Literature Review - Informal

Aamal Hussain

October 30, 2019

Introduction

Note: This review is a work in progress. Please expect an update on the Game Theoretic, Hard Coded and Multi Agent Dynamics sections of this review within the week. Verification in swarms may take a little more time for me to catch up with.

The areas of interest regarding Multi Agent Systems fall into two distinct categories (which have some slight overlap). The first is denoted as 'Decision Making', though the terms 'Planning' and 'Distributed Control' may sometimes be used as substitutes. It is important to note that, in my case, I do not include Planning Formalisation techniques such as PDDL. Instead, I focus on the particular methods which involve interaction in the world. The second, I will refer to as 'Multi Agent Dynamics', though sometimes I will refer to this as 'Stability Analysis'.

Scope

Both of the aforementioned topics have been applied to all variants of multi agent systems and have sufficient room for further exploration. The typical variants of multi agent systems that I will consider are

- Swarms
- Multi Agent Reinforcement Learning (MARL)
- Decentralised Partially Observed Markov Decision Processes (Dec-POMDP)
- Game Theoretic Approaches
- Hard Coded

In the above, 'Game Theoretic Approaches' cover a wide spectrum, including zero-sum (minimax) games, bayesian games, and many more. The final category, 'Hard Coded', refers to any approach which does not fit neatly into any of these categories. The name is chosen since they often refer to methods which revolve around a series of if-else statements.

I will first consider Decision Making as this is most closely related to the research proposal, before I consider Multi Agent Dynamics, which has strong implications for Safe and Trusted Multi Agent Systems.

Objectives

The aim of the following sections is not to provide an exhaustive list of all work done in the aforementioned areas. To attempt to do this would be an exercise in futility. Instead, it is to identify research directions which lie within the broad scope of Multi-Agent Systems (MAS). With these directions, we may be able to narrow down the remaining literature review and hone in on particular problems and, in fact, we may find that we address multiple of these over the next four years (while we, of course add more). As such, the end of each chapter will provide a list of research directions which I have identified from the preceding review.

Chapter 1

Decision Making

Decision making refers to the generalised problem of considering how a multi-agent system should interact in the world to achieve their goals. This subsumes both the cases where the agents' goals are aligned (cooperative) or in conflict with one another (competitive).

The following sections will provide an overview of the literature aimed towards coordinating such systems as followed by interesting sub-fields in each approach which present avenues for research related to Safe and Trusted AI (STAI). These, of course, are not exhaustive and more will be added as the review progresses.

1.1 Swarms

Swarm based systems comprise of multiple homogeneous agents which are able to organise themselves through a formation using a series of simple local interactions with their neighbours [1]. Whilst the individual agents are generally simplistic, the collective behaviour may exhibit complex phenomena emulating systems observed in biological organisations such as bee or ant colonies [2]. Hybrid algorithms such as in [3] show an accelerated performance in reaching globally optimal solutions in search-based tasks. The advantage of many swarm algorithms is that they are based on local interactions and so are incredibly scalable [4].

The reduction of the complexity in interactions between agents also allows for the robots to perform other calculations on board. In [5], Pini et al. leverage this by considering adaptive task partitioning across swarms. This allows a swarm, in a decentralised manner, to deliberate whether to partition a task into its sub-tasks or to perform the task in its whole. As of now (to the best of my knowledge) the problem of partitioning general tasks into its N sub-tasks is unexplored. This, however, highlights another advantage of swarm systems; they are readily divided into sub-groups (as in [6]) to perform a divide-et-impera (divide and conquer) approach to solving problems [5]).

Furthermore, swarm systems may be designed in a leaderless manner and so do not require the use of a central controller [1]. This presents the advantage that the system can rapidly adapt to the loss of agents or separation of groups throughout the task. However, the assumptions made regarding the homogeneity of individual agents and the simplicity of their local interactions result in significant limitations placed on the complexity of the tasks that swarm systems can accomplish.

Co-evolution and Self-Healing

Recently, there has been an increased interest in introducing heterogeneity into the swarm systems to improve their real world applicability. An example of this which have been shown strong real world success can be found in [7]. Here, the authors consider two swarm teams, referred to as 'foot bots' and 'eye bots' who work in unison to explore an environment and solve a navigation task.

This area of research is sparsely populated and warrants further exploration. This is since the use of co-evolutionary teams can improve the robustness of the swarm optimisation. This is since it will be possible for a team to automatically determine when robots in the other team are not exhibiting expected behaviour and ensure that the other team self corrects. This process is referred to as 'Self-Healing' in [8]. Here, the authors allow a user to define the goals of a swarm system. From this, a 'trust' metric can be defined which measures the deviation of each agent from the expected behaviour. The self-healing process occurs by limiting communication of all agents with 'untrusted' agents and encouraging communication with 'trusted' agents. The unison of self-healing with co-evolution may present the opportunity for heterogenous swarms to maintain their evolutionary stability, even in the face of environmental disturbances.

Fault Detection

The above sections have considered the fact that swarm systems are robust to losses in the group. However, any MAS system must first be able to recognise that an agent has undergone some failure.

To this end, Tarapore et al. [9] develop a robust fault detection approach in which the swarm itself, in a decentralised manner, is capable of assessing deviation from 'normal' behaviour, even when the behaviour of the swarm itself is altered (perhaps by a remote operator). The authors achieve this by requiring that the agents themselves sense and characterise their own behaviour. This characterisation is formulated as a binary feature vector which is then communicated to the agent's neighbours. These neighbours will reach a consensus over whether the agent should be treated as faulty based on their collective behaviour. The results presented in [9] show extremely promising results and suggest that their method is, in fact, able to determine faults with high accuracy in the presence of various fault types (including sensor and actuator failures), although poor performance is seen in actuation failures in some instances. It should be noted that this method requires that each robot transmit their feature vector to the nearest neighbours. In environments where communication may be severely limited, this may present further errors. Furthermore, it is unlikely that, when a robot is damaged, only one of its components will be affected. Therefore, it is important to determine the effect on performance in the face of multiple agent failures and in communication losses. This exploration may open the possibility of improving the state of the art in terms of failure detection in swarm systems. (Of course, this conclusion is based off two papers so further review is required).

Verification

Both of Alessio's papers [10, 11] fit in here but they require further reading

1.1.1 Directions

From the above consideration of swarms, the following research directions have been identified which, in my view, concern themselves with STAI.

- Healing through co-evolution: The application of heterogenous robot swarms towards ensuring that emergent phenomenon and swarm behaviour are as expected by the user.
- Fault Detection in limited communication: Considering the ability of a swarm to, in a decentralised approach, consider which robots in the team have failed, even in cases of no communication or multiple failures.

1.2 Dec-POMDPs

The use of POMDPs in multi-agent settings is formalised as decentralised POMDP (Dec-POMDP) which aims for a team of agents to maximise a common utility. However, it has been found that determining the exact solution to Dec-POMDP problems is NEXP [12] and so is intractable for all but toy problems. A number of methods have been presented to attempt to solve Dec-POMDPs. Oliehoek gives a review of these in [13]. Most solutions (such as brute force) are intractable for all but toy problems.

Approximate solutions to Dec-POMDP have been proposed, perhaps most notable of which is the proposal of MacDec-POMDP [14] by Amato et al. Here, macro actions (actions which extend over multiple time steps) are used, as opposed to low-level actions which are re-evaluated at each time step. This allows an exact solution to be found as it does not need to be evaluated at each time step. This method assumes that, once macro-actions are distributed, the policies (sequence of state-action pairs) are known. Since this is not the case, Amato also proposes the use of a Dec-POSMDP [15], where 'SMDP' refers to 'Semi-Markov Decision Process, in which a high level model is defined without the underlying Dec-POMDP's actions and observations.

However, this is largely applicable in passive settings where common payoffs can be determined by an offline planner. They also require a significant amount of data with which to allow the system to learn the underlying models and payoff structures. This limits the applicability of the system when communication is limited and the system is presented with environments that it has not seen before. Recent work in MDPs [16] has considered learning in the face of Significant Rare Events (SREs) which the system has not yet observed. Currently, it is required that a model of such SREs is known and so it would be interesting to consider the application of Dec-POMDPs in situations where the SRE model is incomplete or erroneous and assess the robustness of the Dec-POMDP framework against such events.

1.2.1 Directions

The area of Dec-POMDPs still requires more review from me, especially for solutions which are not developed by Amato as he seems to largely dominate the field. Based on this initial review of the area, a potential direction for research is

- Significant Rare Events: Consider Dec-POMDP capability to remain robust to SREs which have a limited or erroneous model.
- Agent Failures: The Dec-POMDP model uses a centralised planner which acts offline. It therefore assumes that the agents will be able to carry out their assigned tasks. It would be interesting to examine the possibility of, either adaptive planning, or planning with contingencies in the Dec-POMDP framework.

1.3 Game Theoretic Approaches

Game theoretic models are generally the go-to method for understanding multi-agent systems. As such they fit into all of the categories in this chapter (except, perhaps, swarms) since Dec-POMDP and MARL methods have both used game theory to support their frameworks. In fact, Dec-POMDP is a subset of Partially Observed Stochastic Games (POSG), in which all agents use the same payoff. Game theory can, therefore, be used in an applied capacity to direct task allocation across heterogenous teams.

Market Based Methods

Stochastic Games

1.4 Hard Coded

1.5 MARL

Reinforcement learning extends the Markov Decision Process problem by considering the case where the payoff model is not known. This, of course, is the case for most real world environments. As such, MARL algorithms can perhaps be considered to be more applicable than Dec-POMDP models. Fortunately, MARL has picked up a lot of traction in research recently, with a large body dedicated towards solving the many problems it presents.

The largest problem in MARL is the non-stationarity of the environment [17]. In single-agent settings, it is assumed that the environment is Markovian. However, this must be lifted in the Multi Agent setting since other agents in the environment will be learning concurrently. This learning will be based on their own history of interactions which extend beyond the previous state. As such, we must now consider that the policy for any one agent will depend on the policy of all other agents. As such, a big concern in this area is regarding convergence guarantees and the stability of the learner system. Approaches to this will be discussed in the next chapter.

Agent Modelling

Returning to the problem of non-stationarity, solutions have been presented in which the agent models the learning of other agents. A noteworthy example of this is found in [18] in which the agent performs a one-step lookahead of the other agents' learning and optimises with respect to this expected return. They show that this leads to stable learning and can even lead to emergent cooperation from competition. However, the method requires that both agents have exact knowledge of the others' value functions in order to perform the one step lookahead. Furthermore, it has only been considered for the case of a two agent adversarial game and so the scalability of the system to multiple agents is not yet understood. Another method presented by Mao et al. [19] uses a centralised critic to collect the actions and observations of all agents and allows it to model the joint policy of teammates. This is shown to generate cooperative behaviour across four agents and so is more applicable to real world settings. However, its disadvantage over the method presented in [18] is that the critic is centralised. In real world settings, this requires the presence of an agent (perhaps a laptop) which is able to handle the computational load of determining a joint policy across all agents and must then communicate the Q-values of all agents back to them. This is both a taxing both in terms of computation and time.

Hong et al [20] present a similar system for modelling teammate policies by tasking a CNN with determining the policy features of other agents and then embedding these as features in its own DQN. This shows strong performance in settings where other agents dynamically change their policies. The concerns with this, however, are that, as the number of agents in the field increase, the CNN in each agent must perform another approximation. This places strong requirements on the performance of the CNN since errors in estimation will accumulate as the number of agents increases. Similarly, the complexity of the DQN will increase as more feature vectors are added.

Finally, all of the above methods are not robust to evolving numbers of agents. The problem of agent modelling is an important one to ensure stable learning and to understand the evolution of the system. It also presents a strong challenge and is open to exploration. To put it in context the methods described in this section are all from 2018-19, so its all very new.

1.5.1 Directions

I still have a lot of reading to do regarding MARL, which, in turn, will identify new directions. However, on initial assessment I put forward

- Modelling evolving teammates: The purpose of this is to more strictly ensure the stability of the learning process. However, the particular problem I suggest is to consider the modelling in a decentralised manner and with the consideration of evolving numbers of agents in teams.

Chapter 2

Multi Agent Dynamics

Multi Agent Dynamics considers the problem of mathematically modelling learning in Multi Agent Systems (MAS). This model then serves to be able to predict the evolution of a learning system as well as to understand its stability. Stability may be looked at from the view point of two perspectives. The first is from a parameter tuning point of view. This considers modelling learning with respect to optimisation parameters, allowing us to better choose our parameters so that they may converge to an optimal result. The second is from the view point of the state-action space of a learnt model. This allows us to determine, before the MAS is deployed, which set of state-action pairs will lead to unstable behaviour. This knowledge allows us to consider which state-action pairs should be avoided. In both cases, stability analysis allows us to build multi agent systems which will learn and act in the way that we expect them to.

The following papers [21, 22, 23, 24, 25, 26, 27] are the ones that I found particularly relevant to this study. However, they will require some further analysis before I write about them here.

Bibliography

[1] M. S. Couceiro, R. P. Rocha, and F. M. L. Martins, "Towards a predictive model of an evolutionary swarm robotics algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 5 2015, pp. 2090–2096. [Online]. Available: http://ieeexplore.ieee.org/document/7257142/

[2] D. Sethi and A. Singhal, "Comparative analysis of a recommender system based on ant colony optimization and artificial bee colony optimization algorithms," in *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*. Institute of Electrical and Electronics Engineers Inc., 12 2017.

[3] Y. Gao, "An improved hybrid group intelligent algorithm based on artificial bee colony and particle swarm optimization," in *Proceedings - 2018 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2018*. Institute of Electrical and Electronics Engineers Inc., 11 2018, pp. 160–163.

[4] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision Making in Multiagent Systems: A Survey," pp. 514–529, 9 2018.

[5] G. Pini, A. Brutschy, M. Frison, A. Roli, M. Dorigo, M. Birattari, G. Pini, A. Brutschy, M. Dorigo, M. Birattari, A. Brutschy, M. Dorigo, M. Birattari, M. Frison, A. Roli, and A. Roli, "Task partitioning in swarms of robots: an adaptive method for strategy selection," *Swarm Intell*, vol. 5, pp. 283–304, 2011.

[6] P. Zahadat and T. Schmickl, "Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition," *Adaptive Behavior*, vol. 24, no. 2, pp. 87–101, 2016.

[7] F. Ducatelle, G. A. D. Caro, C. Pinciroli, Luca, M. Gambardella, F. Ducatelle, . Dalle, G. A. D. Caro, C. Pinciroli, and L. M. Gambardella, "Self-organized Cooperation between Robotic Swarms," Tech. Rep. [Online]. Available: http://www.swarmanoid.org

[8] R. Liu, F. Jia, W. Luo, M. Chandarana, C. Nam, M. Lewis, and K. Sycara, "Trust-Aware Behavior Reflection for Robot Swarm Self-Healing *," Tech. Rep. [Online]. Available: www.ifaamas.org

[9] D. Tarapore, J. Timmis, and A. L. Christensen, "Fault Detection in a Swarm of Physical Robots Based on Behavioral Outlier Detection," *IEEE Transactions on Robotics*, pp. 1–7, 8 2019.

[10] P. Kouvaros, A. Lomuscio, E. Pirovano, and H. Punchihewa, "Formal Verification of Open Multi-Agent Systems," Tech. Rep., 2019. [Online]. Available: www.ifaamas.org

[11] A. Lomuscio and E. Pirovano, "A Counter Abstraction Tech-nique for the Verification of Probabilistic Swarm Systems," Tech. Rep., 2019. [Online]. Available: www.ifaamas.org

[12] B. Eker, E. Ozkucur, C. Mericli, T. Mericli, and H. L. Akin, "A finite horizon DEC-POMDP approach to multi-robot task learning," in *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 10 2011, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/6111001/

[13] F. A. Oliehoek, "Decentralized POMDPs," Tech. Rep.

[14] C. Amato, G. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for decentralized control of multiple robots under uncertainty," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2015, pp. 1241–1248. [Online]. Available: http://ieeexplore.ieee.org/document/7139350/

[15] C. Amato, "Decision-Making Under Uncertainty in Multi-Agent and Multi-Robot Systems: Planning and Learning," Tech. Rep., 2017. [Online]. Available: https://youtu.be/34xHxXrnPHw,

[16] R. Klima, D. Bloembergen, M. Kaisers, and K. Tuyls, "Robust Temporal Difference Learning for Critical Domains," Tech. Rep., 2019. [Online]. Available: www.ifaamas.org

[17] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A Survey and Critique of Multiagent Deep Reinforcement Learning $," Tech. Rep.

[18] J. Foerster, R. Y. Chen, O. Maruan Al-Shedivat, S. Whiteson, P. Abbeel, I. Mordatch OpenAI, and M. Al-Shedivat, "Learning with Opponent-Learning Awareness," Tech. Rep., 2018. [Online]. Available: www.ifaamas.org

[19] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Z. . Gong, "Modelling the Dynamic Joint Policy of Teammates with Attention Multi-agent DDPG," Tech. Rep. [Online]. Available: www.ifaamas.org

[20] Z.-W. Hong, S.-Y. Su, T.-Y. Shann, Y.-H. Chang, and C.-Y. Lee, "A Deep Policy Inference Q-Network for Multi-Agent Systems," Tech. Rep., 2018. [Online]. Available: www.ifaamas.org

[21] J. P. Bailey, G. Gidel Mila, and G. Piliouras, "Finite Regret and Cycles with Fixed Step-Size via Alternating Gradient Descent-Ascent," Tech. Rep., 2019.

[22] J. P. Bailey and G. Piliouras, "Multi-Agent Learning in Net-work Zero-Sum Games is a Hamiltonian System," Tech. Rep., 2019. [Online]. Available: www.ifaamas.org

[23] V. Boone, G. Piliouras, and E. De Lyon, "From Darwin to Poincaré and von Neumann: Recurrence and Cycles in Evolutionary and Algorithmic Game Theory," Tech. Rep., 2019.

[24] L. Dickens, K. Broda, and A. Russo, "The Dynamics of Multi-Agent Reinforcement Learning," Tech. Rep.

[25] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe Model-based Reinforcement Learning with Stability Guarantees," Tech. Rep.

[26] M. Jin and J. Lavaei, "STABILITY-CERTIFIED REINFORCEMENT LEARNING: A CONTROL-THEORETIC PERSPECTIVE *," Tech. Rep.

[27] A. Letcher, D. Balduzzi, S. Racanì, J. Martens, J. Foerster, K. Tuyls, and T. Graepel, "Differentiable Game Mechanics," Tech. Rep., 2019. [Online]. Available: https://github.com/deepmind/symplectic-gradient-adjustment.