SYDNEY

Business School
The University of Sydney

OME WORKING PAPER SERIES

Convergent learning algorithms for potential games with unknown noisy rewards

Archie C. Chapman
Business School
The University of Sydney

David S. Leslie
School of Mathematics
The University of Bristol

Alex Rogers
Electronics and Computer Science
The University of Southampton

Nicholas R. Jennings
Electronics and Computer Science
The University of Southampton

Abstract

In this paper, we address the problem of convergence to Nash equilibria in games with rewards that are initially unknown and which must be estimated over time from noisy observations. These games arise in many real–world applications, whenever rewards for actions cannot be prespecified and must be learned on–line. Standard results in game theory, however, do not consider such settings. Specifically, using results from stochastic approximation and differential inclusions, we prove the convergence of variants of fictitious play and adaptive play to Nash equilibria in potential games and weakly acyclic games, respectively. These variants all use a multi–agent version of Q–learning to estimate the reward functions and a novel form of the e–greedy decision rule to select an action. Furthermore, we derive e–greedy decision rules that exploit the sparse interaction structure encoded in two compact graphical representations of games, known as graphical and hypergraphical normal form, to improve the convergence rate of the learning algorithms. The structure captured in these representations naturally occurs in many distributed optimisation and control applications. Finally, we demonstrate the efficacy of the algorithms in a simulated ad hoc wireless sensor network management problem.

August 2011

OME Working Paper No: 05/2011
http://www.econ.usyd.edu.au/ome/research/working_papers

CONVERGENT LEARNING ALGORITHMS FOR POTENTIAL GAMES WITH UNKNOWN NOISY REWARDS

ARCHIE C. CHAPMAN[*], DAVID S. LESLIE[†], ALEX ROGERS[‡], AND NICHOLAS R. JENNINGS[§]

Abstract. In this paper, we address the problem of convergence to Nash equilibria in games with rewards that are initially unknown and which must be estimated over time from noisy observations. These games arise in many real–world applications, whenever rewards for actions cannot be prespecified and must be learned on–line. Standard results in game theory, however, do not consider such settings. Specifically, using results from stochastic approximation and differential inclusions, we prove the convergence of variants of fictitious play and adaptive play to Nash equilibria in potential games and weakly acyclic games, respectively. These variants all use a multi–agent version of *Q*–learning to estimate the reward functions and a novel form of the ε–greedy decision rule to select an action. Furthermore, we derive ε–greedy decision rules that exploit the sparse interaction structure encoded in two compact graphical representations of games, known as graphical and hypergraphical normal form, to improve the convergence rate of the learning algorithms. The structure captured in these representations naturally occurs in many distributed optimisation and control applications. Finally, we demonstrate the efficacy of the algorithms in a simulated ad hoc wireless sensor network management problem.

Key words. Game theory, distributed optimisation, learning in games.

AMS subject classifications. 91A10, 68T05, 68W15

1. Introduction.

The design and control of large, distributed systems is a major engineering challenge. In particular, in many scenarios, centralised control algorithms are not applicable, because limits on the system's computational and communication resources make it impossible for a central authority to have complete knowledge of the environment and direct communication with all of the components of the system. In response to these constraints, researchers have focused on decentralised control mechanisms for such systems.

In this context, a class of non–cooperative games called *potential games* [Monderer and Shapley, 1996b] have gained prominence as a design template for decentralised control in the distributed optimisation and multi–agent systems research communities. Potential games have long been used to model congestion problems on networks [Wardrop, 1952; Rosenthal, 1973]. However, more recently, they have been used to design decentralised methods of solving large–scale distributed problems, such as power control, channel selection and scheduling problems in ad hoc wireless networks [Scutari et al., 2006], target assignment problems [Arslan et al., 2007], task allocation and scheduling problems [Marden and Wierman, 2008; Chapman et al., 2010] and distributed constraint optimisation problems [Chapman et al., ress]. In more detail, a potential game is constructed from a global target function by distributing the system's control variables among a set of *agents* (or players). Each agent's reward function is derived so that it is *aligned* with the system–wide goals. That is, an increase in an agent's reward corresponds to an increase in the global reward (as in Wolpert and Tumer [2002]). If the agents' rewards are aligned with the global target function, then the global target function is a *potential* for the game, and the game is a potential game. This, in turn, implies that the (pure) Nash equilibria of the constructed game correspond to the local optima of the potential function. This is a very useful property of potential games, because a Nash equilibrium is an

[*] Business Analytics and Operations, University of Sydney Business School, Sydney, NSW 2006, Australia (a.chapman@econ.usyd.edu.au).

[†] School of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, United Kingdom (david.leslie@bristol.ac.uk).

[‡] School of Electronics and Computer Science, University of Southampton, Highfield, Southampton, SO17 1BJ, United Kingdom (acr@ecs.soton.ac.uk).

[§] School of Electronics and Computer Science, University of Southampton, Highfield, Southampton, SO17 1BJ, United Kingdom (nrj@ecs.soton.ac.uk).

action profiles that is robust to unilateral changes in agents' strategies; and as a consequence, the local optima of the potential function are stable in these games. One useful way to align the agents rewards is to set each agent's reward to the marginal contribution it makes to the global target function. This often reduces the coupling between agents' reward functions, which facilitates significant reductions in the communication and computation requirements facing each agent. Furthermore, under the common assumption that the global target function is submodular in the agents' contributions (i.e. each agent has a decreasing marginal contribution), then the ratio of the worst–case Nash equilibrium to the optimum can be bounded. This ratio is known as the *price of anarchy*, and Marden and Wierman [2008] show that in submodular marginal contribution games it is at most $1/2$ (i.e. the value of the worst Nash equilibrium solution (local maximum) is within $1/2$ of that of the global optimum). This is effectively a bound on the cost of distributing the control of the problem among multiple autonomous agents.

Given this framework for distributing an optimisation problem, the second problem a designer of a decentralised optimisation method faces is specifying a distributed algorithm for finding a Nash equilibrium. This is addressed by the literature on *learning in games*; the dynamics of learning processes in repeated games is a well investigated branch of game theory (see Fudenberg and Levine [1998], for example). In particular, the results that are relevant to this work are the guaranteed convergence of *fictitious play* and *adaptive play*, and their generalisations, to Nash equilibrium in potential games [Monderer and Shapley, 1996a; Young, 1998; Leslie and Collins, 2006]. Using these algorithms, a decentralised method for an optimisation problem can be found by, first, constructing a potential game from the optimisation problem (via the method described above), and then using fictitious play or adaptive play to compute a Nash equilibrium.

There is, however, one major shortcoming to this model. As is standard in game theory, there is an assumption that the value of each configuration of variables, or the agents' rewards for different joint strategy profiles, is known from the outset. Although this is a sound assumption in some domains, in many of the large, distributed control application domains to which the decentralised control methods described above are targeted, it is not realistic to assume that the rewards for different variable configurations can be prespecified. For example, in many situational awareness problems, the system's task is to learn about the phenomena under observation, but the rewards earned by the agents in the system are a function of the phenomena detected, so their rewards cannot be known before they are deployed.

Thus, against this background, in this paper, we address the problem of distributed computation of Nash equilibria in games with rewards that are initially unknown and which must be estimated on–line from noisy observations. The primary objective is to derive learning algorithms that provide convergence to both the true mean values of the reward functions and to the Nash equilibria of the one–shot game in those true mean rewards. Now, as noted above, learning in repeated games is well understood. Similarly, online learning of unknown noisy reward functions is a well understood problem tackled effectively by techniques from reinforcement learning, and, in particular, here we consider Q–learning [Sutton and Barto, 1998]. However, the joint problem of learning the equilibria of games with unknown noisy reward functions is less well understood, even though it is a commonly faced problem in real–world applications. It is this shortcoming in the literature that we address in this article.

In more detail, a potential game with unknown noisy rewards is a game in which an agent's payoff for each outcome is drawn from a distribution with bounded variance whose mean is consistent with a potential function, in the sense of the standard definition of potential games [Monderer and Shapley, 1996b]. In this paper we derive new fictitious play and adaptive play processes for playing potential games with unknown noisy rewards, and provide conditions under which the agents' actions converge to Nash equilibria and the reward estimates converge to their true mean values. The adaptive processes we derive simultaneously perform recursive estimation of reward function means using Q–learning and adaptation to the strategies of others in the game. Our approach to these types of problems gives agents the ability to effectively learn their reward functions, while coordinating on a pure strategy Nash equilibrium. We choose to focus on fictitious play and adaptive play because their convergence is guaranteed using two different analytic techniques. Moreover, because these are the two main methods for proving convergence to Nash equilibrium in the learning in games literature, the versions of these two algorithms that we derive are exemplars for many other algorithms whose convergence to Nash equilibrium is proven using these methods (or very similar ones).

Furthermore, the typical application domains for these distributed optimisation methods are very large, and since the number of joint actions in a game is exponential in the number of agents, estimating a reward for each joint action quickly becomes an intractable problem (e.g. if a swarm of n autonomous vehicles each have four directions to move in, the resulting game has 4^n joint actions). In such settings, a typical technique for avoiding such computational difficulties is to find compact representations of the problem at hand. In this vein, we consider two common compact graphical representations of games known as *graphical normal form* and *hypergraphical normal form* [Kearns et al., 2001; Gottlob et al., 2005; Papadimitriou and Roughgarden, 2008]. These representations use a graph or hypergraph to summarise reward dependencies, and can be exponentially more compact than the standard normal form if the agents' interaction structure is sufficiently sparse.[1] We show how the structure that these representations encode can be exploited to derive efficient exploration policies for Q–learning, such that the learning problem facing the agents is significantly reduced.

Specifically, the main theoretic results of this paper are:

1. We derive multi–agent versions of Q–learning with ε–greedy exploration for which reward estimates converge to their true mean value, for games in standard normal form and two compact game representations (graphical and hypergraphical normal form). We use these as components when analysing the two families of learning algorithms below.
2. We prove that a novel variant of fictitious play using Q–learned estimates and employing the ε–greedy action selection rule converges to Nash equilibrium in repeated potential games, zero–sum games and several smaller classes of games with the fictitious play property.
3. We prove that three novel versions of adaptive play, in which agents evaluate their actions using Q–learned estimates of the reward and play ε–best responses to these estimates, converge to Nash equilibrium in repeated weakly acyclic games. These are: Standard adaptive play; *Payoff–based* adaptive play, a novel low–computation variant in which agents do not model their opponents directly, but rather evaluate their actions by their cumulative Q–learned estimates; and *Spatial* adaptive play, a low–memory variant in which agents ε–greedy respond to the last action profile played according to the Q–learned estimates, with the restriction that only one agent changes its strategy at a time.

As an example of a setting where potential games with unknown noisy rewards offer an effective framework for distributed optimisation, consider the ad hoc wireless sensor network management problem described in Farinelli et al. [2008], which is a version of a wide–

[1] Following Gottlob et al. [2005], we use the term "representation" to make clear that we are not considering a sub-class of games that do not fit the standard normal form for games. Rather, we use the distinction only to identify those games with useful (i.e. sparse) interaction structure.

area surveillance problem. The authors of this paper consider the problem of maximising the efficiency of a sensor network deployed for wide–area surveillance by coordinating the sense/sleep schedules of power constrained energy-harvesting sensor nodes. A sensor's daily available battery charge is constrained, so they can only be actively sensing for a limited time each day. For example, if this period is one third of the day, then the agent has to make a decision on which of the three thirds it chooses to actively sense, and which it should sleep for. In order to cover the entire field of observation, the sensors' observation ranges overlap, which means that the usefulness of each sensor's observations is coupled with that of its neighbours'. Hence, the first part of the problem is to coordinate sense/sleep cycles of the sensors so to maximise the expected number of events observed each day. However, these events occur at random, and, at the outset, the mean frequency of events is unknown to the sensors. This means that the sensors have to search the joint action space in order to learn the mean frequencies of events occurring, while also coordinating their sense/sleep cycles to reduce the likelihood to redundant event observations. Furthermore, the number of neighbours each node has is bounded because they each have limited observation ranges. This can be exploited by using one of the compact graphical representations introduced above to reduce the nodes' reward estimation task. We will come back to this example domain to demonstrate the efficacy of the learning algorithms we derive for playing potential games with unknown noisy rewards.

The paper progresses as follows: In the next section we review some related work in the area of algorithms for playing games with unknown and/or noisy rewards. We then introduce non–cooperative games, potential games, and the graphical and hypergraphical normal form representations, and define games with unknown and noisy rewards. In Section 4 we review Q–learning and the ε–greedy action selection rule, and derive multi–agent versions of Q–learning using ε–greedy action selection for which agents' estimates of their rewards for joint actions converge, for games in standard, graphical and hypergraphical normal form. We then prove the convergence of the two families of algorithms. Specifically, in Section 5, we show that if agents use fictitious play to adapt to their opponents' strategies, then play converges to a Nash equilibrium in potential games, and in Section 6 we show the same for different variants of adaptive play in weakly–acyclic games. Following these theoretical results, in Section 7 we compare the performance of the algorithms in two test domains. The first is a simple three–player game in which the performance of the algorithms can be clearly evaluated and compared to each other. The second is the ad hoc wireless sensor network for a wide–area surveillance problem, as described briefly above. This scenario also gives us the opportunity to demonstrate how an optimisation problem is transformed into a potential game using marginal contribution payoffs. Section 8 summarises the contributions of this paper, and discusses how our results may be extended to further algorithms.

2. Related work. Several authors have previously tackled the problem of learning Nash equilibria in games with unknown noisy rewards by applying Q–learning based approaches. Most closely related to our work is that of Claus and Boutilier [1998], who specify a *joint action learner* process, in which each agent keeps track of the frequency of other agent's actions (as in fictitious play), while at each time updating the reward for the joint action played. Although this does not substantially differ from the Q–learning fictitious play algorithm we derive here, the authors do not provide convergence conditions for their algorithm, and rely instead on experimental evidence of convergence. Specifically, they do not investigate the exploration rates required to ensure that the reward function estimates converge, nor do they make the link between the convergence of these estimates and convergence of the actions played to Nash equilibrium. Furthermore, their investigation is restricted to team games (games with a common payoff function), whereas we consider several further classes of games, and they do not consider the generalised and/or weakened versions of fictitious play or the adaptive play variants we consider in this paper. Additionally, several other authors consider *independent action learners*, in which agents use variants of the *Q*–learning procedure independent of each other, oblivious of the effects of changes in other agents' actions on their own payoffs. In particular, under the independent action learner algorithms of Claus and Boutilier [1998] and Cominetti et al. [2010], the agents update their estimate of the reward they receive for each of their actions, independent of the other agents, using *Q*–learning. These algorithms both use a Boltzmann distribution to guide action selection and sample the actions, but differ in the specific manner in which this is used, with Claus and Boutilier [1998] specifying an annealing schedule for the temperature coefficient and Cominetti et al. [2010] using a constant temperature. Neither authors prove convergence to Nash equilibrium; as a consequence, neither can make use of the price of anarchy bound on solution quality derived by Marden and Wierman [2008].

Single–agent learning in unknown noisy game environments has also been investigated in the context of zero–sum games. Baños [1968] considers two–player zero–sum games, in which one agent does not know the payoffs and receives only a noisy observation of the mean payoff for the action it plays each time a move is made. The author derives a class of strategies for this player that perform as well aysmtotically as if the player had known the mean payoffs of the games from the outset; that is, the player's average payoff converges to the maximin value of the game. Auer et al. [1995] consider an adversarial multi–armed bandit (MAB) problem, in which an adversary has control of the payoffs of each of the MAB's arms and aims to minimise the player's payoff (which contain the zero–sum games studied by Baños [1968] as a subclass). The authors provide an algorithm for general multi–player games that asymptotically guarantees a player its maximin value. Although for two–player zero–sum games this is the same guarantee as the strategy derived by Baños, the authors also show that their algorithm is more efficient than that of Baños. Both of these approaches, however, converge to a Nash equilibrium only in 2–player zero–sum games (where the Nash equilibrium, minimax, and maximin concepts give the same solution), so do not apply to our problem of computing Nash equilibria in potential and/or weakly–acyclic games.

Several other algorithms have been proposed for games where agents cannot observe their opponents' actions, so the payoffs that they receive may differ as the other player's actions change.[2] One such approach is that of Hart and Mas-Colell [2000], who introduce *regret matching*. This algorithm converges to the set of correlated equilibria in all finite games, and in particular, a variant of it converges even when the players do not know the game payoffs and cannot observe their opponents' actions, so they must be learned over the course of the game. However, although the set of correlated equilibria include all Nash equilibria, correlated equilibria are not (necessarily) optimal in potential games in the same way as Nash equilibria, in that they do not locally maximise the potential function of such games, so regret matching is not directly applicable to our setting. A second relevant approach to games with unknown rewards and unobserved opponent actions is given in Marden et al. [2009], who provide three payoff–based dynamics that converge to pure–strategy Nash equilibria in weakly acyclic games, one of which, *sample experimentation dynamics*, can admit perturbations in agents' rewards. This algorithm alternates between two phases — exploration and exploitation. However, its main drawback is that it requires several parameters to be

[2] Note that this is a different scenario to the situation we consider: here, agents' payoffs are corrupted by noise that is induced by their opponents' switching actions unobserved, whereas our work considers noise in rewards that is caused by some exogenous random perturbation under the assumption that opponents' actions can be observed. The first case can be thought of as model–free setting, while in the second (our setting), each agent has a model of their opponents.

set, which control the exploration phase length, exploration rates, and tolerances on payoff difference and switching rates for deciding when to change strategies. Now, because these parameters are integral to the algorithm's convergence guarantees, a user of these algorithms must have sufficient *a priori* knowledge of the problem at hand to set the parameters in a way that ensures that the algorithms do indeed converge. Moreover, the sample experimentation dynamics is designed for games where agents cannot observe their opponents' actions, whereas our work does not address scenarios with this restriction.

Finally, the only algorithms proven to converge, in some sense, to Nash equilibrium in all games are the *regret–testing* algorithms of Young and Foster [Foster and Young, 2006; Young, 2009]. These algorithms will stay near a Nash equilibrium for a long time once it has been reached, but essentially perform a random exhaustive search to find an equilibrium in the first place. We sacrifice this convergence in all games in order to improve the search mechanism in the games we are interested in (i.e. classes of games directly associated with distributed optimisation problems).

3. Background. In this section we review non–cooperative games, potential games, and graphical and hypergraphical representations of games, and define games with unknown noisy rewards.

3.1. Noncooperative games. A finite noncooperative game in standard normal form (SNF), $\Gamma = \langle N, \{A_i, r_i\}_{i\in N}\rangle$, consists of a finite set of agents $N = \{1,\dots,n\}$, and for each agent $i \in N$, a finite set of (pure) *actions* A_i, with joint action space $A = \times_{i=1}^{N} A_i$, and a *reward function* $r_i : A \to \mathbb{R}$. An agent's reward function specifies its ranking over all joint action profiles, $a \in A$, also called *outcomes* of the game. Agents can also choose to play an action according to a lottery π_i, known as a *mixed strategy*. This is a probability distribution over the pure action set A_i, so that $\pi_i \in \Delta_i$, the set of probability distributions over A_i. The reward functions of the mixed extension of the game are given by the expected value of r_i under all agents' joint independent lottery $\pi \in \times_{i\in N}\Delta_i$ over A:

$$r_i(\pi) = \sum_{a\in A}\left(\prod_{j\in N}\pi_j(a_j)\right) r_i(a).$$

We will use the notation $a = (a_i, a_{-i})$ where a_{-i} is the joint action chosen by all agents other than i, and $\pi = (\pi_i, \pi_{-i})$ where π_{-i} is the joint independent lottery of all agents other than i.

In this paper we assume, as is standard, that the r_i are bounded, and consequently there exists $\bar{r}$ such that $\max_{i\in N, a\in A}|r_i(a)| \leq \bar{r}$. An agent's goal is to maximise its reward, and its *best response*, $b_i(\pi_{-i})$, is the set of i's best strategies, given the strategies of the other agents:

$$b_i(\pi_{-i}) = \{\pi_i \in \Delta_i : r_i(\pi_i, \pi_{-i}) = \max_{\tilde{\pi}_i\in\Delta_i} r_i(\tilde{\pi}_i, \pi_{-i})\}$$

Stable points are characterised by the set of *Nash equilibria*, which are defined as those joint strategy profiles, π^*, in which no individual agent has an incentive to change its action:

$$r_i(\pi_i^*, \pi_{-i}^*) - r_i(\pi_i, \pi_{-i}^*) \geq 0 \quad \forall\, \pi_i,\ \forall\, i.$$

That is, in a Nash equilibrium, $\pi_i^* \in b_i(\pi_{-i}^*)$.

We can also define a δ–best response, and the associated δ–Nash equilibrium, which will be useful in the analysis of exploratory action selection in order to estimate action values. First, let the δ-*best response correspondence*, $b_i^{\delta}(\pi_{-i})$ be the set of strategies that come within δ of maximising an agent's reward, conditional on other agents' strategies:

$$b_i^{\delta}(\pi_{-i}) = \{\pi_i \in \Delta_i : r_i(\pi_i, \pi_{-i}) \geq \max_{\tilde{\pi}_i\in\Delta_i} r_i(\tilde{\pi}_i, \pi_{-i}) - \delta\}. \tag{3.1}$$

Then, a strategy profile π^* is an δ–Nash equilibrium if $\pi_i^* \in b_i^{\delta}(\pi_{-i}^*)$ for all $i \in N$.

3.2. Potential and weakly acyclic games. The exposition so far has considered general classes of games. Of particular interest to the control community are identical interest games, in which all individuals receive an identical reward so that

$$r^i(a) = r(a) \quad \forall i,$$

and their generalisation to potential games [Monderer and Shapley, 1996b]. Here the common reward function, or the potential function, represents a system reward to be optimised through action selection by independent agents.

The class of potential games is characterised as those games that admit a function specifying the participants' joint preference over outcomes [Monderer and Shapley, 1996b]. This function is known as a potential function and, generally, it is a real-valued function on the joint action space A such that the difference in the potential induced by a unilateral deviation of action equals the change in the deviator's reward.

DEFINITION 3.1 (Potential games). *A function $P : A \to \mathbb{R}$ is a* potential *for a game if:*

$$P(a_i, a_{-i}) - P(a_i', a_{-i}) = r_i(a_i, a_{-i}) - r_i(a_i', a_{-i}) \quad \forall\, a_i,\, a_i' \in A_i \quad \forall a_{-i} \in A_{-i} \quad \forall\, i \in N.$$

A game is called a potential game *if it admits a potential.*

A potential function has a natural interpretation as representing opportunities for improvement to an agent defecting from any given action profile. As the potential function incorporates the strategic possibilities of all agents simultaneously, the local optima of the potential function are Nash equilibria of the game; that is, the potential function is locally maximised by self-interested agents in a system.

A useful property of potential games is the fact that the existence of a potential function for a game implies a strict joint preference ordering over game outcomes. This, in turn, ensures that the game possesses the *finite improvement property*, or FIP. A *step* in a game Γ is a change in one agent's strategy. An *improvement step* in Γ is a change in one agent's strategy such that its reward is improved. A *path* in Γ is a sequence of steps, $\phi = (a^0, a^1, \ldots, a^t \ldots)$, in which exactly one agent changes its strategy at each step t. A path has an *initial point*, a^0, and if it is of finite length T, a *terminal point a^T*. A path ϕ is an *improvement path* in Γ if for all t, $r_i(a^{t-1}) < r_i(a^t)$ for the deviating agent i at step t. A game Γ is said to have the *finite improvement property* if every improvement path is finite, and Monderer and Shapley [1996b] prove that this is the case for every potential game.

Related to this is the concept of a *weakly acyclic* game, which is needed to discuss the convergence of the adaptive play processes in Section 6. A game is *acyclic* if there is no improvement path with $a^0 = a^T$ for $T > 0$. A game is *weakly acyclic* if, from any joint strategy, there is an improvement path that reaches a pure strategy Nash equilibrium. Note that an acyclic game is weakly acyclic, and any potential game in which no agent is indifferent between distinct strategies is acyclic [Young, 1998]. In a weakly acyclic game, for each $a \in A$, let L_a be the length of the shortest improvement path from a to a pure strategy Nash equilibrium, and let $L_\Gamma = \max_{a \in A} L_a$; we will need this constant in Section 6.

3.3. Compact graphical representations of games. In this paper we investigate a scenario where the individuals attempt to estimate their expected reward $r^i(a)$ for each joint action $a \in A$. In general games, however, the joint action space A grows exponentially with the number of agents, so this estimation problem (as well as standard adaptive processes such as fictitious play) becomes impractical, because the number of joint actions to sample is so large. However in systems with an inherent structure, such as those with a natural spatial structure in which interaction only directly occurs between geographically close individuals, agents should only need to consider the actions of their neighbours. If a game admits a compact representation, then this form of the game can be exploited to improve the agents' learning rates, and in this paper we will show how two representations of sparse interaction in games can be used in this way.

The first is *graphical normal form* (GNF), a representation that can represent noncooperative games in which some agents' rewards are independent of others' strategies [Kearns et al., 2001]. In this form, the nodes of a graph correspond to the set of agents, while edges connect an agent to the others with which it shares a reward dependency, called its neighbours. The neighbourhood of i is the smallest set ν_i of players such that agent i's reward is entirely determined by a_i and $\{a_j : j \in \nu_i\}$. We say an undirected reward dependency exists between i and $j(\neq i)$ if either $j \in \nu_i$ or $i \in \nu_j$.

DEFINITION 3.2. *A game in **GNF** comprises a set of agents located on the nodes of a graph. An agent is connected to those with which it shares an undirected reward dependency, which make up its set of neighbours $\nu_i \subseteq N$. Its reward function, $r_i(a_{i,\nu_i})$, is then given by an array indexed by tuples from the set $\times_{j\in\{i,\nu_i\}}|A_j|$.*

The second useful compact representation is *hypergraphical normal form* (HNF) [Gottlob et al., 2005; Papadimitriou and Roughgarden, 2008], which comprises hyperedges representing a set of local games that each contain several agents. An agent is typically involved in more than one local game, and its neighbours are those it is linked to via any local game.

DEFINITION 3.3. *A game in **HNF** comprises a set of agents located on the nodes of a hypergraph. Each hyperedge represents a local game: $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$, where $\gamma = \langle N_\gamma, \{A_i, r_{i,\gamma}\}_{i\in N_\gamma}\rangle$, defined as in SNF. Let $\Gamma_i = \{\gamma : i \in N_\gamma\}$ be the set of local games containing agent i. Player i's action set, A_i, is identical in all $\gamma \in \Gamma_i$, and it selects a single action $a_i \in A_i$ to play in all of its local games. Its neighbours in $\gamma \in \Gamma_i$ are $\nu_{i,\gamma} = N_\gamma \setminus i$, and its reward from γ, $r_{i,\gamma}(a_\gamma)$ is given by an array indexed by tuples from the set $\times_{j\in N_\gamma}|A_j|$. Its full set of neighbours is given by $\nu_i = \cup_{\gamma\in\Gamma_i} N_\gamma \setminus i$, and its reward is the sum of its rewards from $\gamma \in \Gamma_i$: $r_i(a_i, a_{i,\nu_i}) = \sum_{\gamma\in\Gamma_i} r_{i,\gamma}(a_i, a_{\nu_{i,\gamma}})$, where $a_{\nu_{i,\gamma}}$ is the joint action of i's neighbours in γ.*

Note that, in both compact representations, $r_i(a)$ now only depends on a_i and a_{ν_i}, where a_{ν_i} is the joint action of all the neighbours of i. Subsequently, we shall write r_i as a function of the joint actions of i and its neighbours, that is, $r_i(a_{i,\nu_i})$.

Finally, note that games in SNF can be represented in both GNF (with a complete graph) and HNF (with a single, global, local game γ). Hence for the rest of the paper we focus on the classes GNF and HNF, and all results will apply directly to games in SNF.

3.4. Games with unknown noisy rewards. We now introduce the model of rewards received in a repeated learning situation that will be studied in the rest of this article. Much work on learning in games either assumes that the reward functions r_i are known in advance [e.g. Hart and Mas-Colell, 2000], or that the observed rewards are deterministic functions of the joint action selected [e.g Rosenthal, 1973; Cominetti et al., 2010]. However, as argued in Section 1, a more realistic scenario is that the observed rewards are noisy, and comprise of an expected value equal to the unknown underlying reward function $r_i(a)$ and a zero–mean random perturbation. We call this scenario *unknown noisy rewards*. This situation therefore requires the individuals to estimate their underlying reward functions, while also adapting their strategies in response to the actions of other agents.

DEFINITION 3.4. *A* game with unknown noisy rewards *is a game in which, when the joint action $a \in A$ is played, agent i receives the reward*

$$R_i = r_i(a) + e_i \tag{3.2}$$

where $r_i(a)$ is the true expected reward to agent i from joint action $a \in A$, and e_i is a random variable with expected value 0 and bounded variance. Games in GNF with unknown noisy rewards *are defined similarly, with the difference being that when the joint action $a \in A$ is played, agent i receives the reward*

$$R_i = r_i(a_i, a_{\nu_i}) + e_i, \tag{3.3}$$

where $r_i(a_i, a_{\nu_i})$ is the true expected reward to agent i for the joint action (a_i, a_{ν_i}), and e_i is a random variable with zero mean and bounded variance. Finally, for games in HNF with unknown noisy rewards, *when the joint action $a \in A$ is played, agent i receives the (independently observable) rewards*

$$R_{i,\gamma} = r_{i,\gamma}(a_\gamma) + e_{i,\gamma} \quad \forall \gamma \in \Gamma_i, \tag{3.4}$$

where $r_{i,\gamma}(a_\gamma)$ is the true expected reward to agent i from local game γ for the joint action a_γ, and each $e_{i,\gamma}$ is a random variable with zero mean and bounded variance.

To avoid unnecessary over–complication in this article, we assume that each realisation of each e_i is independent of all other random variables.[3] Note that a game with unknown noisy rewards is a generalisation of the bandit problem discussed by Sutton and Barto [1998], and we shall use similar reinforcement learning strategies to estimate the values of $r_i(a_{i,\nu_i})$.

3.5. Problem definition. We are now in a position to precisely describe the problem which we address. We imagine a game with unknown noisy rewards which is repeated over time. On each play of the game, the individuals select an action, and receive rewards as in (3.3) or (3.4) for games in GNF and HNF, respectively (recalling that a game in SNF can be captured by either representation). The individuals also observe the actions selected by their neighbours (as defined for each representation). Based on this information, the individuals update their estimates of the reward functions and adapt their strategies.

We are interested in the evolution of strategies under this scenario, and in particular whether strategies converge to a Nash equilibrium. If the underlying game is a potential game corresponding to a distributed optimisation problem, convergence to Nash equilibrium gives us distributed convergence to a (locally) optimal joint strategy with only noisy evaluations of the target function.

4. Convergence of reward function estimates using *Q*–learning. In this section we show that, in a game with unknown noisy rewards, agents can form estimates of the true reward functions which are asymptotically accurate, provided that all joint actions are played infinitely often. We also show how this condition can be guaranteed for games in GNF and HNF. In particular, for each representation, we show that if the agents update their estimates of the expected rewards for joint actions using *Q*–learning, and select actions using an appropriate ε–greedy action selection policy, then with probability 1 the reward function estimates will converge to their true mean values.

4.1. Review of *Q*–learning. In noisy environments, reinforcement learning is often used to estimate the mean value of a perturbed reward function [Sutton and Barto, 1998]. In particular, we consider *Q*–learning for single–agent multi–armed bandit problems, in which one learner selects actions a and receives rewards R. This algorithm operates by recursively updating an estimate of the value of the action taken at time t, and in single state problems analogous to repeated games takes the form:

$$Q^{t+1}(a) = Q^t(a) + \lambda^t \, I\{a^t = a\} \left(R^t - Q^t(a)\right), \tag{4.1}$$

[3] This assumption can be significantly relaxed without comprising our results, but requires significant effort to explain how estimation is adapted to handle correlated errors, which is beyond the scope of this paper.

where $I\{a^t = a\}$ is an indicator function taking value 1 if $a^t = a$ and 0 otherwise and $\lambda^t \in (0,1)$ is a learning parameter.

In general, $Q_t(a) \to \mathbb{E}[R^t \,|\, a^t = a]$ with probability 1 if the conditions

$$\sum_{t=1}^{\infty} \lambda^t I\{a^t = a\} = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} (\lambda^t)^2 < \infty$$

hold for each a [Singh et al., 2000]. This can be achieved, under the condition that all $Q_i(a)$ are updated infinitely often, if

$$\lambda^t = \left(C_\lambda + \#^t(a^t)\right)^{-\rho_\lambda}$$

where $C_\lambda > 0$ is an arbitrary constant, $\rho_\lambda \in (1/2, 1]$ is a learning rate parameter, and $\#^t(a)$ is the number of times the action a has been selected up to time t.

The condition that all actions a are played infinitely often can be met with probability 1 by using a randomised *action decision rule* (or learning policy, in the terminology of Singh et al. [2000]) in which the probability of playing each joint action is bounded below by a sequence that tends to zero sufficiently slowly as t becomes large. Furthermore, this action decision rule can be chosen so that it is greedy in the limit, in that the probability with which it selects maximal reward actions tends to 1 as $t \to \infty$. Such policies are called *greedy in the limit with infinite exploration (GLIE)* [Singh et al., 2000].

One common GLIE decision rule is known as ε–greedy, and the results derived in this paper depend on the use of this particular rule. Under this rule, an agent selects an action with maximal expected reward at time t with probability $(1-\varepsilon^t)$, and a random other action with probability ε^t. In the single agent case, if $\varepsilon^t = c/t$ with $0 < c < 1$, then for any a,

$$\sum_{t=1}^{\infty} Pr(a^t = a) \geq \sum_{t=1}^{\infty} \frac{\varepsilon^t}{|A|} = c \times \sum_{t=1}^{\infty} 1/t = \infty,$$

and so (by a generalised Borel–Cantelli lemma [Singh et al., 2000]) with probability 1 each action is selected infinitely often.

We now state a lemma giving conditions for convergence of Q-learning for general action spaces. The proof of the lemma is a simple application of stochastic approximation theory, as in Singh et al. [2000], and is not given here.

LEMMA 4.1. *Let A be any action space, and let $Q^t(a)$ follow the recursion in (4.1) for each $a \in A$. Suppose, for each $a \in A$, and for all t,*

$$Pr(a^t = a) \geq C\varepsilon^t \quad \textit{with} \quad \sum_{t=1}^{\infty} \varepsilon^t = \infty, \tag{4.2}$$

where $C > 0$ is a constant. Then

$$\lim_{t\to\infty} Q_t(a) = r(a) \quad \textit{with probability 1.}$$

In the next section we derive a new version of Q–learning specifically for estimating reward functions in multi–agent settings, which is the first of the contributions listed in Section 1, and is used in the subsequent derivations of convergent fictitious play and adaptive play algorithms.

4.2. Estimating rewards in games in standard normal form (SNF). The Q-learning scheme above can be applied independently by each player of a game, who learns the expected reward for each action $a_i \in A_i$, ignoring the actions selected by the other agents [Claus and Boutilier, 1998; Leslie and Collins, 2005; Cominetti et al., 2010]. However this can result in very slow adaptation of strategies towards Nash equilibrium. Instead, in this paper, we allow the learning of reward functions of joint actions, and simultaneous explicit reasoning about the action selection of the other agents. This is the joint action learning approach suggested (without analysis) in the context of fictitious play by Claus and Boutilier [1998], and furthermore we argue that the applicability of the technique relies on the compact representations introduced in Section 3.3.

To begin, we consider the convergence of this Q–learning scheme in games in SNF with unknown noisy rewards using an ε–greedy decision rule (although note that we have not yet defined what a greedy action should be in this context, since an optimal action will depend on the assumed strategy of the other agents). After playing action a_i^t, observing actions a^t, and receiving reward R_i^t, each individual i updates estimates Q_i^t using the equation

$$Q_i^{t+1}(a) = Q_i^t(a) + \lambda^t I\{a^t = a\}\left(R_i^t - Q_i^t(a)\right) \quad \forall a \in A. \tag{4.3}$$

In contrast to single agent settings, in multi–player games, the choice of joint action is made by the independent choices of more than one agent. As such, for each Q value to be updated infinitely often, the schedule that the exploration sequence $\{\varepsilon^t\}_{t\to\infty}$ follows must reflect the fact that the agents cannot explicitly coordinate to sample specific joint action profiles.

LEMMA 4.2. *In a game with unknown noisy rewards, if agents select their actions using a policy in which, for all $i \in N$, $a_i \in A_i$ and $t \geq 1$,*

$$Pr(a_i^t = a_i) \geq \varepsilon_i^t, \quad \textit{with} \quad \varepsilon_i^t = c_\varepsilon t^{-1/|N|},$$

where $c_\varepsilon > 0$ is a positive constant, then

$$\lim_{t\to\infty} |Q_i^t(a) - r_i(a)| = 0 \quad \forall i \in N, \quad \forall a \in A. \tag{4.4}$$

Proof. If the probability that agent i selects an action is bounded below by $\varepsilon_i^t = c_\varepsilon t^{-1/|N|}$, then the probability that any joint action a is played is bounded below by

$$\left(t^{-1/|N|}\right)^{|N|} = (c_\varepsilon)^{|N|} t^{-1}.$$

Hence we apply Lemma 4.1 to Q_i with action space A, and the result follows. □

This may result in a practical learning procedure if $|N|$ is sufficiently small. However, in large games, visiting each joint action infinitely often is an impractical constraint — to achieve sufficiently high experimentation rates through independent random sampling, as in the ε–greedy approach, would require the agents' independent ε sequences to decrease so slowly that in any practical sense the agents will never move into an exploitation phase. On the other hand, joint exploration requires a large degree of cooperation between the agents in order to select and sample specific joint actions, to the point that it ceases to be a truly decentralised system.

However, if each agent interacts with only a few other agents, as is the case in a game that can be succinctly represented in GNF or HNF, then the joint action space to be explored by each agent, and the number of reward values each individual estimates, can be significantly reduced. This allows the agents to use independent ε–greedy strategies that succeed in sampling all the joint actions *within each neighbourhood* while still becoming greedy over a useful time frame. Indeed the GNF and HNF representations allow agent i to learn the reward functions of the reduced joint action space given by the Cartesian product of it and its neighbours' action spaces only, which for large games is a much more feasible task than estimating the full reward function on A. In the following two sections, we formalise sufficient conditions on the ε^t schedule for games that may be succinctly represented in GNF or HNF that ensure Q–learning converges.

4.3. Estimating rewards in games in graphical normal form (GNF). For games in GNF, each agent needs to learn only the reduced space of joint actions given by the Cartesian product of it and its neighbours' action spaces. As such, each individual i now updates its estimates Q_i^t using the equation

$$Q_i^{t+1}(a_{i,\nu_i}) = Q_i^t(a_{i,\nu_i}) + \lambda^t I\{a^t_{i,\nu_i} = a_{i,\nu_i}\}\left(R_i^t - Q_i^t(a_{i,\nu_i})\right) \quad \forall a_{i,\nu_i} \in A_{i,\nu_i}. \tag{4.5}$$

In this case, the schedule that the sequence $\{\varepsilon^t\}_{t\to\infty}$ follows in order to guarantee that each Q value is updated infinitely often can be altered to take advantage of the reduced size of each agent's joint action space.

LEMMA 4.3. *In a game in GNF, let i's* neighbourhood size *be the number of neighbours of i plus 1 for i itself. Given this, let J_i be the size of the largest of the neighbourhoods of i or any j in ν_i. In a game with unknown noisy rewards, if agents select their actions using a policy in which, for all $i \in N$, $a_i \in A_i$ and $t \geq 1$,*

$$Pr(a_i^t = a_i) \geq \varepsilon_i^t, \quad \textit{with} \quad \varepsilon_i^t = c_\varepsilon t^{-1/J_i},$$

where $c_\varepsilon > 0$ is a positive constant, then

$$\lim_{t\to\infty} |Q_i^t(a_{i,\nu_i}) - r_i(a_{i,\nu_i})| = 0 \quad \forall i \in N, \quad \forall a_{i,\nu_i} \in A_{i,\nu_i}. \tag{4.6}$$

Proof. If the probability that agent i selects an action is bounded below by $\varepsilon_i^t = c_\varepsilon t^{-1/J_i}$, then the probability that any joint action a_{i,ν_i} is played is bounded below by

$$\prod_{j\in\{i\}\cup\nu_i} c_\varepsilon t^{-1/J_j} \geq \left(c_\varepsilon t^{-1/(|\nu_i|+1)}\right)^{|\nu_i|+1} = (c_\varepsilon)^{|\nu_i|+1} t^{-1},$$

because $J_j \geq |\nu_i| + 1$. Hence we apply Lemma 4.1 to Q_i with action space A_{i,ν_i}, and the result follows. □

4.4. Estimating rewards in games in hypergraphical normal form (HNF). In the setting of a game in HNF, each agent can learn the payoffs for joint actions in each of its local games independently. Hence, an individual i now updates its estimate $Q^t_{i,\gamma}$ of its reward function for each γ using the equation

$$Q^{t+1}_{i,\gamma}(a_\gamma) = Q^t_{i,\gamma}(a_\gamma) + \lambda^t I\{a^t_\gamma = a_\gamma\}\left(R^t_{i,\gamma} - Q^t_{i,\gamma}(a_\gamma)\right) \quad \forall a_\gamma \in A_\gamma. \tag{4.7}$$

For games in HNF, each joint action in each local game is guaranteed to be sampled infinitely often by following the $\{\varepsilon^t\}_{t\to\infty}$ schedule given in the following Lemma.

LEMMA 4.4. *In a game in HNF, let J_i be the maximum number of participants in any single local game in Γ_i (i.e. $J_i = \max_{\gamma\in\Gamma_i} |N^\gamma|$). In a game with unknown noisy rewards, if agents select their actions using a policy in which, for all $i \in N$, $a_i \in A_i$ and $t \geq 1$,*

$$Pr(a_i^t = a_i) \geq \varepsilon_i^t, \quad \textit{with} \quad \varepsilon_i^t = c_\varepsilon t^{-1/J_i},$$

where $c_\varepsilon > 0$ *is a positive constant, then*

$$\lim_{t\to\infty} |Q^t_{i,\gamma}(a_\gamma) - r_{i,\gamma}(a_\gamma)| = 0 \quad \forall i \in N, \quad \forall \gamma \in \Gamma_i, \quad \forall a_\gamma \in A_\gamma. \tag{4.8}$$

Proof. If the probability that agent i selects an action is bounded below by $\varepsilon^t_i = c_\varepsilon t^{-1/J_i}$, then the probability that any joint action a_{i,ν_i} is played is bounded below by

$$\prod_{j\in N_\gamma} c_\varepsilon t^{-1/J_j} \geq \left(c_\varepsilon t^{-1/|N_\gamma|}\right)^{|N_\gamma|} = (c_\varepsilon)^{|N_\gamma|} t^{-1},$$

because $J_j \geq |N_\gamma|$ Again, the result follows from applying Lemma 4.1 to $Q_{i,\gamma}$ with action space A_γ. □

We have now derived techniques for estimating an agent's reward functions that can overcome the computational problems associated with learning rewards in large games by exploiting structured interaction between the agents. When interleaved with a suitable strategy adaptation process, these will result in an algorithm that converges to a Nash equilibrium in potential games with unknown noisy rewards. The following two sections discuss two families of such strategy adaptation processes, fictitious play and adaptive play. Although the approaches to proving convergence of algorithms in these families differ, we can, nonetheless, use the Q–learning results just proven to derive convergence conditions for algorithms in both families.

5. Fictitious play with learned reward functions. In this section, we show that if the agents (i) update their estimates of the expected rewards for joint actions using the Q–learning approach outlined above, (ii) update their beliefs over their opponents' actions using a FP process, and (iii) select a new action using an appropriately defined ε–greedy action selection policy, then their actions converge to a Nash equilibrium (in expected rewards) in potential games with unknown noisy rewards.

5.1. Review of generalised weakened fictitious play. To begin, we describe the classical fictitious play (FP) process [Brown, 1951], and then consider the broader class of generalised weakened fictitious play processes [Leslie and Collins, 2006]. Let agent i's *historical frequency* of playing a_i, be defined as:

$$\sigma^t_{i,a_i} = \frac{1}{t}\sum_{\tau=0}^{t-1} I\{a^\tau_i = a_i\}. \tag{5.1}$$

We write $\sigma^t = \{\sigma^t_{i,a_i}\}_{i\in N, a^i \in A^i}$ for the vector of these beliefs, and σ^t_{-i} for the beliefs about all agents other than i. In classical FP, the chosen action is a best–response to the historical frequencies of all the other agents; $a^t_i \in b_i(\sigma^t_{-i})$. Writing $b(\sigma) = \times_{i\in N} b_i(\sigma_{-i})$ for the set of joint best responses, the FP recursion can be restated as the recursive inclusion:

$$\sigma^{t+1} \in \left(1 - \frac{1}{t+1}\right)\sigma^t + \frac{1}{t+1} b(\sigma^t).$$

Building on this, Leslie and Collins [2006] define the class of generalised weakened fictitious play (GWFP) processes. These are processes that admit a more general belief–updating process and allow δ–best responses to be played by the agents. We write $b^\delta(\sigma) = \times_{i\in N} b^\delta_i(\sigma_{-i})$ for the set of joint δ-best responses. In a GWFP process, beliefs follow the inclusion:

$$\sigma^{t+1} \in (1-\alpha^{t+1})\sigma^t + \alpha^{t+1}(b^{\delta^t}(\sigma^t) + M^{t+1}), \tag{5.2}$$

with $\alpha^t \to 0$ and $\delta^t \to 0$ as $t \to \infty$, $\sum_{t\geq 1} \alpha^t = \infty$, and $\{M^t\}_{t\geq\, 1}$ a sequence of perturbations satisfying conditions on tail behaviour. Throughout this section we assume that the M^t are martingale differences (with expected value 0 given the history up to time t and bounded variance), which ensures that the tail conditions hold if $\sum_{t=1}^{\infty}(\alpha^t)^2 < \infty$. Leslie and Collins [2006] show that trajectories of process given in (5.2) are stochastic approximations of the differential inclusion:

$$\frac{d}{dt}\sigma^t \in b(\sigma^t) - \sigma^t. \tag{5.3}$$

Hence the limit set of a GWFP process (5.2) is a connected internally chain-recurrent set of the differential inclusion (5.3), which in turn implies that the limit set of a GWFP process consists of a connected set of Nash equilibria in potential games, two–player zero–sum games, and generic $2 \times n$ games [Leslie and Collins, 2006]. We make use of this result in the next section.

5.2. ***Q*–learning fictitious play.** We now show that if agents adapt their strategies by playing ε–greedy responses to their opponents' historical frequencies of play, with these best responses calculated with respect to learned reward functions, then not only do the reward function estimates converge, but the agents' strategies also converge to a Nash equilibrium in the classes of games mentioned previously. In order to prove this, it suffices to show that Q–learning FP is a GWFP process; that is, an agent's ε–greedy action selection policy with respect to the estimated Q values corresponds to a δ^t–best response to its opponents' historical frequency of play, with $\delta^t \to 0$ as $t \to \infty$.

Our model is that the agents use the versions of Q–learning described in (4.5) or (4.7) for games in GNF or HNF, respectively, to estimate each $r_i(a_{i,\nu_i})$, and update their beliefs over opponents' actions σ_{ν_i} using the FP belief updating rule given in (5.1). For ease of exposition, from here on we will use the uniform notation i, ν_i to refer to those agents whose actions affect i's payoff, as is used for GNF, with the understanding that this is N in SNF and $\{i, \cup_{\gamma\in\Gamma_i}\nu_{i,\gamma}\}$ in HNF. For HNF we also use as shorthand Q_i^t to denote the set of independent estimates $Q_{i,\gamma}^t$.

Given this, in all representational forms, the agents' estimated expected reward for selecting action $a_i \in A_i$ is:

$$\hat{r}_i(a_i, \sigma_{\nu_i}^t, Q_i^t) = \sum_{a_{i,\nu_i}\in A_{i,\nu_i}} \left(\prod_{j\in\nu_i} \sigma_{j,a_j}^t\right) Q_i^t(a_{i,\nu_i}).$$

Note that agent i need only know the historical frequencies of agents in ν_i in order to calculate $\hat{r}_i$. The reward for a mixed strategy π_i is then a linear combination of probabilities and rewards:

$$\hat{r}_i(\pi_i, \sigma_{\nu_i}^t, Q_i^t) = \sum_{a_i\in A_i} \pi_i^t(a_i)\hat{r}_i(a_i, \sigma_{\nu_i}^t, Q_i^t).$$

Now consider the case in which agents employ an adaptation of the ε–greedy action selection policy to choose an action based on their expected rewards $\hat{r}_i$. Specifically, we write the best response set based on the estimates Q_i^t as

$$B_i(\sigma_{\nu_i}^t, Q_i^t) = \underset{a_i\in A_i}{\operatorname{argmax}} \left[\hat{r}_i(a_i, \sigma_{\nu_i}^t, Q_i^t)\right].$$

Note that $B_i(\sigma_{\nu_i}^t, Q_i^t)$ is a set of actions, whereas the other best–response correspondences in this section are all sets of mixed strategies. This allows us to define an ε-greedy rule which places the following probability of selection on each a_i:

$$\tilde{B}^{\varepsilon}_{i,a_i}(\sigma^t_{\nu_i}, Q^t_i) = \begin{cases} \frac{1-\varepsilon}{|B_i(\sigma^t_{\nu_i},Q^t_i)|} & \text{if } a_i \in B_i(\sigma^t_{\nu_i}, Q^t_i), \\ \frac{\varepsilon}{|A_i|-|B_i(\sigma^t_{\nu_i},Q^t_i)|} & \text{otherwise.} \end{cases} \tag{5.4}$$

Agents will select actions according to the mixed strategy $\tilde{B}^{\varepsilon^t}_i(\sigma^t, Q^t_i)$, with ε^t following a suitably decreasing schedule, to ensure that all Q values are updated infinitely often, but that as $t \to \infty$ the strategy of agent i is close to being a best response according to $\hat{r}_i(\cdot, \sigma^t_{\nu_i}, Q^t_i)$.

DEFINITION 5.1. *A Q–learning FP process is a process $\{\sigma^t, Q^t\}_{t\to\infty}$ such that*

$$\begin{aligned} a^t_i &\sim \tilde{B}^{\varepsilon^t_i}_i(\sigma^t_{\nu_i}, Q^t_i) \quad \forall i \in N, \\ \sigma^{t+1}_{i,a_i} &= (1-\alpha^{t+1})\sigma^t_{i,a_i} + \alpha^{t+1} I\{a^t_i = a_i\} \quad \forall i \in N, \quad \forall a_i \in A_i, \quad \textit{and} \\ Q^{t+1}_i(a_{i,\nu_i}) &= Q^t_i(a_{i,\nu_i}) + \lambda^t I\{a^t_{i,\nu_i} = a_{i,\nu_i}\}(R^t_i(a_{i,\nu_i}) - Q^t_i(a_{i,\nu_i})) \quad \forall i \in N, \quad \forall a_{i,\nu_i} \in A_{i,\nu_i}. \end{aligned}$$

THEOREM 5.2. *Suppose that the agents' beliefs and estimates follow a Q–learning FP process $\{\sigma, Q\}$ for which:*

- $\alpha^t = (c_\alpha + t)^{-\rho_\alpha}$*, where* $c_\alpha > 0$ *and* $\rho_\alpha \in (1/2, 1]$,
- $\lambda^t = (c_\lambda + \#^t(a_i, a_{\nu_i}))^{-\rho_\lambda}$*, where* $c_\lambda > 0$ *and* $\rho_\lambda \in (1/2, 1]$,
- $\varepsilon^t_i = c_\varepsilon t^{-1/J_i}$*, where* $c_\varepsilon > 0$ *and* J_i *is as defined in Lemmas 4.3 and 4.4, for the GNF and HNF representations, respectively.*

Then the σ^t *follow a GWFP process.*

Proof. We know by Lemmas 4.3 and 4.4 and the conditions on λ_t and ε_t, that with probability 1:

$$\lim_{t\to\infty} |Q^t_i(a_{i,\nu_i}) - r_i(a_{i,\nu_i})| \to 0,$$

and there exists a sequence $\eta^t \to 0$ such that

$$\max_{i\in N} \max_{a_{i,\nu_i}\in A_{i,\nu_i}} |Q^t_i(a_{i,\nu_i}) - r_i(a_{i,\nu_i})| < \eta^t,$$

so the same can be said for any mixed strategy; specifically,

$$\max_{i\in N} |\hat{r}^t_i(\pi_i, \sigma^t_{\nu_i}, Q^t_i) - r_i(\pi_i, \sigma^t_{-i})| < \eta^t$$

for any $\pi_i \in \Delta_i$.

Now, let $\tilde{B}^{\varepsilon^t}_i(\sigma^t, Q^t_i)$ be the mixed strategy played by i at the tth time step to select action a^t_i. Then (recalling that $\max_{i\in N, a\in A} |r_i(a)| = \overline{r} < \infty$), for every t and i:

$$\begin{aligned} \hat{r}_i(\tilde{B}^{\varepsilon^t}_i(\sigma^t_{\nu_i}, Q^t), \sigma^t_{\nu_i}, Q^t_i) &\geq (1-\varepsilon^t) \max_{a_i\in A_i} \hat{r}_i(a_i, \sigma^t_{\nu_i}, Q^t_i) + \varepsilon^t \min_{a_i\in A_i} \hat{r}_i(a_i, \sigma^t_{\nu_i}, Q^t_i) \\ &\geq (1-\varepsilon^t)(\max_{a_i\in A_i} r_i(a_i, \sigma^t_{-i}) - \eta^t) + \varepsilon^t(-\overline{r} - \eta^t) \\ &\geq \max_{a_i\in A_i} r_i(a_i, \sigma^t_{-i}) - [\eta^t + 2\varepsilon^t \overline{r}]. \end{aligned}$$

Hence $\tilde{B}^{\varepsilon^t}_i(\sigma^t_{\nu_i}, Q^t_i) \in B^{\delta^t}_i(\sigma^t_{-i})$ for $\delta^t = \eta^t + 2\varepsilon^t \overline{r}$ and $\delta^t \to 0$ as $t \to \infty$.

Now $(\alpha^{t+1})^{-1}\left[\sigma^{t+1}_i - (1-\alpha^{t+1})\sigma^t_i\right]$ is a unit vector with a 1 in the position corresponding to action a^t_i. Hence, conditional on the history up to t, the expected value of this unit vector is simply $\tilde{B}_i^{\varepsilon^t}(\sigma^t_{\nu_i}, Q_i^t)$, and the variance is bounded. Therefore, defining M_i^t to be the martingale difference between the realised and expected value of this unit vector, we have that

$$(\alpha^{t+1})^{-1}\left[\sigma_i^{t+1} - (1-\alpha^{t+1})\sigma_i^t\right] = \tilde{B}_i^{\varepsilon^t}(\sigma^t_{\nu_i}, Q_i^t) + M_i^t \in B_i^{\delta^t}(\sigma^t_{-i}) + M_i^t.$$

Noting also that $\sum_{t=1}^{\infty}(\alpha^t)^2 < \infty$, so that the tail conditions on M^t hold, the σ^t therefore follow a GWFP process. □

The above result implies the following:

COROLLARY 5.3. *The strategies in a Q–learning FP process, under the conditions on* α^t, λ^t *and* ε^t *specified in Theorem 5.2, converge to a connected subset of Nash equilibria in potential games, two–player zero–sum games, and generic* $2 \times n$ *games.*

From the perspective of distributed optimisation, the most important consequence of this result is that it shows that *Q*–learning FP can be used to compute pure strategy Nash equilibria in potential games in which the potential function corresponds to the global objective function.

6. Adaptive play with learned reward functions. The second family of algorithm we consider is adaptive play. This is a class of processes in which agents maintain a finite history over their opponents' actions, and construct an estimate of their mixed strategies by sampling from this history [Young, 1993]. In this section we address the convergence properties of *Q*–learning variants of Young's standard adaptive play, analogous to the *Q*–learning FP investigated in Section 5. Specifically, if agents (i) update their reward estimates using the *Q*–learning approach outlined in Section 4.2, (ii) update their beliefs over their opponents' actions using an appropriate adaptive play process, and (iii) select a new action using the ε–greedy decision rule, then their actions converge to a Nash equilibrium in potential games with unknown noisy rewards. In this section we first review standard adaptive play, then detail two important versions of adaptive play — payoff–based adaptive and spatial adaptive play — and finally characterise the conditions on the game and the agents' memory and sample sizes for which these and other variants of adaptive play converge.

6.1. Review of adaptive play. Adaptive play (AP) is a learning process in repeated normal form games. It is similar to FP, in that agents observe the actions of opponents and select best responses (or δ-best responses). It differs in that each individual only has a finite memory, of length m, and recalls the previous m actions taken by its opponents. On each play of the game, each individual takes a sample of size $k \le m$ from this memory, and plays either a best response to the actions in the sample (with probability $1-\varepsilon$) or otherwise selects a random action. If $\varepsilon > 0$ this results in an ergodic Markov chain on the state space $\mathcal{M}$ consisting of all possible joint memories, and therefore there is a unique stationary distribution $\mu(\varepsilon) = \{\mu_M(\varepsilon)\}_{M\in\mathcal{M}}$. Call a memory configuration $M \in \mathcal{M}$ a stochastically stable state if $\lim_{\varepsilon\to 0}\mu_M(\varepsilon) > 0$. For this setting, Young [1993] shows that in a weakly acyclic game Γ the stochastically stable states are homogeneous joint memories each consisting entirely of one pure strategy Nash equilibrium provided that $k \le {}^{m}/_{(L_\Gamma+2)}$, where L_Γ is the constant defined in Section 3.2.

However, Young is not entirely clear in which way the best response should be calculated, in that it is only stated that the next action is a best response to the sample. This could mean at least two things when there are more than two agents. Individuals i could, as in FP, estimate the individual mixed strategies of all opponents (essentially, calculate σ_j^t independently for each $j \neq i$) based on the finite sample (instead of the full history, as in FP). Alternatively, individual i could calculate a joint mixed strategy over the other agents, as in Marden et al. [2005], and play a best response to this joint mixed strategy. Young's proofs of convergence are valid in both cases, since they rely entirely on best responses to pure strategies, which are identical under both regimes.

6.2. Payoff–based adaptive play (PAP). Building on the last point of the previous section, note that agents using AP need never actually estimate mixed strategies if the joint strategy approach is to be used, as in Marden et al. [2005]. Indeed identical decisions will be made by considering cumulative reward against the sampled actions, while reducing the informational demands on the agents. This motivates the following definition:

DEFINITION 6.1. Payoff-based adaptive play (PAP) *with memory size $m > 0$, sample size $k \leq m$, and error rate $\varepsilon \in (0,1)$ is a process under which each individual i samples k of the previous m plays of the game, then calculates the cumulative reward that each $a_i \in A_i$ would have received against the joint actions selected by the other agents on those plays of the game. With probability $1-\varepsilon$ the action maximising that cumulative reward is selected. Otherwise a random action is selected.*

THEOREM 6.2. *Suppose Γ is weakly acyclic and $k \leq m/(L_\Gamma+2)$. The stochastically stable states of payoff-based adaptive play are homogeneous joint memories each consisting entirely of one pure strategy Nash equilibrium.*

Proof. Since the proof of Young [1993] relies only on best responses to pure strategies, which are the same for best response to both individual and joint mixed strategies, the proof holds for both cases. Action selection under the cumulative reward paradigm is the same as under the joint strategy paradigm [Marden et al., 2005] and hence the same result holds for PAP as for AP. □

6.3. Review of spatial adaptive play (SAP). The third variant of AP we consider is spatial adaptive play [Young, 1998], a variation of AP in which not all individuals update their strategy simultaneously. Now, if both the memory m and the sample size k are 1, and only one agent at a time updates their strategy, then the procedure reduces to log-linear learning [Blume, 1993]. The convergence of this scheme, and generalisations, has recently been thoroughly investigated by Marden and Shamma [2008], showing that as $\varepsilon \to 0$ in a potential game the stochastically stable states are maximisers of the potential function.

Furthermore Arslan et al. [2007] suggest that if $\varepsilon \to 0$ as play proceeds in log-linear learning then the played joint strategy will converge to a globally optimal element of the set of Nash equilibria. We here clarify the relationship with simulated annealing, and in particular we indicate why the convergence proof of Geman and Geman [1984] will hold for log-linear learning. In particular, individual agents select actions according to a distribution under which the probability of choosing any action is bounded below by ε_t. Hence Lemma 2 of Geman and Geman [1984] continues to hold if $\varepsilon_t \geq t^{-1/N}$. Furthermore, since $\varepsilon_t \to 0$, Lemma 3 also continues to hold, and therefore their Theorem B holds, showing that strategies converge to the global maximum of the potential function. Although the lower bound on ε_t looks less strict than the logarithmically decreasing temperature of standard simulated annealing, it has exactly the same effect on the sampling probabilities of actions, resulting in very slow convergence. Indeed this $t^{-1/N}$ is precisely the rate of exploration that we introduced the GNF and HNF game representations to avoid (see Section 4). Note that a similar phenomenon is observed in the parameters for the multinomial logit action selection and ε-greedy action selection in Singh et al. [2000] — in the multinomial logit decision rule the temperature decreases logarithmically, while for the ε-greedy decision rule ε decreases as t^{-1}, but the action selection probabilities in the case of two actions are identical.

6.4. *Q*–learning adaptive play variants. As with FP, the adaptive play processes discussed above — AP, PAP and SAP — rely on knowledge of the reward functions r_i, whereas we are interested in situations where these reward functions are not known and can only be observed subject to stochastic perturbations. The synchronous varieties of adaptive play (AP and PAP) both satisfy $Pr(a_i^t = a_i) \geq \varepsilon^{|N|}$, so Lemmas 4.3 and 4.4 apply for the GNF and HNF representations, respectively, and the Q-learning approach will be useful. With the asynchronous updates of SAP we need to be more careful, since not all agents' actions are updated simultaneously. With fixed ε we have an ergodic Markov chain on the joint action space, so all joint actions will be played infinitely often; with decreasing $\varepsilon^t \geq t^{-1/N}$, this schedule was specifically chosen so that all actions are visited infinitely often, as in Lemma 2 of Geman and Geman [1984]; again the Q–learning approach will be successful. Hence we can consider Q–learning variations of these adaptive play processes, in which the action selection procedure is exactly as in the original process, but uses estimated Q values instead of the true reward functions.

THEOREM 6.3. *Q–learning versions of AP, PAP and SAP have the same convergence properties as the algorithms that use the true reward function.*

Proof. Since the reward functions are bounded in absolute value, and the action spaces and memory are finite, there exists an $\eta > 0$ such that if for all $i \in N$, and for all $a_{i,\nu_i} \in A_{i,\nu_i}$,

$$|Q_i^t(a_{i,\nu_i}) - r_i^t(a_{i,\nu_i})| < \eta. \tag{6.1}$$

then the decisions made are the same whether the individuals use r_i or Q_i^t.

We know that (4.4) holds, so that, with probability 1, there exists a T such that for all $t \geq T$, (6.1) holds.

Since, after T, the strategies of agents evolve exactly as if they were following the standard AP type process using r_i instead of Q_i^t, the convergence properties are just the same. □

The above results imply the following:

COROLLARY 6.4. *For small but fixed $\varepsilon > 0$, in weakly acyclic games, the stochastically stable states of Q–learning adaptive play, Q–learning payoff-based adaptive play, and Q–learning spatial adaptive play (with $k = m = 1$) are homogeneous joint memories each consisting entirely of one pure strategy Nash equilibrium. Under Q-learning spatial adaptive play with $k = m = 1$ and $\varepsilon^t = t^{-1/N}$ in a potential game the joint action converges to the joint action that globally maximises the potential.*

Consequently, like Q–learning FP, this result shows that the Q–learning AP algorithms can be used to compute pure strategy Nash equilibria in potential games in which the potential function corresponds to the global objective function.

7. Experimental evaluation. In this section we illustrate the efficacy of the Q–learning FP, AP, PAP and SAP algorithms as distributed optimisation tools, using two sets of simulated problems (in what follows, we drop the common Q–learning prefix and refer to the algorithms by only their belief updating rule). This experimental evaluation is necessary because of the absence of analytic methods for rigorously comparing the learning performance of the algorithms. In Section 7.1 we compare the algorithms in a simple game, so that their differences can be clearly demonstrated. Specifically, the mean values of the game form a three–player potential game with two strict Nash equilibria, one with a higher social welfare (sum of utilities) than the other. Then, in Section 7.2, the second set of simulated problems we consider is a wireless sensor network coordination game, based on a real–world wide–area surveillance problem. This is a large–scale distributed optimisation problem, and it allows us to demonstrate the overall efficacy of the approaches we have derived.

The main metric of performance we consider in both sets of problems is the expected value of the solution found by an algorithm. Given that the algorithms are guaranteed to converge to a (locally optimal) Nash equilibrium, this measure accounts for the respective probabilities of converging to different Nash equilibria. We used the same learning parameters

		Matt			
		Colin		*Colin*	
		Left	Right	Left	Right
Rowena	Up	(5,5,5)	(0,1,0)	(0,0,1)	(1,0,0)
	Down	(1,0,0)	(0,0,1)	(0,1,0)	(2,2,2)

Fig. 1: Three–player potential game.

for the algorithms throughout all of the experiments. Specifically, the Q–learning parameters and ε–greedy parameter were constant across all four algorithms with $\rho_\lambda = 1$, $c_\lambda = 0$ and $c_\varepsilon = 1/10$. We used standard FP, with $\rho_\alpha = 1$, $c_\alpha = 0$, and AP and PAP with memory size $m = 15$ and sample size $k = 3$ (for SAP $m = k = 1$ by definition). As benchmarks, we use two independent Q–learning procedures. The first, CMS QL, is the algorithm by Cominetti et al. [2010], which learns the values of independent actions via Equation (4.1) and uses Boltzmann action selection:

$$Pr(a^t = a') = \frac{e^{Q(a')/\eta}}{\sum_{a \in A} e^{Q(a)/\eta}}$$

with the temperature parameter fixed at $\eta = 1/20$. The second, CB QL, is the independent action learner presented in Claus and Boutilier [1998], which also uses Boltzmann action selection, but with the temperature parameter following the schedule $\eta = 16(0.9^t)$ (as is used in the author's description of this algorithm).

7.1. A simple three–player game. In this section we compare the algorithms in a simple three–player two–action potential game, with mean rewards given in Figure 1, in which Rowena selects the row, Colin the column and Matt the matrix, respectively. The agents receive rewards equal to these values plus uniform noise $e \in [-\zeta, \zeta]$, as in Equation 3.2, where ζ itself is uniformly drawn from $[5, 10]$ at the beginning of each scenario. The game in mean rewards has two strict (pure) Nash equilibria and one mixed Nash equilibrium. The strict Nash equilibrium located at (U, L, L) is globally optimal, or social welfare maximising, while the other pure Nash equilibrium at (D, R, R) is sub–optimal. The strict Nash equilibria have equal–sized basins of attraction, in that the same number and length best response paths lead to each one.

We use this game to compare the algorithms' learning performances in a transparent setting. Furthermore, since the algorithms are only guaranteed to converge to one of the strict Nash equilibria, and not necessarily to the optimum, we use this game to investigate the quality of the solutions found by the algorithms, and to compare their behaviour to that of the benchmarks. The value of an action profile is measured by the sum of the actual rewards to the agents playing the game, i.e. $\sum_{i \in N} R_i(a^t)$, and the mean values earned by each of the algorithms were recorded for 50 repetitions of 50 scenarios generated randomly as described above. We consider a duration of 1000 time–steps, not because all algorithms converge in this time, but because most interesting behaviour occurs during this period and the clearest differentiations can be made.

At a high level, Figure 2 shows that AP and PAP are the best performing algorithms in this simple game. FP, SAP and CB QL tend to perform comparably by the end of the simulation, although the trajectory of their behaviour is quite different, while CMS QL is significantly outperformed by all algorithms. Note also that the two Q–learning algorithms' solution qualities increases very rapidly at the beginning of the games, but plateau quite early. In contrast, the game–theoretic learning algorithm's average solutions increase in quality through

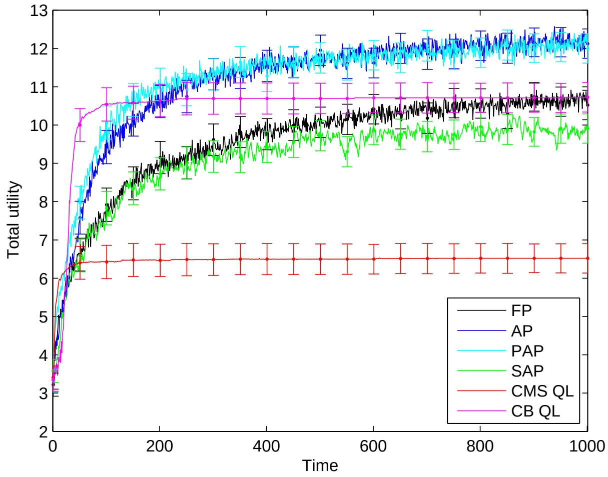

Fig. 2: Average total reward earned by the players in the simple three–player game.

the simulation. A further difference between the algorithms is that the *Q*–learning algorithms have much smoother mean trajectories than the game–theoretic algorithms.

To understand why this is the case, we look at the plots in Figure 3. The coloured areas of the plots in this figure illustrate the proportion of runs in which the optimal equilibrium (dark), sub-optimal equilibrium (light) or a non–equilibrium outcome (medium) is played. The bold dashed or dotted lines on the plots show the same proportions for the agents' intended play, that is, the actions they would have played if they played pure best–responses rather than sampling with probability ε. The distance between the actual and intended play of an equilibrium gives the proportion of non–equilibrium play that is due to the sampling induced by the ε–greedy rule.

The most noticeable feature of these plots is that, in a high proportion of simulations, FP, AP, PAP and SAP converge towards a Nash equilibrium, whereas in CMS QL and CB QL this is definitely not the case. Specifically, by the final time–step of the simulations, the proportion of runs in which the intended play is not a Nash equilibrium is less than 3% for AP and PAP, and approximately 8% for FP and 17% for SAP (this higher rate of non–convergence is to be expected with a learning algorithm that effectively replicates distributed simulated annealing). Furthermore, these proportions tend down over the duration of the game for all four algorithms. This indicates that, even when noise in early observations causes these algorithms to become temporarily stuck in low–payoff configurations, they continue to sample other actions at a rate that is high enough to learn to play better actions, leading them towards Nash equilibria. In contrast, at the termination of the simulations, CMS QL intend to play non–equilibrium profiles in approximately 20% of runs, and a huge 50% of CB QL runs, and these proportions are constant for a long period of play. That is, after 1000 time–steps, the best game–theoretic algorithms are almost 5 times less likely to have failed to converge to an equilibrium than the best naïve *Q*–learning approach. The explanation for this is that the *Q*–learning algorithms are becoming stuck in non–equilibrium, low–payoff action configurations. In particular, the annealing schedule of CB QL reduces the sampling rate more quickly than the conditions on the multi–agent GLIE policy derived in Section 4 permit. As such, it becomes mired with incorrect reward estimates in non–equilibrium outcomes,

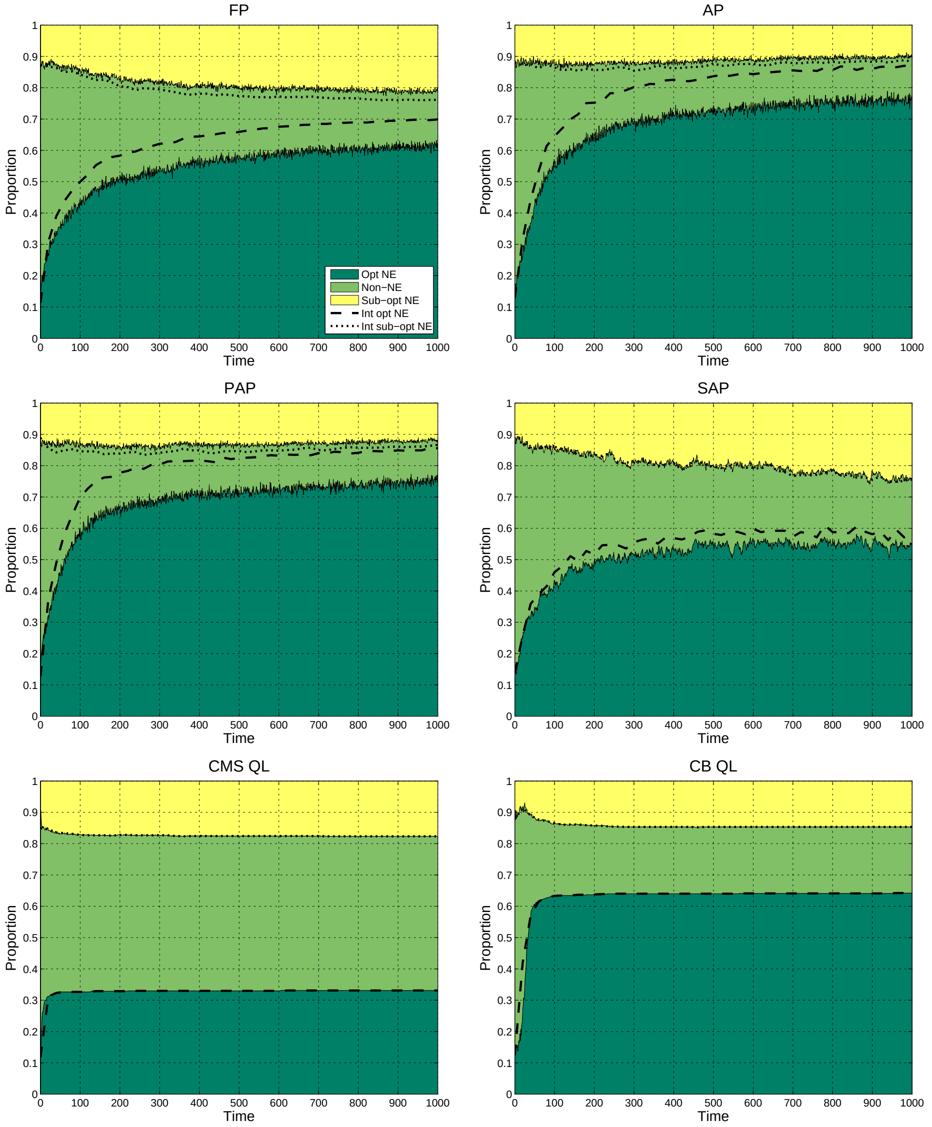

Fig. 3: Action profile time–series for the simple three–player game. The plots show proportions of actions played corresponding to the optimal Nash equilibrium (dark), non–Nash equilibrium (medium) and the sub–optimal Nash equilibrium (light), with intended play (i.e. without ε–greedy exploration) superimposed in bold dash or dotted lines.

and does not sample new actions frequently enough to learn that it is not playing a best response. This effect is also reflected in the *Q*–learning algorithms' relatively smooth global utility curves plotted in Figure 2. Moreover, besides the obvious effects on the global reward, because CMS QL and CB QL fail to converge a much greater proportion of the time, the price of anarchy bounds put on marginal contribution games constructed from optimisation problems cannot be validly applied to systems where the algorithms used do not have Nash equilibrium convergence guarantees.

The results in this section have illustrated the differences in the algorithms' behaviour and subsequent performance. Building on this, we now go on to demonstrate the usefulness of these algorithms in a large–scale optimisation problem.

7.2. An ad hoc wireless sensor network management problem. In this section we aim to demonstrate the usefulness of the learning algorithms derived in Sections 5 and 6 in large–scale optimisation and control, and to also give an example of how an optimisation problem can be transformed into a potential game using marginal contribution payoffs. The problem we consider is that of maximising the efficiency of a sensor network deployed for wide–area surveillance by coordinating the sense/sleep schedules of power constrained energy-harvesting sensor nodes. Specifically, the domain is a deployment of sensors distributed in an urban setting that can sense nearby traffic — these are acoustic and vibration sensors that can be used to detect foot or vehicle traffic. The sensors run on energy harvested from the environment, so are limited by their generation and storage capacities. That is, they operate in an "energy–neutral" mode, such that the energy that they expend is equal to that which they can generate [Farinelli et al., 2008; Kansal et al., 2007]. Since the activation of the sensor and the necessary signal–processing required to detect events is typically the most energy intensive activity, the sensors cannot be permanently powered. Rather, they must adopt a duty cycle and sensing schedule that maintains energy neutral operation. For example, if the length of time that it can sense for is one third of the day, then the agent has to decide on which third of the day it senses, and in which periods it sleeps. The sensors are assumed to be placed randomly, so in order to cover the entire field of observation, they are dispersed densely enough to ensure that nearby sensors' observation ranges overlap. As such, the usefulness of each sensor's observations is coupled with that of its neighbours', which are those sensors that cover a common section of road under surveillance. This spatial structure allows us to represent the problem as a game in GNF. An example of the simulation domain is given in Figure 4, which shows the sensors' locations and ranges and the underlying road network on which traffic flows.

The problem of optimising the coverage of the sensor network (i.e. the number of events observed) is divided into two parts. The first is to coordinate sense/sleep cycles of the sensors so as to maximise the expected number of events observed each day. However, these events occur at random, and, at the outset, the mean frequency of events is unknown to the sensors — below, we show that this makes the sensors' rewards unknown and noisy. The second part of the problem, then, is to learn the payoffs for different configurations of sensor cycles (which are a function of the unknown mean frequencies of events in the different regions under surveillance and the sleep/sense cycles of the sensors). To do this, the sensors have to learn their payoffs while also coordinating their sense/sleep cycles to maximise the number of event observed. The large number of sensor nodes (there may be hundreds in the system) and the constraints on their computation and communication rules out a centralised optimisation method, so a distributed method must be used. In particular, we use this problem domain to demonstrate the efficacy of the learning algorithms we derive for playing potential games with unknown noisy rewards.

In more detail, at the system–wide level, during any particular day a set X of traffic events occurs. The simulator generates several hundred potential event locations, and the probability of an event occurring at a particular location in a particular period of the day is given by a fixed probability (i.e. the probability of an event occurring varies across the periods of the day, but is fixed from day to day). Furthermore, the probabilities of events at different locations occurring at particular times are correlated, such that if the probability that x_1 occurs at t is high, then the likelihood of an event occurring at x_2 at the same time is higher than it would usually be. These correlations have their origins in the flow of traffic through the underlying

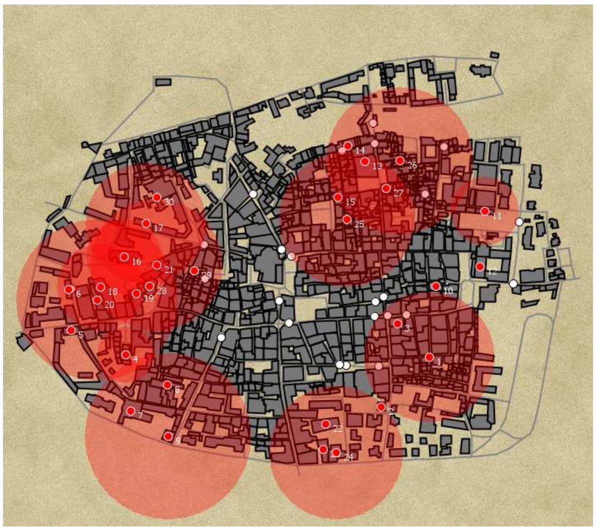

Fig. 4: The ad hoc wireless sensor network with overlapping sensor regions and the underlying road network. Solid red numbered dots are sensors, opaque red circles indicate the observation regions of active sensors, and white dots represent the vehicles causing events.

network. We define the value of a sensor observing an event, $x \in X$ as:

$$V^x(a) = 1 - \theta^{\#^x(a)}$$

where $\#^x(a)$ is the number of sensors that observe x (it is observed if it occurs within the sensing radius of a sensor at a time when the sensor is on), $0 < \theta < 1$ is a parameter that is used to differentiate between sensing cycle configurations that result in many redundant observations of the same event. It does this by imposing diminishing contributions to the global reward for each additional observation made of any single observation. Note that V^x is 0 if x goes unobserved. The agents time–stamp the events that they sense, and at the end of each day, they compare the lists of time–stamped events to evaluate their action (their choice of sensing period) for that day. An agent's reward for observing an event is the difference in reward it earns for the system for observing or not observing the event; that is, its marginal contribution to the system's performance:

$$R_i^x(a_i, a_{\nu_i}) = \theta^{-\#_x(a_i, a_{\nu_i})-1} - \theta^{-\#_x(a_i, a_{\nu_i})}$$

Then, each day, its total reward from a sensing cycle is the sum of rewards for all events it observes, $x \in X_i$:

$$R_i(a_i, a_{\nu_i}) = \sum_{x \in X_i} R_i^x(a_i, a_{\nu_i})$$

Note that i's reward depends on the actions of only those agents whose sensing ranges overlap with its own. In this way, neighbouring agents' payoffs are coupled, and the optimisation problem can be viewed as a game in GNF. This utility derivation results in a marginal–contribution potential game, with a potential given by the total system value for all events:

$$V(a) = \sum_{x \in X} 1 - \theta^{\#^x(a)},$$

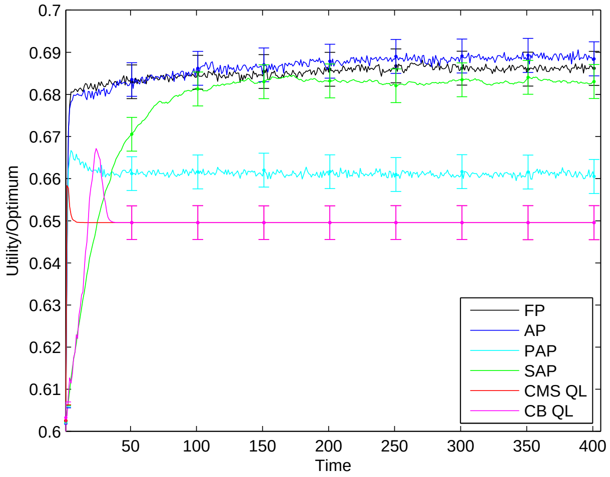

Fig. 5: Results of the sensor network simulations, plotting the average ratio of the reward earned to the global optimum for each scenario.

whose maxima correspond to the Nash equilibria of the associated game. By focusing on high–reward event observations, which are those that are observed by fewer sensors, an agent moves the system towards observing more events in total.

Given the daily rewards above, an agent uses Q-learning to estimate the reward it receives in a given time period (e.g. third of the day) and given the actions of its neighbours. Importantly, because an agent does not know what portion of its sensing areas overlap with its neighbours, it cannot use any observations they have made in the periods it was asleep to update the Q-values of joint actions other than the one it made, because it does not know if it would also have seen the events. The agent then uses FP or an AP variant to predict the strategies of its neighbours. It combines these two values to compute its expected reward, in terms of the expected number of unique event observations, it makes during each of the time periods in the next day, and then chooses a time period using the ε–greedy action selection rule. We recorded the ratio of the value of the solution found by an algorithm at each time step to the scenario optimum — that is, the proportion of the optimum, $V(a^t)/V(a^*)$, where a^* is optimal joint action for that scenario — so that we can aggregate our simulation results across scenarios with different payoff levels. We averages this measure over 30 runs each of 50 different scenarios.

The results of our simulations are given in Figure 5, which shows an overall good performance by all of the algorithms: apart from PAP, the game theoretic algorithms outperform the naïve Q–learning algorithms at a statistically significant level (standard error bars do not overlap). Furthermore, note the flattening–out of the Q–learning algorithms from an early point in time. The same regularity in the global reward earned by the Q–learning algorithms was seen in the previous section in the simple game scenario. In that setting, the relatively low payoffs to CMS QL and CB QL were due to the fact that they do not necessarily converge to a Nash equilibrium (local optimum). We conclude that the low–payoffs to these algorithms here is caused by the same effect. That is, these algorithms become stuck in low–value configurations because they do not sample new actions with a sufficient frequency to learn that this is the case. In contrast, the plots show that FP, AP and SAP, and to a lesser extent PAP, continue to improve their average performance over time, which matches with the guaranteed convergence to Nash equilibrium we derived in Sections 5 and 6.

We have now demonstrated that the FP, AP, PAP and SAP algorithms we derive in this article provide effective methods for controlling large distributed systems, in which agents initially have no knowledge of the value of their actions, so must learn their rewards online. These algorithms, which we have shown to out–perform their Q–learning alternatives, have the additional advantage over their alternatives of having bounds on their worst–case convergence points, if the optimisation problem is transformed into a game using marginal contribution payoffs.

8. Conclusions. In this article, we proved the convergence to Nash equilibria of variants of fictitious play and adaptive play in potential games and weakly acyclic games, respectively, with rewards that are initially unknown and which must be estimated over time from noisy observations. Potential games capture many important cooperative control problems in multi–agent systems, including the management of congested networks and task allocation and scheduling problems, and the results contained in this paper are directly applicable to such models with initially unknown reward functions, as we demonstrated via their instantiation in a wireless sensor network domain exemplar.

There are a number of ways in which this work may be taken forward. First, it may be possible to develop similar convergence proofs to cover other families of algorithms, such as joint–strategy fictitious play or regret matching. Second, different frameworks for online learning of noisy rewards may be employed to speed up estimating a game's payoffs, and consequently an algorithm's convergence to Nash equilibrium, such as PAC learning or by accurately learning only a best response path, rather than all of an agent's payoffs. Third, it may be possible to derive efficient sampling rate annealing schedules for other compact game representations, such as action–graph games. Fourth, there is an opportunity to extend the convergence of fictitious play and adaptive play variants in even more complicated settings, such as those where action observations are also perturbed, or where payoffs in the game vary according to some (possibly partially observable) state variable, such as is addressed for individual agents in the growing literature on contextual multi–armed bandits and multi–armed bandits with covariates.

References.

Arslan, G., Marden, J. R., and Shamma, J. S. (2007). Autonomous vehicle-target assignment: A game theoretical formulation. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):584–596.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pages 322–331, Washington, DC, USA. IEEE Computer Society.

Baños, A. (1968). On pseudo–games. *The Annals of Mathematical Statistics*, 39:1932–1945.

Blume, L. (1993). The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5:387–424.

Brown, G. W. (1951). Iterative solution of games by fictitious play. In Koopmans, T. C., editor, *Activity Analysis of Production and Allocation*, pages 374–376. John Wiley & Sons, Inc., New York.

Chapman, A. C., Micillo, R. A., Kota, R., and Jennings, N. R. (2010). Decentralised dynamic task allocation using overlapping potential games. *The Computer Journal*, 53(9):1462–1477.

Chapman, A. C., Rogers, A., Jennings, N. R., and Leslie, D. S. (in press). A unifying framework for iterative approximate best response algorithms for distributed constraint optimisation problems. *The Knowledge Engineering Review*.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *In Proceedings of the 15th AAAI National Conference on Artificial Intelligence*, pages 746–752. AAAI Press.

Cominetti, R., Melo, E., and Sorin, S. (2010). A payoff-based learning procedure and its application to traffic games. *Games and Economic Behavior*, 70(1):71–83. Special Issue In Honor of Ehud Kalai.

Farinelli, A., Rogers, A., and Jennings, N. R. (2008). Maximising sensor network efficiency through agent–based coordination of sense/sleep schedules. In *Workshop on Energy in Wireless Sensor Networks in conjunction with DCOSS 2008*, pages IV–43–IV–56, Marina Del Rey, CA, USA.

Foster, D. P. and Young, H. P. (2006). Regret testing: learning to play Nash equilibrium without knowing you have an opponent. *Theoretical Economics*, 1:341–367.

Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games*. MIT Press, Cambridge, MA.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Gottlob, G., Greco, G., and Scarcello, F. (2005). Pure Nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research*, 24:357–406.

Hart, S. and Mas-Colell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150.

Kansal, A., Hsu, J., Zahedi, S., and Srivastava, M. B. (2007). Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*, 6(4):32:1–32:38.

Kearns, M., Littman, M., and Singh, S. (2001). Graphical models for game theory. In *Proceedings of the 17th on Uncertainty in Artificial Intelligence (UAI–01)*, pages 253–260. Morgan Kaufmann.

Leslie, D. S. and Collins, E. J. (2005). Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44:495–514.

Leslie, D. S. and Collins, E. J. (2006). Generalised weakened fictitious play. *Games and Economic Behavior*, 56:285–298.

Marden, J. R., Arslan, G., and Shamma, J. S. (2005). Joint strategy fictitious play with inertia for potential games. *Proceedings of the 44th IEEE Conference on Decision and Control (CDC '05)*, pages 6692–6697.

Marden, J. R. and Shamma, J. S. (2008). Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Submitted to Games and Economic Behavior*.

Marden, J. R. and Wierman, A. (2008). Distributed welfare games. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC–08)*.

Marden, J. R., Young, H. P., Arslan, G., and Shamma, J. S. (2009). Payoff–based dynamics for multi–player weakly acyclic games. *SIAM Journal on Control and Optimization*, 48:373–396.

Monderer, D. and Shapley, L. S. (1996a). Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68:258–265.

Monderer, D. and Shapley, L. S. (1996b). Potential games. *Games and Economic Behavior*, 14:124–143.

Papadimitriou, C. H. and Roughgarden, T. (2008). Computing correlated equilibria in multi–player games. *Journal of the ACM*, 55(3):14:1–14:29.

Rosenthal, R. W. (1973). A class of games possessing pure–strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67.

Scutari, G., Barbarossa, S., and Palomar, D. P. (2006). Potential games: A framework for vector power control problems with coupled constraints. In *31st IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06)*, volume 4, pages 241–244.

Singh, S. P., Jaakkola, T., Littman, M. L., and C. Szepesvári (2000). Convergence results for single–step on–policy reinforcement–learning algorithms. *Machine Learning*, 38(3):287–308.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press, Cambridge, MA.

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. *Engineers*, Part II:325–378.

Wolpert, D. H. and Tumer, K. (2002). Collective intelligence, data routing and Braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387.

Young, H. P. (1993). The evolution of conventions. *Econometrica*, 61:57–84.

Young, H. P. (1998). *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton University Press.

Young, H. P. (2009). Learning by trial and error. *Games and Economic Behavior*, 65:626–643.