Reinforcement learning extends the Markov Decision Process problem by considering the case where the reward model is not initially known to the agent. In a similar manner, Multi Agent Reinforcement Learning (MARL) extends the Markov Game setting to one where the payoff structure is not a priori knowledge.

The task of MARL is to determine an optimal joint policy for all agents across the game. This joint policy may be the concatenation of all the individual policy or it may just be options for each agent to take. In either case, optimality is defined through the standard notions of Nash equilibria and so, in this section, I will try to consider the broad spectrum of methods which attempt to achieve this Nash equilibria. The largest problem in MARL is the non-stationarity of the environment [**?**]. 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. As such, we must now consider that the policy for any one agent will depend on the policy of all other agents.

This chapter begins with a selection of the foundational methods which were developed towards solving the MARL problem. The interested reader may find a stronger review and analysis in [**?**]. I then go on to consider more contemporary approaches, most notably that of agent modelling.

0.1 Learning in Two Player Matrix Games

The most fundamental method to learning in Matrix games is the simplex algorithm. This is a popular method of linear programming (in which constraints are linear). This will be important in considering more current methods. A similar consideration is given to the infinitesimal gradient ascent algorithm, in which the step size converges to zero. This method guarantees that, in the infite horizon limit, the payoffs will converge to the Nash equilibrium payoff. Note that this does not necessarily mean that both agents will converge to a single Nash equilibrium. This is a particular problem in games where there are multiple Nash equilibria. However, in practice it is difficult to choose a convergence rate of the step size and, without an appropriate choice the strategy may oscillate as shown in the book. To address this, a modified approach is presented by Bowling and Veloso which incorporates the notion of Win or Learn Fast (WoLF) to produce WoLF-IGA. WoLF is a notion we will come across often in MARL and is shown by the authors to converge to always to a NE. The concern with WoLF methods, however, is that it requires explicit knowledge of the payoff matrix (which is not so much of a problem for model based methods) and the opponent's strategy (which is more of a problem in real-world methods). Finally, the Policy Hill Climbing method (PHC) is shown to converge to an optimal mixed strategy if the other agents are stationary (i.e. are not learning). However, it is shown that, when this is not the case, the algorithm again oscillates. The WoLF-PHC adaptation of this method is shown to converge to a NE strategy for both players with minimal oscillation.

0.2 Learning in Multiplayer Stochastic Games

Stochastic Games (or Markov Games) form a basis for MARL settings. However, in this case the agents must learn about the equilibrium strategies by playing the game, which means they do not have a priori knowledge of the reward or transition functions. Schwarz considers two properties which should be used for evaluating MARL algorithm: rationality and convergence. The latter simply states that the method should converge to some equilibrium whereas the former suggests that the method should learn the best response to stationary opponents. A similar set of conditions is considered by Conitzer and Sandholm in [**?**], whose algorithm we will consider shortly. Schwarz then presents a review of MARL methods (as of September 2014)

TODO: Consider the case for partially observed/decentralised settings. Find the relevant material within [**?**] which addresses this issue.

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 [**?**] 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. [?] 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 [?] 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 [?] 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.

The above methods are all centralised techniques, in which a controller determines the optimal joint policy for both agents. However, in real scenarios it is often preferable that each agent learns their own strategy, a task which must be completed without information of the other agent's strategy. The methods presented towards this problem are: linear reward-inaction (L_{R-I}) which guarantees convergence to NEs in games which contain pure NEs, linear reward-penalty (L_{R-P}) which can guarantee convergence to mixed strategies given the appropriate parameters, lagging anchor algorithm which also converges to mixed strategies, and the author's own proposal of the L_{R-I} lagging anchor algorithm which can converge to both pure and mixed NEs.

0.3 Game Dynamics

Game Dynamics (which I often refer to as Multi Agent Dynamics or Learning Dynamics) considers the problem of mathematically modelling Multi Agent Systems who adapt through repeated interact with one another. This model then serves to be able to predict the evolution of the system as well as to understand the trajectory of learning. Typically, this looks at considering whether or not the method is likely to converge towards a Nash equilibrium. This is generally a difficult problem to solve [?] for all but toy problems. [?] shows that the stable equilbrium and Nash equilibrium (NE) are not necessarily the same and, in fact, argue that stable points are more informative than NEs. Stability provides some guarantees against the stochastic nature of the environment since a stable equilibrium will always be returned to even after perturbations. This extremely important in Safe and Trusted AI as it provides guarantees against undesired behaviour in real world environments.

The area of dynamics which has shown most promise in Multi-Agent Reinforcement Learning is that of evolutionary dynamics. This draws from the principles of Evolutionary Game Theory (EGT) which considers similar assumptions to that of MARL: agents are no longer required to be rational and play the game optimise their expected return through repeated play. Importantly, players have no knowledge of the others' payoffs [?]. The first major breakthrough in this field is in [?] where Boergers and Sarin determine the relation between the replicator dynamics concept of EGT (a differential equation defining the evolution of the proportion of a subgroup in an evolving population) and the 'Cross' learning method for multi agent reinforcement learning. This idea is advanced in [?] where Tuyls et al. determine a similar relation for Q-Learning using Boltzmann probabilities as Q-values. The result was a dynamics equation which describes, for each action, the evolution of its selection probability which could even account for random exploration. There have since been a number of works which apply the same insight into different game types and MARL algorithms. In [?], these are broken into the following categories

- Stateless games with discrete actions. Here, stateless refers to the idea that the game is static and so the environment has no impact on the result. The aforementioned result [?] fits into this category.

- Stateless games with continuous actions. These consider more realistic MARL than the previous category by replacing each agent's strategy vector with a probability density function (pdf) over a continuous action space.

- Stateful games with discrete actions. This mostly considers stochastic games, where there are multiple states with probabilistic (usually Markovian) transitions between them. However, extensive form games, which considers more complex phenomena such as sequential moves and imperfect information, are also briefly mentioned.
- Stateful games with continuous actions. This is one of the more realistic assumptions considered. However, the authors point out that this area is yet to see results, leaving it open for possible research.

Though the above have seen success through experimental validation, most work is applied to bimatrix games (games of two to three players with finite action spaces). An example of this can be found in [?] in which Hofbauer shows that learning a zero-sum bimatrix game follows Hamiltonian dynamics and results in stable equilibria. Recent work has begun the consideration of improving the models towards more complex scenarios with larger agent populations. For instance, Bailey and Piliouras show in [?] that the results of [?] can be extended to the n-agent case, even with different network structures. They show that the result generalises to any zero-sum network game where agents use a Follow-The-Regularised-Leader learning approach. Another, more recent example of this is [?], in which Hu uses a mean field approximation to model the dynamics of a population of Q-Learning agents. As a reminder, mean field (MF) approaches in MARL consider that an agent updates its strategy based on the mean effect of the population. The result is a system of three equations which describes the evolution of Q-values over a large population in a symmetric bi-matrix game.This presents an important first step in modelling the learning dynamics of large agent populations and has the scope to be expanded to systems of asymmetric games, heterogeneous populations and stateful n-agent games.

An advantage of determining the evolutionary dynamics of learning is that it can describe the expected behaviour in learning different game settings. This is particularly important to understand the convergence of the applied learning method; certain games will often show cyclic behaviour even with the existence of a strict NE. For instance, Imhof et al. show in [?] that a repeated prisoner's dilemma game results in cyclic behaviour when considering the options of cooperation or defection. In an interesting twist, Galla shows in []

0.4 Stable Learning

This section considers a dynamical systems approach towards learning agents. Its aim is to develop learning systems which prioritise stability.

Stability may be looked at from the view point of two perspectives. The first is from an optimisation point of view. This considers the dynamics of the learning model, allowing us to better choose our parameters and design our models so that they may converge to a stable 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.

In [?], Letcher et al. model gradient descent learning of generative-adversarial-networks (GANs) as a two-player differentiable game. A differentiable game is one in which the loss function is twice differentiable. Using this formulation, they are able to analyse the system from new perspectives by considering the current state-of-the-art understanding of differentiable game theory. Whilst, at first glance, this may seem like a purely theoretical exercise, they go on to show that the insights gained allow them to develop a new multi-objective optimisation technique for GANs which shows stronger convergence properties, most notably of which is that it guarantees that the method finds a stable equilibrium (and avoids saddles) between the two players' loss functions.

Jin and Lavaei [?] consider the policy of a reinforcement learning agent as a non-linear, time varying feedback controller. Using this notion, they then consider the bounded-input-bounded-output stability of the system. They do this by analysing the ratio between the total output and total input energy (called the L2 gain). If the L2 gain remains finite then the system may be considered to be stable. They then apply these considerations on real-world applications including multi-agent flight formation and obtain stability certificates (essentially confirming that the system will remain stable under certain conditions) for the learned controller.

Berkenkamp et al. [?] consider a similar problem from a different definition of stability. Specifically, they look at stability from the point of view of Lyapunov functions. A system is said to be stable if the applying the policy will result in stricly lower evaluations of the function. In other words, a system is stable if its corresponding Lyapunov function is decreasing towards a minimum point. The authors use this idea to define a 'region of attraction' in which the system is stable in the sense of Lyapunov. The goal of Safe Lyapunov Learning, a method which they develop from these insights, is to learn a policy without leaving this region of attraction. They do this by taking measurements within the set and using this to learn about the system dynamics, thereby increasing the safe set. With this, we now have a guarantee that policy optimisation will not result in unsafe behaviour, even in the presence of stochasticity and exploration.

As we can see from the discussion thus far, there are many definitions and perpectives of stability, each lending to a new understanding of learning systems. To add another to the mix, Milchtaich in [**?**], presents a notion of static stability. This means that it is based solely on the incentives of the players and does not require a consideration of the dynamics of the system. Milchtaich's definition of a stable system is one in which, when perturbed, it is more beneficial for an agent to move back towards the equilibrium than it is for them to move away. This is perhaps the most fundamental definition of a stable equilibrium and, as such, Milchtaich shows that it is applicable to all strategic games, within certain assumptions (finite set of player and continuous strategy space) and considers probabilistic perturbations from the original state. This is particular applicable to Multi Robot Reinforcement Learning (or any MARL in continuous strategy spaces) since these systems require random exploration of the strategy space and the dynamics are not known a priori. However, the lack of dynamics means that we do not consider the evolution of learning.

TODO: In the above, look specifically at the assumptions made within the papers. In the multi-agent case, the biggest assumption is that of independent learning. Perhaps consider how advances in coupled dynamical systems may be used to consider learning agents whose actions affect one another.