GSMDPs for Multi-Robot Sequential Decision-Making

João V. Messias
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
jmessias@isr.ist.utl.pt

Matthijs T. J. Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Pedro U. Lima
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
pal@isr.ist.utl.pt

Abstract

Markov Decision Processes (MDPs) provide an extensive theoretical background for problems of decision-making under uncertainty. In order to maintain computational tractability, however, real-world problems are typically discretized in states and actions as well as in time. Assuming synchronous state transitions and actions at fixed rates may result in models which are not strictly Markovian, or where agents are forced to idle between actions, losing their ability to react to sudden changes in the environment. In this work, we explore the application of Generalized Semi-Markov Decision Processes (GSMDPs) to a realistic multi-robot scenario. A case study will be presented in the domain of cooperative robotics, where real-time reactivity must be preserved, and synchronous discrete-time approaches are therefore sub-optimal. This case study is tested on a team of real robots, and also in realistic simulation. By allowing asynchronous events to be modeled over continuous time, the GSMDP approach is shown to provide greater solution quality than its discrete-time counterparts, while still being approximately solvable by existing methods.

Introduction

Planning under uncertainty has long been an actively researched topic in the fields of Operations Research and Artificial Intelligence. In many realistic environments, it is necessary to take into account uncertainty in actions and/or observations of an agent as it interacts with its environment. A Markov Decision Process (MDP) is a widely known and well-studied mathematical framework to model problems where the outcome of an agent's actions is probabilistic, but knowledge of the agent's state is assumed (Puterman 1994). Since its introduction, the versatility of the MDP framework has led to its application in diverse "real-world" domains, *i.e.*, in industry, commerce, or other areas outside of research environments (White 1993). However, the number of such applications which have actually been implemented as permanent, "in-the-loop" decision making methods in their respective domains, has been manifestly small (White 1993).

This work is part of an ongoing attempt to address the limitations of MDP-based frameworks in modeling general real-world decision-making problems, with a particular emphasis on multi-robot environments (Messias, Spaan, and Lima 2010; 2011). The most general MDP approach to multiagent problems is that of the Dec-MDP framework (Bernstein et al. 2002), in which no communication between agents is assumed. Consequently, such models can quickly become intractable (Roth, Simmons, and Veloso 2007). For teams of multiple robots, it is typically safe to assume that communication is possible and relatively inexpensive. However, models which assume free communication (Multiagent MDPs, (Boutilier 1996)) or costly communication, also typically assume that state transitions are experienced synchronously by all agents, at a fixed rate. This, in turn, can lead either to sub-optimal policies in which, for instance, robots are *forced to idle* until the next time step is reached, or to the loss of the Markovian property.

The approach taken in this work lifts the assumption of synchrony in the dynamics of the system. That is, states and actions will still be regarded as discrete, but a continuous measure of time will be maintained, and state transitions under actions will be regarded as randomly occurring "events". This approach will be shown to have several advantages for multi-robot teams. First, by explicitly modeling the temporal occurrence of events in an MDP, the non-Markovian effects of state and action space discretization can be minimized, increasing solution quality. Second, since events are allowed to occur at any time, the system is fully reactive to sudden changes. And finally, communication between agents will only be required upon the occurrence of an event, as opposed to having a fixed rate.

The framework of Generalized Semi-Markov Decision Processes (GSMDPs), proposed by Younes and Simmons (2004) is ideally suited for the requirements of this work. It allows generic temporal probability distributions over events, while maintaining the possibility of modeling persistently enabled (concurrent) events, which is essential in multi-robot domains. Other related work on event-driven MDPs deals with such events without explicitly modeling the effect of continuous time: by keeping track of event histories in the system state (Becker, Zilberstein, and Goldman 2004), or by considering the occurrence of non-Markovian events as being unpredictable (Witwicki et al. 2013).

GSMDP models can be solved by commonly used discrete-time MDP algorithms, by first obtaining an equivalent (semi-)Markovian model through the use of *Phase-type* approximations of temporal distributions (Younes and Simmons 2004; Younes 2005). However, to our knowledge, this framework has never been applied in a realistic multi-robot context. We show that, by allowing event-driven plan execution, the application of the GSMDP framework to multi-robot planning problems allows us to avoid the negative effects of its synchronous alternatives, resulting in greater performance. We also take into account the fact that some events which are characteristic of robotic systems are not amenable to phase-type approximations, and that, if so, the resulting approximate systems remain semi-Markovian.

We present a case study of robots executing a cooperative task in a robotic soccer scenario. Cooperative robotics forms a particularly suitable testbed for multiagent decision-theoretic methods, since, in this domain, agents must typically operate over inherently continuous environments, and are subject to uncertainty in both their actions and observations (although the latter is not yet considered here). The dynamic nature of robotic soccer requires agents to take joint decisions which maximize the success of the team, while maintaining the possibility of reacting to unexpected events. This case study is tested in a team of RoboCup Middle-Sized League (MSL) soccer robots, both in a realistic simulation environment, and on the actual robots.

Background

This section provides the required definitions on GSMDPs which form the background of this work, and interpret the original formulation in (Younes and Simmons 2004) under the context of multi-robot problems.

Definition 1 *A multiagent Generalized Semi-Markov Decision Process (GSMDP) is a tuple* $\langle d, \mathcal{S}, \mathcal{X}, \mathcal{A}, T, \mathcal{F}, R, C, h\rangle$ *where:*

d *is the number of agents;*

$\mathcal{S} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_{n_f}$ *is the* state space, *a discrete set of possibilities for the state* s *of the process. The state space is decomposable into* n_f state factors $\mathcal{X}_i \in \{1, ..., m_i\}$ *which lie inside a finite range of integer values.*

$\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_{n_f}\}$ *is the set of all state factors;*

$\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_d$ *is a set of* joint actions. A_i *contains the individual actions of agent* i. *Each joint action* $\mathbf{a} \in \mathcal{A}$ *is a tuple of individual actions* $\langle a_1, a_2, \ldots, a_d\rangle$, *where* $a_i \in \mathcal{A}_i$.

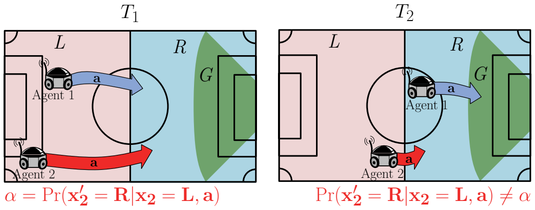
$T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ *is a* transition function, *such that* $T(s, \mathbf{a}, s') = \Pr(s'|s, \mathbf{a})$. *It models the stochasticity in the actions of the agent;*

$\mathcal{F}$ *is the* time model, *a set of probability density functions* $f^{\mathbf{a}}_{s,s'}$ *which specify the probability over the instant of the next decision episode, given that the system state changes from* s *to* s' *under the execution of joint action* $\mathbf{a}$.

$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the instantaneous* reward function. *The value* $R(s, \mathbf{a})$ *represents the "reward" that the team of agents receives for performing joint action* $\mathbf{a}$ *in state* s, *representing either its utility or cost.*

$C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the cumulative reward rate which, in addition to the* instantaneous *reward model* R, *allows the modeling of an utility or cost associated with the sojourn time at* s *while executing* $\mathbf{a}$.

$h \in \mathbb{R}^+_0$ *is the planning horizon over continuous time.*


T_1
T_2
L
R
G
Agent 1
Agent 2
a
L
R
G
Agent 1
Agent 2
a
$\alpha = \Pr(\mathbf{x_2'} = \mathbf{R}|\mathbf{x_2} = \mathbf{L}, \mathbf{a})$
$\Pr(\mathbf{x_2'} = \mathbf{R}|\mathbf{x_2} = \mathbf{L}, \mathbf{a}) \neq \alpha$

(a)

(b)

Figure 1: An example of a discretized environment in which persistently enabled events are an issue: (a) At time $\mathcal{T}_1$, two agents attempt to move from state L to G through a simple navigation action a. (a) At $\mathcal{T}_2 > \mathcal{T}_1$, agent 1 detects that it has changed its local state factor (x_1), triggering a new *joint* decision. For agent 2, $\Pr(x_2' = R|x_2 = L, a)$ is now intuitively higher, given the time that it has been moving so far. However, a memoryless discrete MDP cannot use this information. The system is not strictly Markovian.

Definition 1 formulates a decentralized multiagent problem, in the most general sense, akin to a Dec-MDP (Bernstein et al. 2002). However, it is here assumed that agents can freely communicate with each other, which means that the model is equivalent, in terms of complexity, to a centralized single-agent model defined over the whole team.

In this work, we will adopt the following definition for what constitutes an "event":

Definition 2 *An* event *in a GSMDP is an element of a countable set* $\mathcal{E}$, *the codomain of a mapping* $\Phi : \mathcal{S} \times \mathcal{S} \to \mathcal{E}$ *such that, for all* $s_1, s_1', s_2, s_2' \in \mathcal{S}$:

- *If* $\Phi(s_1, s_1') = \Phi(s_2, s_2')$, *then* $T(s_1, \mathbf{a}_1, s_1') = T(s_2, \mathbf{a}_2, s_2')$ *and* $f^{\mathbf{a}_1}_{s_1,s_1'} = f^{\mathbf{a}_2}_{s_2,s_2'}$, $\forall \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$;
- $\Phi(s_1, s_1') \neq \Phi(s_1, s_2')$ *if* $s_1' \neq s_2'$.

An event $e \in \mathcal{E}$ *is said to be* enabled *at* $\langle s, \mathbf{a}\rangle$ *if, for* s' *such that* $e = \Phi(s, s')$, $T(s, \mathbf{a}, s') > 0$. *The set of all enabled events in these conditions will be represented as* $E(s, \mathbf{a})$.

An event is then seen as an abstraction to state transitions that share the same properties.

The goal of the planning problem is to obtain a policy π^h, which provides, for each $t \in [0\,, h]$ a joint action that maximizes the total expected discounted reward. A stationary (infinite-horizon) policy will be represented simply as π.

Decision Theoretic Multi-Robot Systems

When approaching a multi-robot environment from a decision-theoretic perspective, it is often necessary to obtain a compact, discrete representation of the states and actions involved in the problem, in order to maintain its computational tractability. We here discuss the implications of this process, for the practical operation of teams of real robots.

Discretization and the Markov Property

Consider the scenario represented in Figure 1, where two robots are concurrently moving across their environment towards a target position. The exact state of the system is defined as the composition of the physical configurations of both robots. Navigation problems, such as this one, which are characteristic of multi-robot scenarios, involve naturally continuous state and action spaces. In the general case, we are interested in partitioning these spaces into discrete states and actions that can capture the decision-making problem. Additionally, for a multiagent MDP to be an accurate representation of the physical multi-robot system, the discretized model must be at least approximately Markovian (*i.e.*, memory of its past states should not be required in order to predict its future states). However, even if a physical multi-robot system is Markovian, a discrete, memoryless model of its dynamics does not necessarily hold this property.

A straightforward discretization of a multi-robot navigation problem is to map the configuration of each robot to a coarse topological location in the environment (Figure 1). A transition model could then be built by determining the probability of switching between these locations through a given control sequence. However, these probabilities would not be independent of the sojourn time of each robot inside a particular location: if one of the robots enters a new location, triggering a change of state, then the probability of the next state change being triggered by its partner should be updated to take into account the relative motion of the latter, even if it did not move to a different local state. These events are enabled in parallel, and the occurrence of one of them does not affect the expected triggering time of the other. These are said to be *persistently enabled events*. More formally,

Definition 3 *An event* $e \in \mathcal{E}$ *is* persistently enabled *from step* n *to* $n+1$ *if* $e \in E(s^n, \mathbf{a}^n)$ *and* $e \in E(s^{n+1}, \mathbf{a}^{n+1})$, *but* e *did not trigger at step* n.

In a fully Markovian system, all events are assumed to be memoryless, and so the problem represented in Figure 1 cannot be modeled directly, since those events are non-Markovian *and* persistently enabled. Non-Markovian state transitions can be modeled under the framework of Semi-Markov Decision Processes (SMDPs); however, SMDPs cannot model persistently enabled transitions.

Effects on Real-Time Performance

The common approach to minimize the non-Markovian effects induced by state and action discretization is to force the agents to act periodically in time, at a pre-defined rate, and in synchrony. This means that any changes in the state of the system are only observed by the agents at these instants, and state transition probabilities can be approximated as if they were stationary. However, as shown in Figure 2, this approach can have a negative effect on the performance of the system. If the decision period is longer than the actual duration of a given action, then robots will have to idle for the rest of that decision episode. Not only does this mean that tasks can take sub-optimal amounts of time to complete, but it also implies that the robots can no longer immediately react to sudden changes. That may be important to the long-term outcome of the decision problem. An asynchronous solution is therefore preferred. Asynchronous operation is possible under the framework of Continuous-Time MDPs (CTMDPs); however, CTMDPs cannot model non-Markovian temporal distributions (Howard 1960).

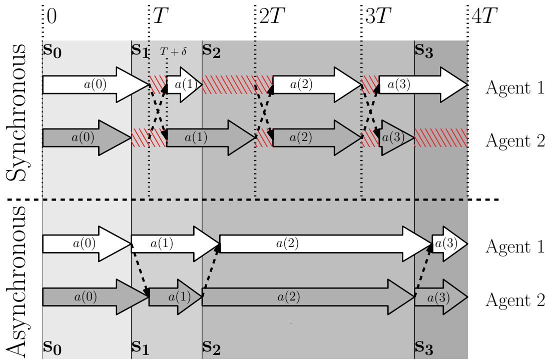


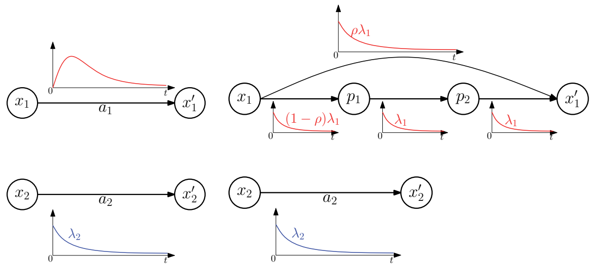
Figure 2: Action selection in synchronous and asynchronous execution of a multi-robot system. In synchronous operation, actions are jointly taken at positive multiples of $\mathcal{T}$. During the gaps between the end of a given local action (filled arrow) and the beginning of the next decision step, agents are forced to idle. Dashed arrows between agents represent communication instances; In asynchronous execution, a new decision episode starts immediately after a transition is detected, so there is no idle time. Furthermore, only the agent that detects an event needs to communicate.

The GSMDP framework adresses the issues so far described: persistently enabled events can be modeled by allowing their temporal distributions to depend on the time that they have been enabled, even if other events have meanwhile been triggered in the system. Furthermore, any non-negative distribution can be used to model the occurrence of an event. Therefore, they allow the asynchronous operation of multi-robot systems, while explicitly modeling the non-Markovian effects introduced by its discretization.

Another advantage of event-driven approaches is their communication efficiency (Figure 2). If the joint state space is not directly accessible by each agent (*i.e.*, not all state factors are observed by every agent), then agents are forced to share information. While a synchronous approach requires that each agent sends its own local state to its partners at each time-step, an asynchronous solution requires only the communication of *changes* in state factors, which effectively minimizes the number of communication episodes accross the team. Minimizing communication is a relevant problem for scenarios in which agents spend energy to transmit data, or where transmissions can be delayed or intercepted (Messias, Spaan, and Lima 2011; Roth, Simmons, and Veloso 2007; Spaan, Oliehoek, and Vlassis 2008; Spaan, Gordon, and Vlassis 2006).

GSMDPs for Multi-Robot Sequential Decision-Making

This section discusses the methodology involved in applying GSMDPs to a generic multi-robot problem, and in obtaining a useful plan from a given GSMDP model. It also describes the aspects of this work which contribute to the practical use of the theory of GSMDPs in real multi-robot scenarios.



(a) (b)

Figure 3: Approximating non-Markovian events through phase-type distributions. The topmost temporal distribution in Figure 3a is approximated through the Markov chain shown in its place, in Figure 3b. The newly added Markov chain models a Generalized Erlang distribution with three phases, which matches the first two moments of the original distribution. The system in Figure 3b is fully Markovian.

Modeling Events

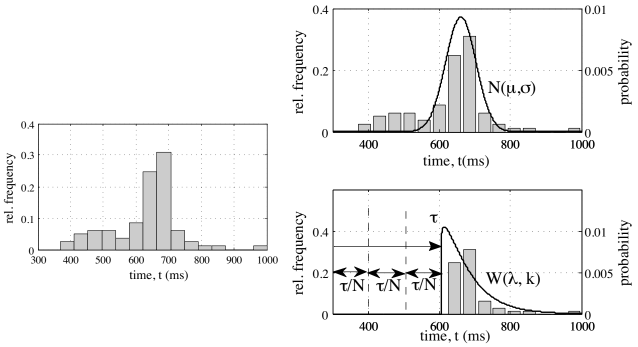
When modeling a multi-robot problem as a (G)SMDP, given a suitable (discrete) state and action space, the identification of the stochastic models of the system, T and $\mathcal{F}$, must be carried out. At this point, it is useful to group state transitions into $\mathcal{E}$, as per Definition 2. For example, for a set of identical robots, each with a state factor representing battery level, the event of running low on battery would be mapped by the same change in any of those factors, and so only one temporal distribution and state transition probability would need to be specified.

For every event in $\mathcal{E}$, the identification procedure for T and $\mathcal{F}$ is technically simple: T can be estimated through the relative frequency of the occurrence of each transition; and by timing this transition data, it is straightforward to fit a probabilistic model over the resulting data to obtain $\mathcal{F}$.

Each non-exponential $f^{\mathbf{a}}_{s,s'}$ in $\mathcal{F}$ can be approximated as *phase-type* distribution (Younes and Simmons 2004). This replaces a given event in the system event with an acyclic Markov chain, in which each of its own states is regarded as a *phase* of the approximate distribution, and each transition is governed by a Poisson process. If this replacement is possible for every event, then the approximate system is fully Markovian, allowing it to be solved as an MDP.

There are, however, limitations to this approach. An arbitrary non-Markovian distribution, with a coefficient of variation $cv = \sigma/\mu$, where μ is its mean and σ^2 its variance, requires the $\lceil \frac{1}{cv^2} \rceil$ phases to be approximated as a Generalized Erlang distribution (one such phase-type distribution), if $cv^2 < 0.5$. This number can quickly become unreasonably large for many processes which are characteristic of robotic systems. In particular, this affects actions with a clear minimum time to their outcome, dictated by the physical restrictions of a robot (*e.g.*, navigation actions given known initial positions), since μ can be arbitrarily large.

Systems with non-Markovian events which do not admit phase-type approximations can still be analyzed as semi-Markovian Decision Processes (SMDPs), **but only if those events are never persistently enabled**, since memory between transitions cannot be kept. We propose a practical alternative, for situations in which this is not a valid assumption, and which can be used to model events with minimum triggering times: such an event can be decomposed into a sequence of deterministically timed transitions, followed by a positive distribution (typically a "lifetime" distribution, see Figure 4b). The latter can then be better approximated by a phase-type distribution with a small number of phases. This requires the addition of intermediate observable states to the system, similar in purpose to the phases of a phase-type approximation, which act as "memory" for the original non-Markovian event. The length of this deterministic sequence can be adjusted to increase the quality of the approximation. Note that deterministically timed transitions are non-Markovian themselves, so the system is still an SMDP.



(a) (b)

Figure 4: (a) Temporal distribution of the event of switching agent roles after a pass, in our experimental domain. (b) Two modeling approaches. **Top**: As a (truncated) normal distribution. Would require 238 phases for a direct Generalized Erlang approximation; **Bottom**: A sequence of deterministically timed events, followed by a Weibull distribution (2 phases). 8 states are required, in total, for the approximation. Right-tailed probability mass is discarded.

Solving a GSMDP

The direct solution of a general GSMDP is a difficult problem, since typical dynamic programming approaches cannot be directly applied under its non-Markovian dynamics. However, in the case where, after introducing approximate phase-type distributions where possible, $\mathcal{F}$ still contains non-exponential distributions, the system can still be approximated by a discrete-time model by first describing its dynamics as an SMDP. Let $U(s,a)$ represent the total (cumulative and instantaneous) reward at (s,a). The value of executing a stationary SMDP policy is:

$$
\begin{aligned}
&V^{\pi}(s) = \\
&= U(s,\pi(s)) + \sum_{s'\in\mathcal{S}} V^{\pi}(s') \int_0^{\infty} \Pr(t, s' \mid s, \pi(s)) e^{-\gamma t} dt \\
&= U(s,\pi(s)) + \sum_{s'\in\mathcal{S}} \mathcal{L}\{f^{\pi(s)}_{s,s'}(t)\} T(s,\pi(s),s') V^{\pi}(s') \quad ,
\end{aligned}
$$

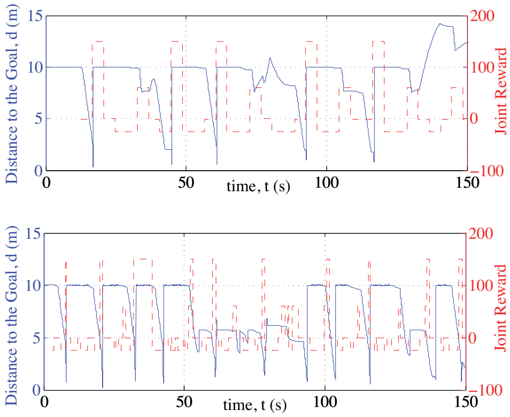


Figure 5: Simulated results. Distance from the ball to the goal (blue, solid) and accrued joint reward (red, dashed) over time. Top: using an MDP model with fixed time-step $\mathcal{T} = 4\,s$; Bottom: using the GSMDP formulation of the same problem. Jumps in reward correspond to new decision episodes. Rewards of 150 correspond to a shooting actions, and those equal to 60 correspond to passing instances in which robots switch roles. Whenever a goal is scored (the distance tends to 0), the ball is reset to its original position. Here, the robots could control and kick the ball efficiently.

where $\Pr(\tau, s' \mid s, a)$ is the joint distribution over resulting states and decision instants, $\gamma \in \mathbb{R}_0^+$ is the discount factor of the problem, and $\mathcal{L}\{\cdot\}$ denotes the Laplace transform. This well-known result allows an SMDP to be viewed as an equivalent discrete-time MDP with state-action dependent discount rates $\gamma(s, \mathbf{a}) = \mathcal{L}\{f_{s,s'}^{\mathbf{a}}(t)\}$ (Puterman 1994; Bradtke and Duff 1995). This, in turn, forms a very positive practical result, since virtually all solution algorithms for MDPs can also be applied to a GSMDP formulated approximately in this way.

Experimental Results

A common task in robotic soccer is to have various robotic soccer players cooperating in order to take the ball towards their opponent's goal. In the two-agent case, one of these players should carry the ball forward, and the other should position itself so that it may receive a pass from its partner, if necessary. During the course of their task, the robots may randomly encounter obstacles. The team must jointly decide which robot should carry the ball, and whether or not to perform a pass in order to avoid imminent obstacles, since it is difficult to avoid collisions while carrying the ball.

In previous work (Messias, Spaan, and Lima 2010), we modeled a version of this problem as a multiagent POMDP. Here, we assume joint full observability in order to instantiate it as a GSMDP. The resulting model has 126 states across 4 state factors. As in (Messias, Spaan, and Lima 2010), there are 36 joint actions. Agents are rewarded for scoring a goal (150) and for successfully switching roles whenever obstacles are blocking the attacker (60).

Every transition was timed and modeled, either according to exponential distributions, where possible; through uniform distributions — the time of entry of the dribbling robot into one of the field partitions; or through truncated normal distributions — the time to a role switch after a pass occurs, represented in Figure 4a. The latter was kept in its normal parameterization, since no concurrent events can trigger in that situation. The model was then reduced to an SMDP by replacing all uniform distributions with phase-type approximations. In order to minimize the state space size, the same phase variable was used to model all phase-type distributions, depending on the context. The value iteration algorithm was used to solve the approximate SMDP.

Simulation Results

Part of our experimental results were gathered using a realistic robotics simulator. In an initial analysis, the abilities of the simulated robots were extended in order to allow them to more efficiently dribble and kick the ball, so that their reactivity to events is not affected by their physical limitations when acting. Figure 5 compares real-time profiles of the system, under these conditions, when executing an event-driven GSMDP solution and a discrete-time multiagent MDP solution with a fixed time-step. These execution profiles are characterized by the distance between the ball and the goal, alongside the reward associated with the joint state and action, accrued at decision instants. While the MDP system is committed to performing the same action throughout the duration of the time-step, the GSMDP reacts asynchronously to events, which eliminates any idle time in the actions of the robots, resulting in more frequently scored goals.

Real Robot Results

The performance of the synchronous (fixed time-step) and event-driven (GSMDP) approaches to this problem in the real team of robots was quantitatively evaluated by testing a synchronous MDP solution with a series of different fixed time-steps, as well as the GSMDP solution. The performance metric for this comparison is the time between consecutive goals using each model. The results are shown in Figure 7. The amount of trials that could be run on the real robots was limited by total time: the average sample size is 5 scored goals for each model (9 for the GSMDP and the best performing MDP). In order to provide further statistical validity for these real robot results, simulated trials were run under equivalent conditions (considering all actuation limitations), in runs of 120 seconds each, to a total of 50 goals per model (box plot in Figure 7a).

The average and median time between goals was shorter with the GSMDP solution than with any of the synchronous MDPs. The time-step of the synchronous models was shown to have a significant, non-monotonic effect on performance. The best MDP model, with a time-step of 0.4 seconds, underperformed the GSMDP model both in the real robot trials (one-way ANOVA, $p = 0.063$), and in the corresponding simulated trials ($p = 0.079$). For time-steps below this value, agents acted on outdated information, due to communication/processing delays, and had difficulty in switching roles (note from Figure 4b that the minimum time for a role switch during a pass is also $\sim 0.4s$). For larger time-steps,

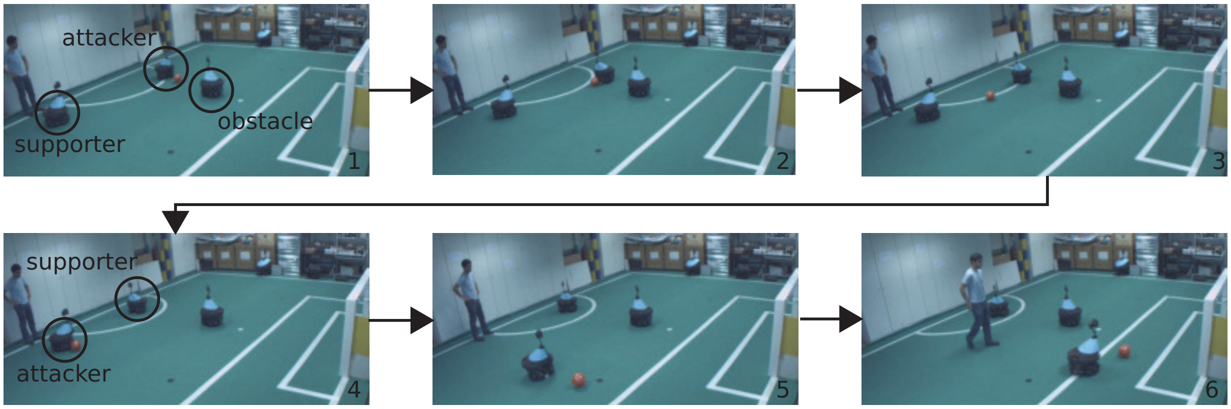
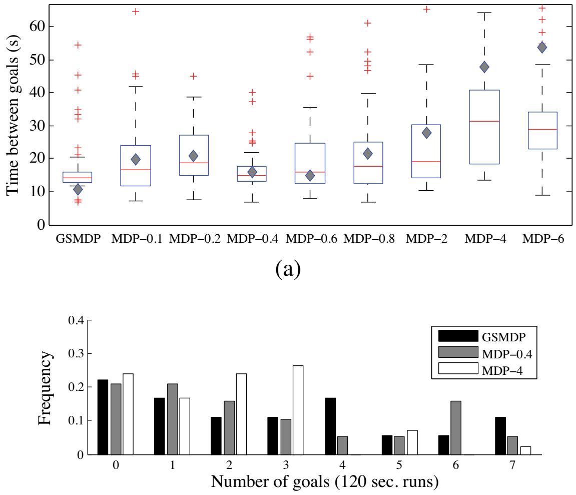


Figure 6: Sequence showing two robots cooperating in order to avoid an obstacle and score a goal (from left to right, top to bottom), in our experimental setup. The team was executing an event-driven GSMDP policy. The ability of the robots to handle the ball individually is very limited, which makes this type of cooperation necessary. In image 4 a role switch has occurred, after the successful passing of the ball. A video of this experiment can be seen at: http://masplan.org/MessiasAAAI2013.



(a)

(b)

Figure 7: Performance of GSMDP / MDP models. Synchronous models are labeled as MDP-T, with T the decision period (seconds). (a) Median time difference between goals, on the real robot trials (diamond markers). Equivalent simulated trials are represented in the underlying box plot. (b) Frequency of goals per each trial, for the GSMDP, and the best and worst MDP models ($0.4s$, $4s$, respectively). Trials with 0 goals are indicative of random system failures.

loss of reactivity and the corresponding idle time in the system also lowered the resulting performance. The average duration of decision episodes (and communication period) with the GSMDP model was $1.09s$. Since the frequency of communication episodes for synchronous MDP models is $2/T$ (Figure 2), this implies a reduction in communication usage of 81.7% with respect to the best MDP model.

Random system failures, occurring mostly due to robot navigation problems, or unmodeled spurious effects, were independent of the modeling approach (Figure 7b).

Figure 6 shows an image sequence of a typical trial. A video containing this trial, and showcasing the differences in behavior between the synchronous MDP and GSMDP approaches to this problem, both in the real and simulated environments, can be found at: http://masplan.org/MessiasAAAI2013.

Conclusions and Future Work

Multi-robot systems form particularly appropriate testbeds for the study and application of multiagent decision-theoretic methods. However, there are non-trivial and often overlooked problems involved in the application of inherently discrete models such as MDPs to dynamic, physical systems which are naturally continuous.

In this work, we showed how discrete models of multi-robot systems are not fully Markovian, and how the most common work-around (which is to assume synchronous operation) impacts the performance of the system. We discussed how the GSMDP framework fits the requirements for a more efficient, event-driven solution, and the methodologies required for GSMDPs to be implemented in practice.

Future work on the topic will explore the extension of this framework to partially-observable domains, which is a relevant problem when agents cannot assume full local knowledge; and the use of bilateral phase distributions to approximate a broader class of non-Markovian events.

Acknowledgments

This work was funded in part by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding, ref. PEst-OE/EEI/LA0009/2011) through the PIDDAC Program funds and was supported by project CMU-PT/SIA/0023/2009 under the Carnegie Mellon-Portugal Program. J.M. was supported by a PhD Student Scholarship, SFRH/BD/44661/2008, from the Portuguese FCT POCTI programme. M.S. is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

References

Becker, R.; Zilberstein, S.; and Goldman, C. 2004. Solving Transition Independent Decentralized Markov Decision Processes. *Journal of Artificial Intelligence Research* 22:423–455.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research* 27(4):819–840.

Boutilier, C. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, 195–210. Morgan Kaufmann Publishers Inc.

Bradtke, S., and Duff, M. 1995. Reinforcement Learning Methods for Continuous-Time Markov Decision Problems. *Advances in neural information processing systems* 7:393–400.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. New York: John Wiley & Sons, Inc.

Messias, J.; Spaan, M. T. J.; and Lima, P. U. 2010. Multi-Robot Planning under Uncertainty with Communication: a Case Study. In *Multi-agent Sequential Decision Making in Uncertain Domains*. Workshop at AAMAS10.

Messias, J.; Spaan, M. T. J.; and Lima, P. U. 2011. Efficient Offline Communication Policies for Factored Multiagent POMDPs. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*.

Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Roth, M.; Simmons, R.; and Veloso, M. 2007. Exploiting Factored Representations for Decentralized Execution in Multi-Agent Teams. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*.

Spaan, M. T. J.; Gordon, G.; and Vlassis, N. 2006. Decentralized Planning under Uncertainty for Teams of Communicating Agents. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, 249–256. ACM.

Spaan, M. T. J.; Oliehoek, F. A.; and Vlassis, N. 2008. Multiagent Planning under Uncertainty with Stochastic Communication Delays. In *Proc. of Int. Conf. on Automated Planning and Scheduling*, 338–345.

White, D. 1993. A Survey of Applications of Markov Decision Processes. *Journal of the Operational Research Society* 44(11):1073–1096.

Witwicki, S.; Melo, F. S.; Capitán, J.; and Spaan, M. T. J. 2013. A Flexible Approach to Modeling Unpredictable Events in MDPs. In *Proc. of Int. Conf. on Automated Planning and Scheduling*.

Younes, H., and Simmons, R. 2004. Solving Generalized Semi-MArkov Decision Processes Using Continuous Phase-Type Distributions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 742–747. San Jose, California. AAAI Press.

Younes, H. 2005. Planning and Execution with Phase Transitions. In *Proceedings of the National Conference on Artificial Intelligence*, 1030–1035.