Current Robotics Reports (2023) 4:55–64
https://doi.org/10.1007/s43154-023-00104-0

Formal Modelling for Multi-Robot Systems Under Uncertainty

Charlie Street[1] · **Masoumeh Mansouri**[1] · **Bruno Lacerda**[2]

Accepted: 19 June 2023 / Published online: 15 August 2023

Abstract

Purpose of Review To effectively synthesise and analyse multi-robot behaviour, we require formal task-level models which accurately capture multi-robot execution. In this paper, we review modelling formalisms for multi-robot systems under uncertainty and discuss how they can be used for planning, reinforcement learning, model checking, and simulation.

Recent Findings Recent work has investigated models which more accurately capture multi-robot execution by considering different forms of uncertainty, such as temporal uncertainty and partial observability, and modelling the effects of robot interactions on action execution. Other strands of work have presented approaches for reducing the size of multi-robot models to admit more efficient solution methods. This can be achieved by decoupling the robots under independence assumptions or reasoning over higher-level macro actions.

Summary Existing multi-robot models demonstrate a trade-off between accurately capturing robot dependencies and uncertainty, and being small enough to tractably solve real-world problems. Therefore, future research should exploit realistic assumptions over multi-robot behaviour to develop smaller models which retain accurate representations of uncertainty and robot interactions; and exploit the structure of multi-robot problems, such as factored state spaces, to develop scalable solution methods.

Keywords Multi-robot systems · Markov models · Uncertainty

Introduction

The demand for multi-robot systems (MRSs) is increasing, due to their performance, flexibility, and fault tolerance [1, 2]. Successful multi-robot deployments have been completed in a range of domains, such as fulfilment centres [3], fruit fields [4], and roads [5]. For safe and robust multi-robot coordination in the real world, it is often desirable to consider *formal models* of the MRS, which enable policy synthesis for well-defined objectives, as well as a formal analysis of such policies. In this review paper, we consider formal models that capture the task-level behaviour of the MRS. These model high-level capabilities such as navigation or manipulation whilst abstracting the lower-level control required to implement these capabilities. Formal models are used alongside multi-robot planning [6] and reinforcement learning (RL) [7] techniques to synthesise robot behaviour, and alongside model checking [8] and simulation [9] techniques to evaluate task-level metrics of multi-robot performance. However, the success of these techniques is limited by the model's accuracy, in particular its capacity to capture and predict execution-time multi-robot behaviour [10]. For example, if we plan on an inaccurate model, our expectations of robot behaviour during planning diverge from what is observed during execution, which can lead to inefficient execution-time behaviour or robot failure in the worst case.

In this paper, we focus on modelling the *stochasticity* of MRSs as, in any environment, robot behaviour is affected by the stochastic dynamics of the environment and the other robots. For example, a mobile robot operating in an office may fail to navigate upon a door being closed unexpectedly, or it may be unable to dock at a charging station if another robot is charging for longer than expected. We begin by introducing the types of uncertainty encountered by MRSs, including uncertainty

✉ Charlie Street
c.l.street@bham.ac.uk

Masoumeh Mansouri
m.mansouri@bham.ac.uk

Bruno Lacerda
bruno@robots.ox.ac.uk

[1] School of Computer Science, University of Birmingham, Birmingham, UK

[2] Oxford Robotics Institute, University of Oxford, Oxford, UK

over action outcomes [11], a robot's current state [12], and the duration and start time of robot actions [13, 14••]. Next, we review modelling formalisms which capture these sources of uncertainty. We then describe how formal multi-robot models have been used to support advances in the application of planning, RL, model checking, and simulation techniques to MRSs.

Uncertainty in Multi-Robot Systems

In this section, we outline the common forms and sources of uncertainty experienced by MRSs.

Outcome Uncertainty Robot uncertainty is most commonly captured over discrete action outcomes [11], such as whether a grasp action is executed successfully. Stochastic outcomes can occur due to robot navigation failure [15], battery depletion [16], or stochastic features of the environment such as hazards [17], resources [18], and doors [19].

Partial Observability In some MRSs, robots only partially observe the environment, which prevents them from knowing each other's states. This is often caused by limited communication and sensing capabilities, such as imperfect localisation [20], limited network range [21], or object occlusion [22]. Under partial observability, robots form a *belief* over the true state of the environment and other robots using possibly noisy observations obtained from sensors.

Temporal Uncertainty Sources of temporal uncertainty affect the duration and start time of robot actions during execution [13, 23, 24]. Temporal uncertainty occurs in almost any robot environment, where action durations are affected by environmental disturbances, such as unknown obstacles or adverse weather conditions. For example, a mobile robot's tire may slip on a carpet whilst navigating through an office, slowing it down. Furthermore, robots may have to wait for stochastic temporal processes in the environment, such as order arrival in a fulfilment centre, before beginning task execution [25].

The Effect of Robot Interactions A particularly relevant driver of uncertainty in MRSs is the fact robots typically *share resources*, such as space or access to a charging station, and *must interact* with each other [14••]. For example, when multiple mobile robots navigate in the same physical space simultaneously, they may experience *congestion*, which increases uncertainty over action duration [23]. Alternatively, a robot manipulator may be more likely to fail a grasp if another robot is nearby, restricting its movement.

Formal Multi-Robot Models

In this section, we review modelling formalisms for MRSs, which we summarise in Table 1. At their foundation, each of these models consists of *states*, which describe a snapshot of the MRS and environment, and *transitions* between states, which define the system dynamics.

Classical Multi-Robot Models

Joint transition systems (JTSs) model MRSs with deterministic dynamics [10, 38–41]. JTS states are often factored into local states for each robot, e.g. their location and

Table 1 A summary of multi-robot modelling formalisms

Model	Stochastic outcomes	Partial observability	Temporal uncertainty	Continuous time	Transition independence	Asynchronous execution	Allows for heterogeneous teams
JTS [10]	✗	✗	✗	✗	✗	✗	✓
MMDP [6]	✓	✗	✗	✗	✗	✗	✓
TI-MMDP [26]	✓	✗	✗	✗	✓	✗	✓
CMMDP [27•]	✓	✗	✗	✗	✓	✗	✓
Team MMDP [28]	✓	✗	✗	✗	✗	✗	✓
Dec-SIMDP/IDMG [29, 30]	✓	✗	✗	✗	✗	✗	✓
SPATAP Model [31]	✓	✗	✗	✗	✓	✗	✓
TVMA per Robot [23]	✓	✗	✓	✗	✗	✓	✓
Dec-POMDP [32]	✓	✓	✗	✗	✗	✗	✓
MacDec-POMDP [33••]	✓	✓	✓	✗	✗	✓	✓
Dec-POSMDP [34]	✓	✓	✓	✗	✗	✓	✓
CTMDP [35]	✓	✗	✓	✓	✗	✓	✓
GSPN [36]	✓	✗	✓	✓	✓	✓	✗
GSMDP [37]	✓	✗	✓	✓	✗	✓	✓
MRMA [14••]	✓	✗	✓	✓	✗	✓	✓

battery level, and a shared set of global state features, such as whether doors in the environment are open. JTSs are fully deterministic and so fail to capture the stochastic dynamics of real robot environments. *Multi-agent Markov decision processes (MMDPs)* are a natural extension of JTSs to stochastic domains [6]. Similar to JTSs, MMDPs capture robots in a joint state and action space, but MMDP actions have probabilistic outcomes. MMDPs are a common formalism for MRSs and have been used to model drone fleets [42••], warehouse robots [25], and human–robot teams [43]. MMDPs and JTSs assume synchronous execution, i.e. robots execute their actions in lockstep, and all actions have the same duration. Furthermore, the joint state and action spaces yield an exponential blow-up in the number of robots being modelled. In practice, robot action durations are inherently continuous and uncertain, where robot interactions contribute towards this uncertainty [14••, 23, 24, 44, 45]. Thus, to accurately capture multi-robot behaviour, we require formalisms which model asynchronous multi-robot execution and uncertainty over action duration. One approach for explicitly doing this is to use continuous-time Markov models, which we discuss later in this section.

Avoiding the Exponential Scalability of Joint Models

The number of MMDP or JTS states and actions increases exponentially in the number of robots [6], which makes optimal solutions for planning [46], RL [47], and model checking [10] intractable. This can be improved by making different assumptions which simplify the model. In fact, there has been a significant research effort to identify realistic assumptions for specific multi-robot problems. *Transition-independent MMDPs (TI-MMDPs)* [26] and *constrained MMDPs (CMMDPs)* [27•] assume the transition dynamics of each robot are independent, but couple the MRS through rewards and shared resources, respectively. *Team MMDPs* [28] also treat the transition dynamics independently, modelling robots sequentially in the context of simultaneous task allocation and planning problems. Transition independence assumptions allow for weakly coupled models that operate outside of the joint state and action space and reduce the model size, thus facilitating the use of more efficient solution methods. However, in cases where execution-time robot interactions affect the outcome and duration of robot actions, the transition-independent models above are unable to accurately reflect the MRS.

For many multi-robot problems, robots can act independently for the majority of execution, as interactions are *sparse*. For example, two robots conducting a handover can ignore each other until they are close. *Interaction-driven Markov games (IDMGs)* [29] and *decentralised sparse interaction MDPs (Dec-SIMDPs)* [30, 48] exploit this to reduce the space complexity whilst still accounting for execution-time interactions. IDMGs and Dec-SIMDPs are equivalent and capture an MRS using an independent MDP per robot and a set of interaction MMDPs, which define joint MRS behaviour in interaction areas, such as near a doorway. Though interaction MMDPs are joint models, they are significantly smaller than the full MMDP, as they are defined over only a small fraction of the full MMDP state space. However, these models are only useful when interactions are localised to a small, fixed part of the environment. If this does not hold, they become equivalent to the full MMDP.

Finally, a commonly used approach to avoid the use of joint models whilst still considering robot dependencies and execution-time interactions is to model the MRS as a set of single-robot models that are extended to include some knowledge of the other robots. In [25, 31], *spatial task allocation problems (SPATAPs)* are modelled using single-robot models which aggregate the response of the other robots. The aggregate response is represented as a distribution which predicts whether any robot is present at a given location. This is computed by combining individual distributions over each robot's location and allows robots to predict which tasks will be handled by other robots during planning. A similar approach is taken in [23], where an MRS is modelled using single-robot *time-varying Markov automata (TVMA)* which capture the probabilistic effects of congestion caused by the other robots. In this context, congestion is represented as a distribution over the number of robots present at each area of the environment, and distributions of navigation duration under the presence of a specific number of robots are obtained from real-world multi-robot navigation data. To solve multi-robot planning problems, [24] augment single-robot models with a cost function which captures the effects of robot interactions. This cost function is then adjusted iteratively during planning to encourage robot collaboration.

Partially Observable Multi-Robot Models

Partially observable MDPs (POMDPs) are widely used to model partially observable problems, where robots make observations which update their belief over their current state [12]. *Decentralised POMDPs (Dec-POMDPs)* extend POMDPs to multi-robot settings [32], where each robot has its own set of local observations. Dec-POMDPs have been used for warehouse robotics [49], cooperative package delivery [34], and teams of unmanned aerial vehicles [50]. If the combined local observations of each robot uniquely identify the joint state, Dec-POMDPs are reduced to Dec-MDPs, which are easier to solve [32]. However, these are still joint models, and optimal solvers for both Dec-POMDPs and Dec-MDPs have even higher time complexity than MMDP solvers [32]. To reduce the space complexity related to the joint modelling in Dec-POMDPs, [51, 52•] consider decoupling them into local POMDPs for each robot. For each of

these local POMDPs, they compute a distribution which captures how external state factors influence its local state. These external state factors include the states of the other robots. This is then used to marginalise out the external state factors to construct single-robot POMDPs. This *influence based abstraction* produces smaller models. However, computing influence distributions is intractable in general [52•].

Another class of relevant POMDP-based models are *macro action Dec-POMDPs (MacDec-POMDPs)* [33••] and *decentralised partially observable semi-MDPs (Dec-POSMDPs)* [34], which consider *macro actions* which execute a series of primitive low-level actions, such as moving one grid cell forward. This hierarchical paradigm is based on the options framework [53] for MDPs and has two main benefits. First, it reduces model size by leveraging existing behaviour, such as navigation, and modelling behaviour at the macro-action level, rather than each time step. Second, the use of temporally extended actions seamlessly enables asynchronous action execution. Each MacDec-POMDP and Dec-POSMDP has an underlying Dec-POMDP which captures the low-level actions that form the macro actions. For MacDec-POMDPs, the underlying Dec-POMDP and the policies for each macro action are assumed to be known [54]. MacDec-POMDP policies can then be evaluated by unrolling the macro actions on the low-level Dec-POMDP. Unlike MacDec-POMDPs, Dec-POSMDPs capture macro actions using distributions over their completion time, where Dec-POSMDP policies can be evaluated through simulation.

Continuous-Time Multi-Robot Models

Several models have been proposed to take into account uncertainty over action duration in the context of MRSs which are evolving asynchronously. These make use of continuous-time distributions which capture the stochasticity in robot action durations. *Continuous-time MDPs (CTMDPs)* extend MDPs to include durative transitions represented as exponential delays [35] and have been used to model multi-robot data collection problems [55]. To model asynchronous multi-robot execution, CTMDPs can be defined over a joint state and action space, similar to MMDPs. Thus, as with MMDPs, they scale exponentially in the number of robots. To mitigate this, [55] constructs single-robot CTMDPs assuming transition independence, similar to [26, 27•]. The duration of each action in a CTMDP is modelled with a single exponential distribution. This is a convenience which allows for simpler solution approaches which exploit the memoryless property of the exponential distribution, but limits the accuracy with which we can capture robot action durations.

Many multi-robot models can capture *heterogeneous* MRSs (see Table 1), where robots have different capabilities and resource usage etc. This is often achieved using local action spaces or reward functions for each robot. *Generalised stochastic Petri nets (GSPNs)* [36] are a modelling formalism for *homogeneous* MRSs, i.e. the robots are identical, where robots are represented *anonymously* as tokens. Furthermore, as in CTMDPs, durations are restricted to exponentials. GSPNs remain exponential in the team size, but robot anonymity provides a practical reduction in the number of states. GSPNs have been used to model teams of football robots [56], autonomous haulers [57], and monitoring robots [58]. *Generalised semi-MDPs (GSMDPs)* can capture concurrent execution and stochastic durations and have been applied to MRSs in [37, 44], but are complex to define and hard to solve, as GSMDPs allow for arbitrary duration distributions. *Multi-robot Markov automata (MRMA)* [14••] also allow for arbitrary duration distributions to capture asynchronous multi-robot execution in continuous time. Markov automata (MA) extend MDPs and CTMDPs by explicitly separating instantaneous robot action choice and the duration of robot actions [59]. MRMA are joint models, where robot action durations are represented as phase-type distributions (PTDs), which are sequences of exponentials capable of capturing any nonnegative distribution to an arbitrary level of precision [60]. In an MRMA, there is a different duration distribution for each spatiotemporal situation an action may be executed under, referred to as the *context*, which captures the effects of robot interactions on action execution. By separating robot decision-making from action duration, robot interactions can be detected at the instant an action is triggered by analysing the joint MRMA state. MRMA are connected to other continuous-time multi-robot models. First, GSPN semantics can be described with an MA [61]. Second, a standard solution for GSMDPs involves converting all duration distributions into PTDs [60], which produces a model similar to an MRMA [37]. However, MRMA are simpler to define and can be solved directly [62], as all durations are exponentials/PTDs by definition.

Model Applications

In this section, we discuss how the multi-robot models in Table 1 have been solved and analysed for multi-robot planning, RL, model checking, and simulation. We summarise this discussion in Table 2. Note that in Table 2, we do not list foundational works which apply to more general models, such as heuristic search approaches for MDPs which can be applied to MMDPs [46] or MA model checking techniques which can be applied to MRMA [62].

Planning

Multi-robot planning techniques synthesise robot behaviour given a formal model of the system. Many multi-robot

Table 2 Applications of the models in Table 1 for multi-robot/multi-agent problems

Model	Planning	Reinforcement learning	Model checking	Simulation
JTS [10]	[38–41]	-	[38–41]	-
MMDP [6]	[6, 42••, 63, 64]	[65, 66]	[63, 64]	-
TI-MMDP [26]	[26]	-	-	-
CMMDP [27•]	[18, 67–72]	[73, 74]	-	-
Team MMDP [28]	[28]	-	[28]	-
Dec-SIMDP/IDMG [29, 30]	[29, 30, 48]	[75, 76]	-	-
SPATAP Model [31]	[25, 31]	-	-	-
TVMA per Robot [23]	[23]	-	-	-
Dec-POMDP [32]	[50, 77–79]	[80••–88]	-	-
MacDec-POMDP [33••]	[33••, 49, 89–91]	[92–95]	-	-
Dec-POSMDP [34]	[34, 77, 96]	-	-	-
CTMDP [35]	[55, 97]	-	-	-
GSPN [36]	[57, 58, 98, 99]	-	-	[56, 99]
GSMDP [37]	[44]	-	-	-
MRMA [14••]	-	-	-	[14••]

models can be solved with standard techniques. MMDPs can be solved exactly using MDP solvers such as value or policy iteration [100, 101]. However, these methods solve for all states, making them intractable for joint multi-robot models. Heuristic and sampling-based methods such as labelled real-time dynamic programming [102] or Monte-Carlo tree search [103] improve upon the limited scalability of exact solvers by restricting search to promising areas of the state space. Despite reducing the explored states, heuristic algorithms are slow to converge on large models, but often provide anytime behaviour such that valid solutions are synthesised quickly and improved with time. The poor scalability of MMDP planning motivates planning on simplified models. For TI-MMDPs [26], transition independence allows for compact representations of reward dependencies in conditional return graphs, which admits efficient solutions. For Dec-SIMDPs and IDMGs, the single-robot MDPs and interaction MMDPs can be solved separately using standard solvers such as value iteration [29]. Similarly, the SPATAP models in [31] are single-robot MDPs which capture the effects of the other robots, and can be solved separately. CMMDP approaches typically exploit the fact that only the resource constraint couples the agents to scale to larger problems. Planning for CMMDPs has considered a range of constraints over resource consumption, such as bounding its worst-case [67], considering a chance-constraint [68, 71], and bounding its conditional value at risk [72].

MMDPs can be solved tractably if they are sufficiently small. Therefore, in [63], robots are grouped into clusters based on robot dependencies, and each cluster is solved as a separate MMDP. Similarly, in [64], robots are incrementally added to an MMDP to control scalability.

Recent work [42••] has begun to address the poor scalability of MMDP planning. There, an anytime planner for MMDPs based on Monte-Carlo tree search is presented, where robot dependencies are exploited to decompose the value function into a set of factors from which the optimal joint action can be computed. This approach scales to previously intractable problems.

Solution methods for continuous-time multi-robot models differ depending on the objective. To solve CTMDPs for time-abstract objectives, such as expected untimed reward, MDP solvers are applied to an embedded time-abstract MDP. For timed objectives, MDP solvers are instead applied to a uniformised MDP, where each state has the same expected sojourn time [37, 104, 105]. Similarly, GSPNs can be converted to an MDP [57] or an MA [58] dependent on the objective and solved with standard techniques. For MRMA, we can plan using MA solution methods [62].

Dec-POMDPs can be solved centrally to synthesise local policies for decentralised execution, which map from local action-observation histories to actions [50, 77–79]. With this, local Dec-POMDP policies are robust to communication limitations and unreliable sensors. Dec-POMDP solutions can be adapted to MacDec-POMDPs and Dec-POSMDPs to synthesise policies over macro actions. In [89], the space of macro-action policies is searched exhaustively, where efficient simulators improve the scalability of policy evaluation [49]. This approach scales poorly, which is addressed in [90], where a heuristic search method optimises finite state controllers for each robot. However, MacDec-POMDP and Dec-POSMDP solutions have not been shown to scale beyond teams of around four robots [33••, 34].

Reinforcement Learning (RL)

An alternative approach to policy synthesis is RL [47]. Planners synthesise behaviour using a model of the system,

whereas RL approaches learn behaviour using data sampled from the environment [46, 47]. Multi-robot RL problems are formulated assuming an underlying multi-robot model which is unknown prior to training. Fully observable, centralised problems can be formulated as an MMDP [65, 66] and solved using standard RL techniques such as deep Q-learning [106]. However, these techniques do not scale to multi-robot problems due to the exponential increase in the state and action space [66, 80••]. In many settings, decentralised policies are required due to limited communication or partial observability [80••, 81]. Here, multi-robot RL can be formulated as a Dec-POMDP and solved under the paradigm of centralised training with decentralised execution [107], which allows additional state information not available during execution to be used during training, such as the joint state. One example of this paradigm is QMix [80••], which uses a mixing network to estimate the joint Q value from single-robot Q values. RL techniques for Dec-POMDPs are still slow to converge, however, and so MacDec-POMDPs can be used to exploit existing behaviours and improve the efficiency of learning [92–95].

Model Checking

Model checking techniques evaluate the behaviour induced by robot policies by systematically checking if a property is satisfied in a formal robot model [10]. Properties are often specified with temporal logics such as linear temporal logic (LTL) or continuous stochastic logic (CSL). Similar to planning, many of the multi-robot models in Table 1 can be verified using techniques for more general models. For example, LTL formulae can be verified on JTSs and MMDPs using techniques for transition systems and MDPs [10]. However, exact LTL model checking approaches compute a product of the model and an automaton that captures the LTL formula, which significantly increases the state space, making them unsuitable for multi-robot problems. MRMA can be model checked against CSL formulae using model checking techniques for MA [62]. This also applies to GSPNs, which can be represented as an MA with identical semantics [61]. Similar CSL model checking techniques are available for CTMDPs [108].

Model checking and planning are often combined to synthesise guaranteed multi-robot behaviour. For LTL specifications, we can plan over a joint product automaton; however, this quickly becomes intractable. To overcome this, [28] concatenate single robot product automata through switch transitions in a team MMDP to reduce the state space. For MMDPs, in [64], robots are added incrementally to a product automaton until the full problem is solved or a fixed computational budget is exceeded. Alternatively, in [41], the product automaton is explored incrementally through sampling for MRSs modelled as a JTS. Combined planning and model checking techniques have been used for multi-robot data gathering [38, 39], monitoring [40], and mobility-on-demand [64].

Statistical model checking (SMC) techniques evaluate properties by sampling through a model given a set of robot policies, which avoids enumerating the state space [109] and bridges the gap between model checking and simulation techniques, which we discuss later in this section. In [8], SMC is used to evaluate quantitative properties of an MRS. SMC techniques can be applied to many of the models in Table 1. For example, we can use SMC techniques for MA [110] to evaluate bounded or unbounded properties on an MRMA. A drawback of SMC is a possible failure to explore states reached with low probability, which can render SMC unsuitable for safety critical systems [110].

Simulation

Simulators evaluate multi-robot behaviour by executing a set of robot policies in an abstracted environment model. Using formal multi-robot models, we can create a discrete-event simulator (DES) by sampling stochastic outcomes and durations and resolving non-determinism using robot policies. DESs mitigate the complexity of physics-based simulators such as Gazebo [111] by abstracting away low-level robot dynamics [112], allowing simulations to run magnitudes faster than real time. GSPNs, or variants thereof, have been used to simulate teams of football robots [56] and human–robot manufacturing teams [99], respectively. In [14••], a DES called CAMAS (context-aware multi-agent simulator) samples through an MRMA to evaluate task-level metrics of multi-robot performance under the effects of robot interactions, such as the time to complete a set of tasks.

Conclusions

In this paper, we reviewed modelling approaches for capturing the task-level behaviour of MRSs. We focused on stochastic models of multi-robot execution and introduced the different types of uncertainty encountered by MRSs. Furthermore, we discussed how these models have been used for multi-robot planning, RL, model checking, and simulation. Recent research has focused on constructing models which accurately capture the effects of uncertainty and robot interactions or constructing models small enough to be solved efficiently. These two objectives are opposing, as to accurately capture multi-robot execution, we often require joint models which are frequently intractable to solve or analyse. Therefore, future research should focus on developing smaller multi-robot models which still accurately capture uncertainty and robot interactions. This may be achieved by identifying realistic assumptions over the sources of

uncertainty and robot interactions, such as interactions only occurring in small portions of the state space. Exploiting these assumptions allows for smaller models which can be solved efficiently without sacrificing model accuracy. An alternative avenue for research is to exploit the structure of multi-robot problems, such as factored state spaces and dependencies between robots, to develop scalable solution methods for multi-robot models.

Funding Charlie Street and Masoumeh Mansouri are UK participants in Horizon Europe Project CONVINCE and supported by UKRI grant number 10042096. Bruno Lacerda is supported by the EPSRC Programme Grant 'From Sensing to Collaboration' (EP/V000748/1).

Declarations

Human and Animal Rights and Informed Consent This article does not contain any studies with human or animal subjects performed by any of the authors.

Conflict of Interest The authors declare no competing interests.

References

Papers of particular interest, published recently, have been highlighted as:
- • Of importance
- •• Of major importance

1. Iocchi L, Nardi D, Salerno M. Reactivity and deliberation: a survey on multi-robot systems. In: Proceedings of the workshop on balancing reactivity and social deliberation in multi-agent systems. Berlin, Heidelberg: Springer; 2000. https://doi.org/10.1007/3-540-44568-4_2.
2. Yan Z, Jouandeau N, Cherif AA. A survey and analysis of multi-robot coordination. Int J Adv Rob Syst. 2013;10(12):399.
3. Ocado Group.: what is an Ocado CFC? Available from https://www.ocadogroup.com/about-us/what-we-do/automated-ocado-customer-fulfilment-centre, 2021. Accessed 11 July 2023.
4. Khan MW, Das GP, Hanheide M, Cielniak G. Incorporating spatial constraints into a Bayesian tracking framework for improved localisation in agricultural environments. In: Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas: IEEE; 2020. p. 2440–2445. https://doi.org/10.1109/IROS45743.2020.9341013.
5. Robotics 24/7 Staff.: AutoX passes 1000 vehicle milestone for its RoboTaxi fleet, the largest in China. Available from: https://www.robotics247.com/article/autox_passes_1000_robotaxi_fleet_milestone_expands_san_francisco_testing. Accessed 11 July 2023.
6. Boutilier C. Planning, learning and coordination in multiagent decision processes. In: Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK); 1996. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., p. 195–210.
7. Buşoniu L, Babuška R, De Schutter B. Multi-agent reinforcement learning: an overview. Innov Multi-Agent Syst Applic. 2010;1:183–221.
8. Herd B, Miles S, McBurney P, Luck M. Quantitative analysis of multiagent systems through statistical model checking. In: Proceedings of the International Workshop on Engineering Multi-Agent Systems. Berlin, Heidelberg: Springer; 2015. p. 109–130. https://doi.org/10.1007/978-3-319-26184-3_7.
9. Damas B, Lima P. Stochastic discrete event model of a multi-robot team playing an adversarial game. In: Proceedings of the IFAC/EU-RON Symposium on intelligent autonomous vehicles. vol. 37(8). Elsevier; 2004. p. 974–979. https://doi.org/10.1016/S1474-6670(17)32107-9.
10. Baier C, Katoen JP. Principles of model checking. Cambridge: MIT Press; 2008.
11. Puterman ML. Markov decision processes: discrete stochastic dynamic programming. USA: John Wiley & Sons, Inc. 1994. https://doi.org/10.1002/9780470316887.
12. Kaelbling LP, Littman ML, Cassandra AR. Planning and Acting in Partially Observable Stochastic Domains. Artif Intell. 1998;101(1–2):99–134.
13. Boyan JA, Littman ML. Exact solutions to time-dependent MDPs. In: Proceedings of Advances in Neural Information Processing Systems (NIPS). Denver, CO: MIT Press; 2000. p. 1026–1032.
14. •• Street C, Lacerda B, Staniaszek M, Mühlig M, Hawes N. Context-Aware Modelling for Multi-Robot Systems Under Uncertainty. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS); 2022. p 1228–1236. **This paper introduces MRMA, the first formulation to explicitly capture the effects of execution-time robot interactions on action duration, as well as CAMAS, which samples through an MRMA to evaluate task-level metrics of multi-robot performance.**
15. Ma H, Kumar TS, Koenig S. Multi-agent path finding with delay probabilities. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence; San Francisco, California, USA; 2017. p. 3605–3612.
16. Tomy M, Lacerda B, Hawes N, Wyatt JL. Battery charge scheduling in long-life autonomous mobile robots via multi-objective decision making under uncertainty. Robot Auton Syst. 2020;133:103629.
17. Tihanyi D, Lu Y, Karaca O, Kamgarpour M. Multi-robot task allocation for safe planning under dynamic uncertainties. arXiv preprint arXiv:210301840. 2021. https://doi.org/10.48550/arXiv.2103.01840.
18. de Nijs F, Spaan M, de Weerdt M. Preallocation and planning under stochastic resource constraints. New Orleans, Louisiana, USA: In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32; 2018. p. 4662–4669.
19. Deng K, Chen Y, Belta C. An approximate dynamic programming approach to multiagent persistent monitoring in stochastic environments with temporal logic constraints. IEEE Trans Autom Control. 2017;62(9):4549–63.
20. Vanegas F, Campbell D, Roy N, Gaston KJ, Gonzalez F. UAV tracking and following a ground target under motion and localisation uncertainty. In: Proceedings of the IEEE Aerospace Conference. Big Sky, MT, USA: IEEE; 2017. p. 1–10. https://doi.org/10.1109/AERO.2017.7943775.

21. Capitan J, Spaan MT, Merino L, Ollero A. Decentralized multi-robot cooperation with auctioned POMDPs. Int J Robot Res. 2013;32(6):650–71.
22. Hubmann C, Quetschlich N, Schulz J, Bernhard J, Althoff D, Stiller C. A POMDP maneuver planner for occlusions in urban scenarios. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV). IEEE; 2019. p. 2172–2179. https://doi.org/10.1109/IVS.2019.8814179.
23. Street C, Pütz S, Mühlig M, Hawes N, Lacerda B. Congestion-Aware Policy Synthesis for Multirobot Systems. IEEE Transactions on Robotics. 2022;38(1). https://doi.org/10.1109/TRO.2021.3071618.
24. Zhang S, Jiang Y, Sharon G, Stone P. Multirobot symbolic planning under temporal uncertainty. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems; 2017. p. 501–510.
25. Claes D, Oliehoek F, Baier H, Tuyls K. Decentralised online planning for multi-robot warehouse commissioning. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems; 2017. p. 492–500.
26. Scharpff J, Roijers D, Oliehoek F, Spaan M, de Weerdt M. Solving transition-independent multi-agent MDPs with sparse interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30; 2016. p. 3174–3180.
27.• De Nijs F, Walraven E, De Weerdt M, Spaan M. Constrained multiagent Markov decision processes: a taxonomy of problems and algorithms. J Artif Intell Res. 2021;70:955–1001. **This paper provides a comprehensive taxonomy of CMMDP problems and solutions and is an effective starting point for new researchers in the area.**
28. Faruq F, Parker D, Lacerda B, Hawes N. Simultaneous task allocation and planning under uncertainty. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE; 2018. p. 3559–3564. https://doi.org/10.1109/IROS.2018.8594404.
29. Spaan MTJ, Melo FS. Interaction-driven Markov games for decentralized multiagent planning under uncertainty. Estoril, Portugal: In: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS); 2008. p. 525–532.
30. Melo FS, Veloso M. Decentralized MDPs with sparse interactions. Artif Intell. 2011;175(11):1757–89.
31. Claes D, Robbel P, Oliehoek F, Tuyls K, Hennes D, Van der Hoek W. Effective approximations for multi-robot coordination in spatially distributed tasks. In: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems; 2015. p. 881–890.
32. Bernstein DS, Givan R, Immerman N, Zilberstein S. The complexity of decentralized control of Markov decision processes. Math Oper Res. 2002;27(4):819–40.
33.•• Amato C, Konidaris G, Kaelbling LP, How JP. Modeling and planning with macro-actions in decentralized POMDPs. J Artif Intell Res. 2019;64:817–859. **This paper presents MacDec-POMDPs and extends three Dec-POMDP solvers to handle macro actions. Furthermore, this paper demonstrates how planning with macro actions can scale to previously intractable Dec-POMDP problems.**
34. Omidshafiei S, Agha-Mohammadi AA, Amato C, Liu SY, How JP, Vian J. Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions. Int J Robot Res. 2017;36(2):231–58.
35. Guo X, Hernández-Lerma O. Continuous-time Markov decision proesses: theory and applications. Springer-Verlag, Berlin Heidelberg; 2009.
36. Balbo G. Introduction to generalized stochastic Petri nets. In: Proceedings of the International School on Formal Methods for the Design of Computer, Communication and Software Systems. Berlin, Heidelberg: Springer; 2007. p. 83–131. https://doi.org/10.1007/978-3-540-72522-0_3.
37. Younes HL, Simmons RG. Solving generalized semi-Markov decision processes using continuous phase-type distributions. San Jose, California: In: Proceedings of the 19th AAAI Conference on Artificial Intelligence; 2004. p. 742–747.
38. Gujarathi D, Saha I. MT*: Multi-robot path planning for temporal logic specifications. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2022. p. 13692–13699.
39. Ulusoy A, Smith SL, Ding XC, Belta C, Rus D. Optimality and robustness in multi-robot path planning with temporal logic constraints. Int J Robot Res. 2013;32(8):889–911.
40. Kloetzer M, Ding XC, Belta C. Multi-robot deployment from LTL specifications with reduced communication. In: Proceedings of the IEEE Conference on Decision and Control and European Control Conference. Orlando, FL, USA: IEEE; 2011. p. 4867–4872. https://doi.org/10.1109/CDC.2011.6160478.
41. Kantaros Y, Zavlanos MM. STyLuS*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems. Int J Robot Res. 2020;39(7):812–36.
42.•• Choudhury S, Gupta JK, Morales P, Kochenderfer MJ. Scalable Online planning for multi-agent MDPs. J Artif Intell Res. 2022;73:821–846. **This paper presents a state-of-the-art anytime planner for MMDPs based on Monte Carlo tree search which can solve previously intractable problems.**
43. Unhelkar VV, Li S, Shah JA. Semi-supervised learning of decision making models for human-robot collaboration. In: Proceedings of the Conference on Robot Learning. PMLR; 2020. p. 192–203.
44. Messias JV, Spaan M, Lima P. GSMDPs for multi-robot sequential decision-making. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence; 2013. p. 1408–1414. https://doi.org/10.1609/aaai.v27i1.8550.
45. de Weerdt MM, Stein S, Gerding EH, Robu V, Jennings NR. Intention aware routing of electric vehicles. IEEE Trans Intell Transp Syst. 2015;17(5):1472–82.
46. Mausam, Kolobov A. Planning with Markov decision processes: An AI Perspective. San Rafael, California, USA: Morgan & Claypool Publishers; 2012.
47. Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge: MIT Press; 2018.
48. Melo FS, Veloso M. Heuristic planning for decentralized MDPs with sparse interactions. In: Distributed Autonomous Robotic Systems. Berlin, Heidelberg: Springer; 2013. p. 329–343. https://doi.org/10.1007/978-3-642-32723-0_24.
49. Amato C, Konidaris G, Cruz G, Maynor CA, How JP, Kaelbling LP. Planning for decentralized control of multiple robots under uncertainty. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Seattle, WA, USA: IEEE; 2015. p. 1241–1248. https://doi.org/10.1109/ICRA.2015.7139350.
50. Floriano B, Borges GA, Ferreira H. Planning for decentralized formation flight of UAV fleets in uncertain environments with Dec-POMDP. In: Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS). Atlanta, GA, USA: IEEE; 2019. p. 563–568. https://doi.org/10.1109/ICUAS.2019.8797928.
51. Oliehoek F, Witwicki S, Kaelbling L. Influence-based abstraction for multiagent systems. Toronto, Ontario, Canada: In:

Proceedings of the AAAI Conference on Artificial Intelligence. vol. 26; 2012. p. 1422–1428. https://doi.org/10.1609/aaai.v26i1.8253.

52.• Oliehoek F, Witwicki S, Kaelbling L. A sufficient statistic for influence in structured multiagent environments. J Artif Intell Res. 2021;70:789–870. **This paper formalises influence-based abstraction for decomposing Dec-POMDPs into single-robot models without sacrificing task performance.**

53. Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif Intell. 1999;112(1–2):181–211.

54. Amato C. Decision-making under uncertainty in multi-agent and multi-robot systems: planning and learning. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI); 2018. p. 5662–5666. https://doi.org/10.24963/ijcai.2018/805.

55. Yin Z, Tambe M. Continuous time planning for multiagent teams with temporal constraints. Barcelona, Catalonia, Spain: In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence; 2011. p. 465–471.

56. Costelha H, Lima P. Robot task plan representation by Petri nets; modelling, identification, analysis and execution. Auton Robot. 2012;33(4):337–60.

57. Mansouri M, Lacerda B, Hawes N, Pecora F. Multi-robot planning under uncertain travel times and safety constraints. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI); 2019. p. 478–484. https://doi.org/10.24963/ijcai.2019/68.

58. Azevedo C, Lacerda B, Hawes N, Lima P. Long-run multi-robot planning under uncertain action durations for persistent tasks. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA: IEEE; 2020. p. 4323–4328. https://doi.org/10.1109/IROS45743.2020.9340901.

59. Eisentraut C, Hermanns H, Zhang L. On probabilistic automata in continuous time. In: Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science. Edinburgh, UK: IEEE; 2010. p. 342–351. https://doi.org/10.1109/LICS.2010.41.

60. Buchholz P, Kriege J, Felko I. Input modeling with phase-type distributions and Markov models: theory and applications. Berlin, Heidelberg: Springer; 2014. https://doi.org/10.1007/978-3-319-06674-5.

61. Eisentraut C, Hermanns H, Katoen JP, Zhang L. A semantics for every GSPN. In: Proceedings of the 34th International Conference on Applications and Theory of Petri Nets and Concurrency (Petri Nets). Springer; 2013. p. 90–109. https://doi.org/10.1007/978-3-642-38697-8_6.

62. Hatefi H, Hermanns H. Model checking algorithms for Markov automata. Electron Commun EASST. 2012;53.

63. Alexandros Nikou, Jana Tumova, Dimos V. Dimarogonas. Probabilistic plan synthesis for coupled multi-agent systems. IFAC-PapersOnLine. 2017;50(1):10766–10771. https://doi.org/10.1016/j.ifacol.2017.08.2280.

64. Wongpiromsarn T, Ulusoy A, Belta C, Frazzoli E, Rus D. Incremental synthesis of control policies for heterogeneous multi-agent systems with linear temporal logic specifications. In: Proceedings of the IEEE International Conference on Robotics and Automation. Karlsruhe, Germany: IEEE; 2013. p. 5011–5018. https://doi.org/10.1109/ICRA.2013.6631293.

65. Melcer D, Amato C, Tripakis S. Shield decentralization for safe multi-agent reinforcement learning. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS); 2022. p. 13367–13379.

66. Yang Y, Juntao L, Lingling P. Multi-robot path planning based on a deep reinforcement learning DQN algorithm. CAAI Trans Intell Technol. 2020;5(3):177–83.

67. Dolgov DA, Durfee EH. Resource allocation among agents with MDP-induced preferences. J Artif Intell Res. 2006;27:505–49.

68. De Nijs F, Walraven E, de Weerdt M, Spaan M. Bounding the probability of resource constraint violations in multi-agent MDPs. San Francisco, California, USA: In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31; 2017. p. 3562–2568. https://doi.org/10.1609/aaai.v31i1.11037.

69. de Nijs F, Stuckey PJ. Risk-aware conditional replanning for globally constrained multi-agent sequential decision making. Auckland, New Zealand: In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS); 2020. p. 303–311.

70. Agrawal P, Varakantham P, Yeoh W. Scalable greedy algorithms for task/resource constrained multi-agent stochastic planning. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). AAAI Press; 2016. p. 10-16.

71. Gautier A, Lacerda B, Hawes N, Wooldridge M. Multi-unit auctions allocating chance-constrained resources. Washington DC, USA: In: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI); 2023. p. 11560–11568. https://doi.org/10.1609/aaai.v37i10.26366.

72. Gautier A, Rigter M, Lacerda B, Hawes N, Wooldridge M. Risk constrained planning for multi-agent systems with shared resources. London, UK: In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS); 2023. p. 113–121.

73. Lu S, Zhang K, Chen T, Başar T, Horesh L. Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35; 2021. p. 8767–8775. https://doi.org/10.1609/aaai.v35i10.17062.

74. He S, Wang Y, Han S, Zou S, Miao F. A robust and constrained multiagent reinforcement learning framework for electric vehicle AMoD systems. arXiv preprint arXiv:220908230. 2022.

75. Ganguly KK, Asad M, Sakib K. Decentralized self-adaptation in the presence of partial knowledge with reduced coordination overhead. Int J Inf Technol Comput Sci (IJITCS). 2022;14(1). https://doi.org/10.5815/ijitcs.2022.01.02.

76. Kujirai T, Yokota T. Greedy action selection and pessimistic Q-value updating in multi-agent reinforcement learning with sparse interaction. SICE J Control Meas Syst Integr. 2019;12(3):76–84.

77. Omidshafiei S, Amato C, Liu M, Everett M, How JP, Vian J. Scalable accelerated decentralized multi-robot policy search in continuous observation spaces. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Singapore: IEEE; 2017. p. 863–870. https://doi.org/10.1109/ICRA.2017.7989106.

78. Floriano BR, Borges GA, Ferreira HC, Ishihara JY. Hybrid Dec-POMDP/PID guidance system for formation flight of multiple UAVs. J Intell Rob Syst. 2021;101:1–20.

79. Lauri M, Oliehoek F. Multi-agent active perception with prediction Rewards. Proceedings of the Conference on Neural Information Processing Systems (NeurIPS). 2020;33:13651–13661.

80.•• Rashid T, Samvelyan M, Schroeder C, Farquhar G, Foerster J, Whiteson S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. PMLR; 2018. p. 4295–4304. **This paper presents QMix, a state-of-the-art reinforcement learning approach for multi-robot problems formulated as a Dec-POMDP.**

81. Xiao Y, Lyu X, Amato C. Local advantage actor-critic for robust multi-agent deep reinforcement learning. In: Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems (MRS). Cambridge: IEEE; 2021. p. 155–163.

82. Jiang S, Amato C. Multi-agent reinforcement learning with directed exploration and selective memory reuse. In: Proceedings of the Annual ACM Symposium on Applied Computing; 2021. p. 777–784. https://doi.org/10.1145/3412841.3441953.
83. Lyu X, Amato C. Likelihood quantile networks for coordinating multiagent reinforcement learning. Auckland, New Zealand: In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS); 2020. p. 798–806.
84. Omidshafiei S, Kim DK, Liu M, Tesauro G, Riemer M, Amato C, et al. Learning to teach in cooperative multiagent reinforcement learning. Honolulu, Hawaii, USA In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 6128–6136. https://doi.org/10.1609/aaai.v33i01.33016128.
85. Peng B, Rashid T, Schroeder de Witt C, Kamienny PA, Torr P, Böhmer W, et al. Facmac: factored multi-agent centralised policy gradients. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS). vol. 34; 2021. p. 12208–12221.
86. Pan L, Rashid T, Peng B, Huang L, Whiteson S. Regularized softmax deep multi-agent Q-learning. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS). vol. 34; 2021. p. 1365–1377.
87. Gupta T, Mahajan A, Peng B, Böhmer W, Whiteson S.UneVEn: Universal value exploration for multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. PMLR; 2021. p. 3930–3941.
88. Willemsen D, Coppola M, de Croon GC. MAMBPO: Sample-efficient multi-robot reinforcement learning using learned world models. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic: IEEE; 2021. p. 5635–5640. https://doi.org/10.1109/IROS51168.2021.9635836.
89. Amato C, Konidaris GD, Kaelbling LP. Planning with macro-actions in decentralized POMDPs. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Paris, France: International Foundation for Autonomous Agents and Multiagent Systems; 2014. p. 1273–1280.
90. Amato C, Konidaris G, Anders A, Cruz G, How JP, Kaelbling LP. Policy search for multi-robot coordination under uncertainty. Int J Robot Res. 2016;35(14):1760–78.
91. Hoang TN, Xiao Y, Sivakumar K, Amato C, How JP. Near-Optimal adversarial policy switching for decentralized asynchronous multiagent systems. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD, Australia: IEEE; 2018. p. 6373–6380. https://doi.org/10.1109/ICRA.2018.8460485.
92. Xiao Y, Hoffman J, Amato C. Macro-action-based deep multi-agent reinforcement learning. In: Proceedings of the Conference on Robot Learning. PMLR; 2020. p. 1146–1161.
93. Xiao Y, Hoffman J, Xia T, Amato C. Learning multi-robot decentralized macro-action-based policies via a centralized Q-Net. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Paris, France: IEEE; 2020. p. 10695–10701. https://doi.org/10.1109/ICRA40945.2020.9196684.
94. Liu M, Sivakumar K, Omidshafiei S, Amato C, How JP. Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC: IEEE; 2017. p. 1853–1860. https://doi.org/10.1109/IROS.2017.8206001.
95. Xiao Y, Tan W, Amato C. Asynchronous actor-critic for multi-agent reinforcement learning. In: Proceedings of the Conference on Neural Information Processing Systems (NeurIPS); 2022. p. 4385–4400.
96. Omidshafiei S, Liu SY, Everett M, Lopez BT, Amato C, Liu M, et al. Semantic-level decentralized multi-robot decision-making using probabilistic macro-observations. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Singapore: IEEE; 2017. p. 871–878. https://doi.org/10.1109/ICRA.2017.7989107.
97. Jia S, Wang X, Shen L. A continuous-time markov decision process-based method with application in a pursuit-evasion example. IEEE Trans Syst Man Cybern Syst. 2015;46(9):1215–25.
98. Azevedo C, Matos A, Lima PU, Avendaño J. Petri net toolbox for multi-robot planning under uncertainty. Appl Sci. 2021;11(24):12087.
99. Chen F, Sekiyama K, Huang J, Sun B, Sasaki H, Fukuda T. An assembly strategy scheduling method for human and robot coordinated cell manufacturing. Int J Intell Comput Cybern. 2011;4(4):487–510. https://doi.org/10.1108/17563781111186761.
100. Bellman R. Dynamic programming. Science. 1966;153(3731):34–7.
101. Howard RA. Dynamic programming and Markov processes. New York, USA: Wiley; 1960.
102. Bonet B, Geffner H. Labeled RTDP: Improving the convergence of real-time dynamic programming. In: Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS); 2003. p. 12–21.
103. Kocsis L, Szepesvári C. Bandit based Monte-Carlo planning. In: Proceedings of the European Conference on Machine Learning. Trento, Italy: Springer; 2006. p. 282–293. https://doi.org/10.1007/11871842_29.
104. Kakumanu P. Relation between continuous and discrete time Markovian decision problems. Naval Res Logist Q. 1977;24(3):431–9.
105. Butkova Y, Wimmer R, Hermanns H. Long-run rewards for Markov automata. In: Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Springer; 2017. p. 188–203. https://doi.org/10.1007/978-3-662-54580-5_11.
106. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529–33.
107. Oliehoek FA, Spaan MT, Vlassis N. Optimal and approximate Q-value functions for decentralized POMDPs. Journal of Artificial Intelligence Research. 2008;32:289–353.
108. Buchholz P, Hahn EM, Hermanns H, Zhang L. Model checking algorithms for CTMDPs. In: Proceedings of the International Conference on Computer Aided Verification (CAV). Springer; 2011. p. 225–242. https://doi.org/10.1007/978-3-642-22110-1_19.
109. Legay A, Delahaye B, Bensalem S. Statistical model checking: an overview. In: Proceedings of the International Conference on Runtime Verification; 2010. p. 122–135. https://doi.org/10.1007/978-3-642-16612-9_11.
110. Butkova Y, Hartmanns A, Hermanns H. A modest approach to Markov automata. ACM Trans Model Comput Simul (TOMACS). 2021;31(3):1–34.
111. Koenig N, Howard A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sendai, Japan; 2004. p. 2149–2154. https://doi.org/10.1109/IROS.2004.1389727.
112. Bakker T, Ward GL, Patibandla ST, Klenke RH. RAMS: a fast, low-fidelity, multiple agent discrete-event simulator. Toronto, Ontario, Canada: In Proceedings of the Summer Computer Simulation Conference (SCSC); 2013. p. 1–10.