663	6.641	6.648	6.681
PENTAGON	2,704	<i>16.031</i>	15.162	<i>15.976</i>	16.312
SUNY	5,476	11.161	11.130	11.139	11.110

For each of the different scenarios in Fig. 2, we ran the four algorithms above and then tested the computed policy for 1,000 independent trials of 100 steps each, in the smaller environments, and 250 time-steps each, in the larger environments. The obtained performance in terms of total discounted reward can be found in Table 1.

The LAPSI algorithm performed very close to the optimal MMDP policy in all environments, in spite of the significant difference in terms of state information available to both methods. Also, in most scenarios, LAPSI and IDMG performed similarly. The only exceptions are Map 2, where LAPSI outperformed IDMG, and ISR, where IDMG outperformed LAPSI. Interestingly, however, the difference in terms of time-to-goal in the ISR environment is not significant. In any case, our results agree with previous ones that showed that IDMG attained close-to-optimal performance in most such scenarios [17]. Another interesting observation is that MPSI typically performed worse than the other methods. As pointed out before, since an agent in MPSI considers the other agents to be selfish and disregard mis-coordinations (each is focused only on its individual goal), it is expected that the agent following MPSI is more “cautious” and takes longer time to reach the goal.

Given the similar performance of IDMG and LAPSI, one may question the advantage of adopting the latter over the former. There are at least two clear

⁵ Both the penalty and the increased action failure rate imply that there is both reward and transition dependence in the interaction areas.

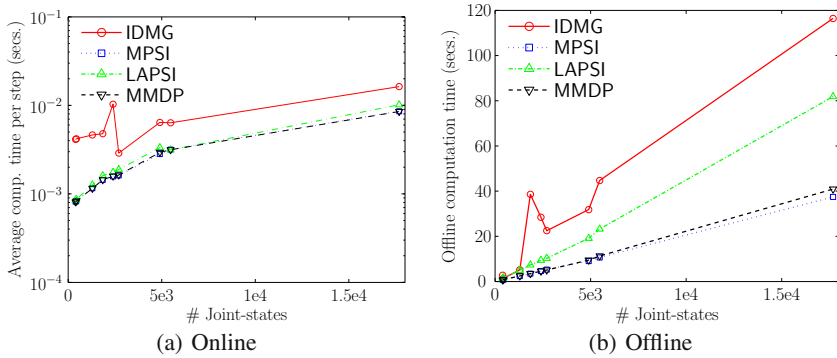


Fig. 3 Computation time for the different algorithms as a function of the problem dimension

advantages. First of all, since the IDMG method requires the computation of several equilibria both during off-line planning and during on-line execution, the computational complexity of the IDMG algorithm may quickly become prohibitive, in scenarios with large action spaces and/or with many interaction areas. To assess whether this is indeed so, we compared the computational effort of our methods with that of IDMG, both in terms of the average off-line computation time and the on-line computation time (see Fig. 3). Clearly, both MPSI and LAPSI are significantly more efficient than the IDMG algorithm, according to any of the two performance metrics. It is also interesting to note how the average computation times evolve with the dimension of the problem.

The second advantage of LAPSI becomes evident by noting that the IDMG method is, by construction, unable to consider future interactions when planning for the action in a non-interaction area. In this sense, the IDMG algorithm is “myopic” to interactions and only handles these as it reaches an interaction area. This can have a negative impact on the performance of the method, as illustrated in the final test scenario (Fig. 4). In this environment, and ignoring the interaction, Robot 1 can reach its goal by using either of the narrow pathways, since both trajectories have the same length. However, Robot 2 should use the upper pathway, since it is significantly faster than using the lower pathway.

By using the IDMG algorithm, Robot 2 goes for the upper pathway while Robot 1 chooses randomly between the two. For concreteness, let’s suppose that Robot 1 chooses to go for the upper pathway. In this case, according to the IDMG algorithm,

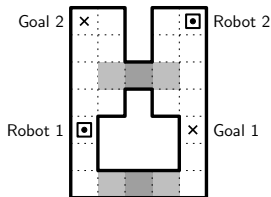


Fig. 4 Example scenario where avoiding the interaction may be beneficial

both robots reach the interaction area simultaneously and Robot 1 must move out of the way for Robot 2 to go on. This means that, in total, the two robots take a mean time of 9 steps to reach the goal. If, instead, Robot 1 takes the lower pathway, the two robots will reach their goal states in 8 steps. Since the IDMG algorithm chooses between the two randomly – or, at least, has no way to differentiate between the two – the average time to the goal is 8.5 time-steps. We ran 1,000 independent trials using the IDMG algorithm in this scenario and, indeed, obtained an average of 8.485 steps to goal, with a standard deviation of 0.5. Clearly, it seems possible to do better in this scenario by considering more convenient to use the lower pathway.

For comparison purposes, we also ran 1,000 independent trials using the LAPSI algorithm in this same scenario. Out of 1,000 trials, Robot 1 *always* picked the lower pathway. As expected, the group had an average time-to-goal of 8 time-steps with a variance of 0. Notice that this difference could be made arbitrarily large by increasing the “narrow doorway” to an arbitrary number of states, thus causing an arbitrarily large delay. As such, in scenarios such as the one above, where interactions should be considered even outside interaction areas, our methods present a clear advantage over the IDMG algorithm.

6 Conclusion

As mentioned in Section 1, Dec-SIMDPs are particularly suited for modeling several multi-robot problems. On one hand, unlike models such as MMDPs, Dec-SIMDPs do not assume full joint state observability that, in a multi-robot scenario, is tantamount to having the robots perceive the state of the other robots at every step. In most settings, this would require the agents to flawlessly communicate in a continued manner, which is quite unrealistic. On the other hand, due to their physical limitations, robots are generally bound to interact locally and, when doing so, they are most likely in a position where communication is possible. Local interactions and communication are abstracted in the Dec-SIMDPs model in the notions of *interaction areas* – meaning that the interaction among robots is “local” and limited to these areas – and *observable interactions* – meaning that, when interacting, robots have access to joint state information, possible through communication. While a Dec-SIMDP is a subclass of Dec-MDPs – and hence any problem modeled as a Dec-SIMDP can be modeled as a Dec-MDP, – the form of interaction explicitly abstracted in Dec-SIMDPs is particularly suited for multi-robot scenarios and allows algorithms such as IDMG, LAPSI and MPSI to exploit them for efficient planning.

Concerning the methods, both the LAPSI and the MPSI algorithm allow each agent to track the other agents in the environment using a belief vector that is then used to choose the actions. The difference between the two algorithms lies in the assumed policy for the other agents. In MPSI and LAPSI, these “modeling strategies” are used to abstract the decision process of each agent into a single-agent decision process (a POMDP). Although we proposed a solution technique based on the generalized α -vectors, the same principle can be used with any other POMDP solver.

Also, the ability that both MPSI and LAPSI have to track the other agent allows the planning process to take into consideration the possibility of future interaction. This, as seen in the example in Fig. 4, is an important property of the method that overcomes one important limitation of the IDMG algorithm.

It is also interesting to notice that the generalized α -vectors used in MPSI and LAPSI can be interpreted in terms of an associated MDP. By comparing the optimal policy in this MMDP and the optimal policies from the individual MDPs it should be possible to pinpoint those joint-states in which the joint action significantly differs from the one prescribed by the individual MDPs and in which the actions for each agent greatly depend on the state of the other agents. This provides one recipe for choosing the interaction states as those in which individual state-information is not sufficient to determine the best action. In [16] a similar approach is used to implement decentralized execution of a jointly optimal policy.

Finally, several open questions remain to be explored. One is concerned with the worst-case complexity of Dec-SIMDP. Is a Dec-SIMDP reducible to any of the simpler Dec-MDP subclasses for which complexity results are known? Another interesting question arises from the observation that, as a particular case of a Dec-(PO)MDP, exact Dec-POMDP methods available (*e.g.*, [4]) can be applied to solve Dec-SIMDPs. It remains an open question whether it is possible to construct a more specific *optimal* solution method that actually leverages the particular structure of Dec-SIMDPs, or whether this structure actually brings a benefit in terms of computational complexity.

Acknowledgements. This research was partially sponsored by the Portuguese Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC Program funds and under the CMU-Portugal Program, and by the Information and Communications Technologies Institute (ICTI), www.icti.cmu.edu.

References

- [1] Allen, M., Zilberstein, S.: Complexity of Decentralized Control: Special Cases. In: Adv. Neural. Information Proc. Systems, pp. 19–27 (2009)
- [2] Becker, R., Zilberstein, S., Lesser, V., Goldman, C.: Solving transition independent decentralized Markov decision processes. *J. Artif. Intell. Res.* 22, 423–455 (2004)
- [3] Bernstein, D., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.* 27(4), 819–840 (2002)
- [4] Bernstein, D., Amato, C., Zilberstein, S.: Policy iteration for decentralized control of Markov decision processes. *J. Artif. Intell. Res.* 34, 89–132 (2009)
- [5] Gerkey, B., Mataric, M.: Sold!: Auction methods for multirobot coordination. *IEEE T. Robot. Autom.* 18(5), 758–768 (2002)
- [6] Ghavamzadeh, M., Mahadevan, S., Makar, R.: Hierarchical multiagent reinforcement learning. *J. Auton. Agent Multiag.* 13(2), 197–229 (2006)
- [7] Goldman, C., Zilberstein, S.: Decentralized control of cooperative systems: Categorization and complexity analysis. *J. Artif. Intell. Res.* 22, 143–174 (2004)
- [8] Guestrin, C., Koller, D., Parr, R.: Multiagent planning with factored MDPs. In: Adv. Neural Information Proc. Systems, pp. 1523–1530 (2001)

- [9] Kearns, M., Littman, M., Singh, S.: Graphical models for game theory. In: Conf. Uncert. Artif. Intell., pp. 253–260 (2001)
- [10] Kok, J., Hoen, P., Bakker, B., Vlassis, N.: Utile coordination: Learning interdependencies among cooperative agents. In: IEEE Symp. Comput. Intell. Games, pp. 61–68 (2005)
- [11] Littman, M., Cassandra, A., Kaelbling, L.: Learning policies for partially observable environments: Scaling up. In: Int. Conf. Mach. Learn., pp. 362–370 (1995)
- [12] Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning in infinite-horizon partially observable Markov decision problems. In: AAAI Conf. Artif. Intell., pp. 541–548 (1999)
- [13] Melo, F., Veloso, M.: Local Multiagent Coordination in Decentralized MDPs with Sparse Interactions. Tech. Report CMU-CS-10-133, CS Dep., Carnegie Mellon Univ. (2010)
- [14] Mostafa, H., Lesser, V.: Offline planning for communication by exploiting structured interactions in decentralized MDPs. Tech Rep. TR 2009-020, CS Dep., Univ. Massachusetts (2009)
- [15] Parker, L.: ALLIANCE: An architecture for fault-tolerant multirobot cooperation. IEEE T. Robot. Autom. 14(2), 220–240 (1998)
- [16] Roth, M., Simmons, R., Veloso, M.: Exploiting factored representations for decentralized execution in multiagent teams. In: Int. Conf. Auton. Agent Multiag., pp. 469–475 (2007)
- [17] Spaan, M., Melo, F.: Interaction-driven Markov games for decentralized multiagent planning under uncertainty. In: Int. Conf. Auton. Agent Multiag., pp. 525–532 (2008)
- [18] Stone, P.: Layered learning in multiagent systems. PhD thesis, Carnegie Mellon Univ. (1998)
- [19] Varakantham, P., Kwak, J., Taylor, M., Marecki, J., Scerri, P., Tambe, M.: Exploiting coordination locales in distributed POMDPs via social model shaping. In: Int. Conf. Autom. Plan Scheduling, pp. 313–320 (2009)
- [20] Xin Jiang, A., Leyton-Brown, K., Bhat, N.: Action-graph games. Tech Rep. TR-2008-13, Univ. British Columbia (2008)