Controlling the Crowd: Inducing Efficient Equilibria in Multi-Agent Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many real world systems such as traffic networks and ride-sharing taxi systems can be tackled using multi-agent reinforcement learning. In such settings, self-interested agents must learn how to interact with each other in a shared stochastic environment. However, current methods within multi-agent reinforcement learning generally lead to agents taking joint actions over time that produce welfare inefficient and globally suboptimal outcomes. To this end, we propose a new method in which a meta-agent modifies agents' rewards leading to convergence to policies that produce globally efficient outcomes in Markov games. Our method does not require agents to have a priori knowledge of their environment - both the meta-agent and the agents learn from interacting with it. Our theoretical results show that using our method, multi-agent reinforcement learning algorithms always produce efficient outcomes. We apply our method to solve a challenging problem within an application in economic systems with thousands of agents.

Introduction

Complex systems such as traffic networks, financial markets and swarm robotics involve many agents strategically interacting with each other. In these systems, self-interested agents take actions over time to maximise their own cumulative rewards that depend on the actions of other agents and the system state. Such systems are modelled as Markov games (MGs). In MGs however, stable outcomes (Nash equilibria) are in general, welfare inefficient and highly undesirable from a central planner's perspective [7]. Multi-agent reinforcement learning methods in general, do not guarantee convergence to efficient NEs that maximise social welfare (e.g. minimise travel time in traffic networks) or optimise external objectives (e.g. taxi-drivers maximising a firm's profit or agents in financial markets minimising systemic risk). Multi-agent reinforcement learning (MARL) methods [10, 22] converge to stable points (where policies do not change) that are also NEs of the given game. However, these algorithms are not guaranteed to converge to efficient equilibria. Consequently, devising methods that ensure convergence to efficient outcomes in MGs is a significant challenge from practical and theoretical standpoints [18].

We propose a new technique to tackle the issue of undesirable outcomes in MGs. Our method uses a meta-agent (MA) to modify agents' reward functions in such a way that ensures convergence to efficient outcomes. In particular, the MA uses black box optimisation to seek the optimal parameter of a parametric reward modifier. In this setup,neither the agent's nor the MA have knowledge of their reward nor transition functions and use MARL techniques to learn them, which permits application to a broad range of problems. We prove theoretical results that demonstrate that for a class of MGs known as Markov potential games (MPGs) the MA's modifications to the game produces a continuous family of NE outcomes. This is a crucial property that allows the MA to use black-box optimisation techniques to find the reward modifications that induce desirable behaviour in the agents. Since the

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

reward modifier influences the potential function - a function that is maximised by all agents' NE strategies, in principle the MA can induce the desired behaviour in any number of agents.

Markov potential games (MPGs) frequently arise in engineering and economics where agents compete for a common resource as in the case of spectrum sharing in wireless communications, oligopoly (market share), transportation networks, ride-sharing applications, supply-chain management, electric power grids or cloud computing [21, 25]. Potential games are also ubiquitous in classical game-theory; *the prisoner's dilemma*, *the battle of the sexes*, *selfish routing games*, *congestion games* and *team games* are all potential games [11, 17].

Contributions. i) We propose an algorithmic framework which involves an MA that learns to modify the rewards within an MPG to optimise system performance, in both cases in which the NE can be altered and cases in which it must be preserved. **ii)** We show that the MG modified by the MA is an MPG, and that the NE set of the new game is continuous on the reward modifications, which allows us to prove existence of an optimal reward modifier. We then formulate the problem in a manner that allows to prove convergence to the reward modifier that induces efficient NE. We provide an approximation bound when the optimal reward modifier is estimated with a method that has low computational complexity. **iii)** We illustrate the framework in a set of experiments that tackle a challenging application: a logistic problem involving a system with 2,000 agents.

Related Work. Our work relates to mechanism design (MD) [14] and its dynamic and learning variants [23]. These incomplete information models analyse the problem of constructing a *mechanism* - a system of rewards and transfers, among self-interested strategic agents that have private information about their reward functions. The problem is to incentivise truth-revealing announcements from the agents. A well-known result in MD rules out (strategy-proof) mechanisms that induce the desired agent behaviour for general agent reward functions [19]. Therefore, in MD, agents' reward functions are (typically) limited to quasi-linear functions that are known up front [14]. Our framework permits a general rewards beyond quasi-linear functions. Moreover, since computing gradients is not required as in, for example [8], we tackle problems when the reward functions are unknown to the agents (and the MA) enabling us to solve complex and analytically intractable systems.

This work relates to leader-follower games (L-FGs) - sequential games in which a leader moves in advance of other agent(s) or *follower(s)*, who each select a best response (BR) strategy [2, 24]. However, in L-FGs, the leader cannot induce efficient outcomes i.e. maximise its own objective (e.g. ex. 98.1 in [15]) since the leader's reward is a function over a fixed joint action set. In our framework however, the MA's reward is determined by the agents' joint actions which are taken after the MA has made a choice of reward functions over a space of continuous functions.

This topic relates to reward shaping through which a reward is added with the aim of inducing convergence to a more desirable equilibrium [2, 6]. The majority the reward shaping literature is concerned with *potential based* reward shaping (PBRS). PBRS leaves the NE set unaltered and does not guarantee convergence to more efficient equilibria [5]. A number of papers handle non-potential based rewards shaping e.g. [16], however, such papers are limited in scope since they consider only specific normal form games settings e.g. the stag hunt game[1]. We tackle the MG case which adds considerable complexity to the problem since it requires a method of incentivising *sequences* of state-action pairs (trajectories) in a stochastic environment. In addition to the case for which the NE is preserved, our framework covers cases for which the MA alters the NE set so that the behaviour of rational agents aligns with some external objective.

Preliminaries

Let $\mathcal{N} \triangleq \{1, \dots, N\}$ denote the (possibly infinite) set of agents where $N \in \mathbb{N} \times \{\infty\}$. An MG is a tuple: $\mathcal{G} = \langle \mathcal{N}, (\gamma_i)_{i\in\mathcal{N}}, \mathcal{S}, (\mathcal{U}^i)_{i\in\mathcal{N}}, P, (R_i)_{i\in\mathcal{N}} \rangle$ which can be described as follows: at each time step $k = 1, 2, \dots T \in \mathbb{N} \times \{\infty\}$, the state of the system is given by $s \in \mathcal{S} \subseteq \mathbb{R}^p$ for some $p \in \mathbb{N}$. The game is equipped with an action set $\boldsymbol{\mathcal{U}} = \times_{i\in\mathcal{N}} \mathcal{U}^i$ – a Cartesian product of each agent's action set $\mathcal{U}^i$. Each set $\mathcal{U}^i$ is a compact, non-empty action set for each agent $i \in \mathcal{N}$. We define by $\mathcal{U}^{-i} = \times_{j\in\mathcal{N}\setminus\{i\}} \mathcal{U}^j$ - the Cartesian product of all agents' action sets except agent i. At each time step, the next state of the game is determined by a probability distribution $P : \mathcal{S} \times \boldsymbol{\mathcal{U}} \times \mathcal{S}$ so that $P(\cdot|s, \boldsymbol{u})$ gives the probability distribution over next states given a current state s when the agents take a joint action $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}$. When the environment is at state s and the agents take action $\boldsymbol{u}$, each

[1] In [16] some experiments on repeated games are performed but no theoretical analysis is provided.

agent i receives a reward computed by the function $R_i : \mathcal{S} \times \mathcal{U}^i \times \mathcal{U}^{-i} \to \mathbb{R}$. The term $\gamma_i \in [0, 1[$ is each agent i's discount factor. Each agent has a stochastic policy $\pi^i : \mathcal{S} \times \mathcal{U}^i \to \mathbb{R}^+$ - a conditional distribution over the action set given the current state. Let Π^i be a non-empty set of stochastic policies over $\mathcal{S} \times \mathcal{U}^i$ such that $\pi^i \in \Pi^i$. We denote by $\boldsymbol{\Pi}$ the set of policies for all agents i.e. $\boldsymbol{\Pi} \triangleq \times_{i \in \mathcal{N}} \Pi^i$, where each π^i, and by $\Pi^{-i} \triangleq \times_{j \in \mathcal{N} \setminus \{i\}} \Pi^j$. For simplicity, we assume $\Pi^j = \Pi^i, \forall i \neq j$. The joint policy of all agents is denoted by $\boldsymbol{\pi} = \left(\pi^i\right)_{i \in \mathcal{N}} \in \boldsymbol{\Pi}$, while the joint policy of all but the i-th agent is denoted $\pi^{-i} = \left(\pi^j\right)_{j \in \mathcal{N} \setminus \{i\}}$. We will sometimes write $\boldsymbol{\pi} = \left(\pi^i, \pi^{-i}\right)$ for any $i \in \mathcal{N}$.

Each agent $i \in \mathcal{N}$ uses a *value function*, $v_i^{\boldsymbol{\pi}} : \mathcal{S} \times \boldsymbol{\Pi} \to \mathbb{R}$, as its objective function:

$$v_i^{\boldsymbol{\pi}}(s) = \mathbb{E}\Big[\sum_{t=0}^{T} \gamma_i^t R_i(s_t, u_{i,t}, u_{-i,t}) \Big| \boldsymbol{u}_t \sim \boldsymbol{\pi}(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, \boldsymbol{u}_t), s_0 = s\Big], \tag{1}$$

where $\boldsymbol{u}_t = (u_{i,t}, u_{-i,t})$ is the joint action at time t. We now give some essential definitions:

Definition 1. *The policy $\pi^i \in \Pi^i$ is a BR policy against $\pi^{-i} \in \Pi^{-i}$ if: $\pi^i \in \operatorname{argmax} v_i^{(\widetilde{\pi}^i, \pi^{-i})}$.*

A Markov-Nash equilibrium (M-NE) is the solution concept for MGs in which every agent plays a BR against other agents. A M-NE is defined by the following:

Definition 2. *A strategy profile $\boldsymbol{\pi} = \left(\pi^i\right)_{i \in \mathcal{N}} \in \boldsymbol{\Pi}$ is an M-NE if $v_i^{\left(\pi^i, \pi^{-i}\right)}(s) \geq v_i^{\left(\pi'^i, \pi^{-i}\right)}(s)$, $\forall \pi'^i \in \Pi$, $\forall \pi^{-i} \in \Pi^{-i}$, $\forall s \in \mathcal{S}$, and $\forall i \in \mathcal{N}$.*

The M-NE condition ensures no agent can improve their rewards by deviating unilaterally from their current strategy. We define $NE\{\mathcal{G}\}$ as the set of M-NE for the game $\mathcal{G}$.

Definition 3. *An MG is called an exact MPG or an MPG for short, if there exists a function $\Phi : \mathcal{S} \times \boldsymbol{\Pi} \to \mathbb{R}$ such that:*
$v_i^{\left(\pi^i, \pi^{-i}\right)}(s) - v_i^{\left(\pi'^i, \pi^{-i}\right)}(s) = \Phi^{\left(\pi^i, \pi^{-i}\right)}(s) - \Phi^{\left(\pi'^i, \pi^{-i}\right)}(s)\ \forall \pi'^i \in \Pi^i,\ \forall \pi^{-i} \in \Pi^{-i}, \forall s \in \mathcal{S},\ \forall i \in \mathcal{N}$

Note that $\Phi^{\boldsymbol{\pi}}(s)$ gives the same value for all agents. We use $\mathcal{G}_{\text{mpg}}$ to denote an MPG. In the rest of the paper, we focus exclusively on MPGs.

The Framework

We now describe how the MA modifies the MG played by the agents. The problem is arranged into a hierarchy of the MA's problem and the set of agents' subproblem.

The agents' subproblem consists of solving the Markov game $\mathcal{G}(\boldsymbol{w}) = \langle \mathcal{N}, (\gamma_i)_{i \in \mathcal{N}}, \mathcal{S}, (\mathcal{U}^i)_{i \in \mathcal{N}}, P, (R_{i,\boldsymbol{w}})_{i \in \mathcal{N}} \rangle$ i.e. finding $\boldsymbol{\pi} \in NE\{\ \mathcal{G}(\boldsymbol{w})\}$ where $\boldsymbol{w}$ is chosen by the MA. Now each agent $i \in \mathcal{N}$ has a value function $v_i^{\boldsymbol{\pi}, \boldsymbol{w}} : \mathcal{S} \times \boldsymbol{\Pi} \times \boldsymbol{W} \to \mathbb{R}$ given by:

$$v_i^{\boldsymbol{\pi}, \boldsymbol{w}}(s) = \mathbb{E}\Big[\sum_{t=0}^{T} \gamma_i^t R_{i,\boldsymbol{w}}(s_t, u_{i,t}, u_{-i,t}) \Big| \boldsymbol{u}_t \sim \boldsymbol{\pi}(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, \boldsymbol{u}_t), s_0 = s\Big] \tag{2}$$

The most natural alteration to an agent's reward function is for it to be modified additively by a *modifier function* $\Theta : \mathcal{S} \times \mathcal{U}^i \times \mathcal{U}^{-i} \times \boldsymbol{W} \to \mathbb{R}$ s.th. the agents' modified reward function becomes:

$$R_{i,\boldsymbol{w}}(s_t, u_{i,t}, u_{-i,t}) \triangleq R_i(s_t, u_{i,t}, u_{-i,t}) + \Theta(s_t, u_{i,t}, u_{-i,t}, \boldsymbol{w})$$

where $R_i : \mathcal{S} \times \mathcal{U}^i \times \mathcal{U}^{-i} \to \mathbb{R}$ is the game 'intrinsic reward' that cannot be modified by the MA.

Note that the modifier function includes cases for which $\Theta(\cdot, u_{-i,t},) = \Theta(\cdot, u'_{-i,t}),\ \forall u_{-i,t} \neq u'_{-i,t} \in \mathcal{U}^{-i}$ in which case the modifier function adds rewards that do not depend on actions other than those taken by agent i.

The meta-agent's problem consists of a tuple $P_{\text{MA}} \triangleq \langle \boldsymbol{w}, R_{\text{MA}} \rangle$ where $\boldsymbol{w} \in \boldsymbol{W} \subset \mathbb{R}^l$ ($l \in \mathbb{N}$) is a set of vector of real-valued parameters over a space of parametric uniformly continuous functions and R_{MA} is the reward function for the MA. The MA's problem is to find Θ (i.e. the vector of parameters $\boldsymbol{w}$) that maximises the following:

$$J(\boldsymbol{w}, \boldsymbol{\pi}) = \mathbb{E}\big[R_{\text{MA}}(\boldsymbol{w}, \boldsymbol{\pi})\big], \tag{3}$$

whilst satisfying the M-NE condition which ensures that the agents play BR policies. Hence the MA's problem is:

$$\underset{\boldsymbol{w}\in\boldsymbol{W}}{\text{maximise}}\ J(\boldsymbol{w},\boldsymbol{\pi})\text{s.t. } v_i^{(\pi^i,\pi^{-i}),\boldsymbol{w}}(s) \geq v_i^{(\pi'^i,\pi^{-i}),\boldsymbol{w}}(s), \forall i \in \mathcal{N}\ , \forall \pi_i' \in \Pi^i\ , \forall \pi^{-i} \in \Pi^{-i}\ , \forall s \in \mathcal{S}.$$

The formulation describes numerous problems within economics, logistics and computer science including revenue management (e.g., ticket pricing), congestion management, and network design problems (e.g. tolling) [13, 4]. We consider two main cases, depending on the MA's goal:

1. **Trajectory targeted**: The MA's payoff is a function of the state trajectories produced by the agent's policies in the MG; i.e., $J(\boldsymbol{w},\boldsymbol{\pi}) \triangleq \mathbb{E}\big[R_{\text{MA}}(\boldsymbol{w}, X^{\boldsymbol{\pi}}, \zeta)\big]$, where $X^{\boldsymbol{\pi}}$ is Markov chain induced by the policy profile $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ in $\mathcal{G}(\boldsymbol{w})$ and ζ is an i.i.d. random variable which captures the noisiness in outcomes. An example is the KL divergence between the distribution of agent locations at every timestep, $D_t^a(\boldsymbol{w},\boldsymbol{\pi})$, and the target distribution of desired locations, $D_t^\star$: $R_{\text{MA}}^{(\text{tra})} = \sum_{t=0}^{T} \text{KL}(D_t^a(\boldsymbol{w},\boldsymbol{\pi}) \| D_t^\star)$. Applications include social planners seeking to minimise congestion in traffic networks through tolls, and firms seeking to smoothen electricity consumption in smart grids through dynamic pricing [4].
2. **Welfare targeted**: The MA's payoff is a function of the agents' joint rewards, that is, $J(\boldsymbol{w},\boldsymbol{\pi}) \triangleq \mathbb{E}\big[R_{\text{MA}}(\boldsymbol{w}, h(v_a^{\boldsymbol{\pi},\boldsymbol{w}}), \zeta)\big]$, for some uniformly continuous function h and $v_a^{\boldsymbol{\pi},\boldsymbol{w}} \triangleq (v_i^{\boldsymbol{\pi},\boldsymbol{w}})_{i\in\mathcal{N}}$. A simple example is the sum of agents' rewards i.e.: $R_{\text{MA}}^{(\text{soc})} = \sum_{i\in\mathcal{N}} v_i^{\boldsymbol{\pi},\boldsymbol{w}}$, resulting in the MA maximising social welfare. Other examples are oligopoly intervention e.g. fishery problems using optimal taxation [21] and worst-case optimisation (maxmin) problems (i.e. $h = -\mathbf{1}$).

The function Θ can be interpreted as a system of wealth transfers leading naturally to consider budgetary constraints. The function Θ can be interpreted as a system of wealth transfers leading naturally to consider budgetary constraints. If the modifier function satisfies :$\sum_{i\in\mathcal{N}} \sum_{t\leq T} \Theta(s_t, u_{i,t}, u_{-i,t}, \boldsymbol{w}) \leq 0,\ \ \forall s_t \in \mathcal{S}$, then the transfer of wealth is constrained so that there is no net transfer of wealth from the MA to the agents.

Note that the MA problem is a bilevel optimisation problem (specifically, a mathematical program with equilibrium constraints). Such problems are generally highly non-convex and the feasible regions might be unconnected and for this reason, such problems are in general highly intractable using analytic methods in all but simple cases (e.g. linear rewards) [3].

In the next section, we overcome these issues by expressing the NE constraint in terms of the potential function, and show that MARL methods can be applied to compute the set of NE for the agents' subgame, so that we can ensure feasibility for the MA problem without requiring closed analytic solutions. We then give a constructive formulation that allows to prove convergence to such an optimal solution. Finally, we provide an approximation bound when the optimal reward modifier is approximated with a truncated power series. We proceed to explain the details.

Theoretical Analysis

We now show that $\mathcal{G}(\boldsymbol{w})$ is an MPG, which enables $NE\{\mathcal{G}(\boldsymbol{w})\}$ to be described in terms of local maxima of function (as opposed to fixed points).

It is necessary to show that the game produced after the MA alters the agents' rewards is still potential.

Lemma 1. *The game $\mathcal{G}(\boldsymbol{w})$ is an MPG.*

Corollary 1. *The following expression holds* $\{\text{argmax}_{\boldsymbol{\pi}\in\boldsymbol{\Pi}}\, \Phi^{\boldsymbol{\pi},\boldsymbol{w}}(s), \forall s \in \mathcal{S}\} \subseteq NE\{\mathcal{G}(\boldsymbol{w})\}$.

Cor. 1 expresses that in playing their BR strategies $\mathcal{G}(\boldsymbol{w})$, each agent inadvertently maximises $\Phi^{\boldsymbol{\pi},\boldsymbol{w}}$, so the function $\Phi^{\boldsymbol{\pi},\boldsymbol{w}}$ is a potential of $\mathcal{G}(\boldsymbol{w})$.

We now prove that $NE\{\mathcal{G}(\boldsymbol{w})\}$ is continuous on $\boldsymbol{w}$, which is required for the use of black-box optimisation to maximise MA's objective. The following result establishes the continuity in MA's reward under changes in $\boldsymbol{w} \in \boldsymbol{W}$ which underpin the existence of a solution for MA's problem and a method for computing the solution. We begin by demonstrating that small changes in MA's action lead only to small changes in the game, in particular, the game itself is continuous in $\boldsymbol{w}$:

Proposition 1. *Given metric space* $\mathbf{X}$*, let* $B_\alpha(\boldsymbol{x}) \triangleq \{\boldsymbol{y} \in \mathbf{X} : \|\boldsymbol{x} - \boldsymbol{y}\| < \alpha\}$ *denote the open ball with radius* $\alpha > 0$ *around* $\boldsymbol{x} \in \mathbf{X}$*. Then for any* $\epsilon > 0$*,* $\exists \delta > 0 : \boldsymbol{w}' \in B_\epsilon(\boldsymbol{w}) \implies \boldsymbol{x}' \in B_\delta(\boldsymbol{x})$*, for any* $\boldsymbol{x}' \in NE\{\mathcal{G}(\boldsymbol{w}')\}$.

Now, we establish the existence of an optimal reward modifier $\boldsymbol{w}^\star \in \boldsymbol{W}$ that solves MA's problem, i.e. $\boldsymbol{w}^\star \in \boldsymbol{W}$ maximises $J(\boldsymbol{w}, \boldsymbol{\pi})$ and thus induces an efficient NE.

Theorem 1. *For* $\mathcal{G}(\boldsymbol{w})$ *there exists a value* $\boldsymbol{w}^\star \in \boldsymbol{W}$ *that maximises MA's reward function* R_{MA}.

Previous results hold for an arbitrarily expressive modifier function Θ. In practice, it is computationally efficient to express Θ using a representation with few parameters. The following bounds MA's loss when the modifier function is approximated by a truncated power series:

Theorem 2. *Let* $\boldsymbol{w}^\epsilon(n) \in \boldsymbol{W}$ *approximate solution to MA's problem for* $\mathcal{G}(\boldsymbol{w})$ *which is generated by an* $n-$*order series expansion, define MA's approximation loss by* $\mathcal{L} \triangleq J(\boldsymbol{w}^\star, \boldsymbol{\pi}) - J(\boldsymbol{w}^\epsilon(n), \boldsymbol{\pi})$*, then* $\mathcal{L}$ *is subject to the following bound:* $\max_{\boldsymbol{w}' \in \boldsymbol{W}, \boldsymbol{\pi}' \in \boldsymbol{\Pi}} \left| D^{N+1} J(\boldsymbol{w}', \boldsymbol{\pi}(\boldsymbol{w}')) \right|$.

The solution $\boldsymbol{w}^\star$ can be closely approximated by a truncated series expansion (other expansions e.g. neural networks are possible) reducing the number of parameters to be computed.

The issue of how to compute $\boldsymbol{w}^\star$ remains. In the following section we demonstrate that $\boldsymbol{w}^\star$ can be computed using black-box optimisation and MARL.

Solution Method

In our problem, the function R_{MA}, its gradient, the function h and $v_a^{\boldsymbol{\pi}, \boldsymbol{w}}$ are all unknown to the MA, who solely observes its realised rewards for each candidate $\boldsymbol{w}$ which suggests a black-box optimisation method. The unknown payoff, J, is treated as a random function with some prior belief over the space of functions. After observing the value of $J(\boldsymbol{w}_k, \boldsymbol{\pi})$ for some $\boldsymbol{w}_k \in \boldsymbol{W}$, the belief is updated to form a posterior distribution which is used to construct an acquisition function (e.g., expected improvement) that indicates which parameter $\boldsymbol{w}_{k+1}$ should be evaluated next, guiding exploration over $\boldsymbol{W}$. Similarly the agents do not know the components $\mathcal{G}(\boldsymbol{w})$ but merely observe their individual rewards after their joint policy $\boldsymbol{\pi}$ is played, we therefore use MARL to solve the game. The agents sample trajectories of experience tuples $(s_t, \boldsymbol{u}_t, (R_{i,\boldsymbol{w}_k}(s_t, \boldsymbol{u}_t))_{i \in \mathcal{N}}, s_{t+1})$, which are used to estimate the joint value function, $v_a^{\boldsymbol{\pi}, \boldsymbol{w}}$. Then, they update their policies by performing stochastic gradient ascent. The optimisation objective is nested; the MA chooses $\boldsymbol{w}$ of $\mathcal{G}(\boldsymbol{w})$ and the agents select a joint policy which generates a reward signal for the MA. Simultaneous updates of both the MA parameters and the agents' policies, in general, lack converge guarantees due to non-stationarity. Therefore, in order to compute the solution iteratively, after an initial choice by the MA, we let the MARL algorithm run until convergence which fulfils the M-NE constraint for the MA's problem (c.f. Prop. 2); the MA receives feedback from the outcome of the game $\mathcal{G}(\boldsymbol{w})$, then updates its choice of $\boldsymbol{w}$. This results in an *inner-outer loop method*. We require an efficient optimisation algorithm. Clearly, BO is a candidate algorithm to allow us to scale the framework. BO is sample efficient and has strong theoretical guarantees for non-convex problems [20].

Convergence. Theorem 1 guarantees the existence of a solution for $\boldsymbol{w}^\star$. Guarantees for convergence of the inner loop of the algorithm are also required. Potential games have strong convergence guarantees with numerous of MARL algorithms e.g. fictitious play [9]. The following proposition provides this guarantee:

Proposition 2 (Convergence)**.** *The algorithm converges to a stable point, moreover the set of stable points of algorithm 1 correspond to M-NE for the MPG.*

Convergence of the inner loop is required to obtain the equilibria of the multi-agent system. Consequently, the method is subject to conditions under which MARL methods converge. In the class of games we consider, MARL methods have been shown in general, to converge to NE solutions [9, 12]. Note also that by Theorem 3, approximate solutions can be computed with a reduced number of parameters in the BO component for a given error bound.

Experiments: Controlling the massive crowd

Consider 2,000 agents each seeking to locate themselves at desirable points in space over a time horizon. The desirability of a region changes with time and decreases with the number of agents

located within the neighbourhood. In this setting, the resulting NE distribution is in general, highly inefficient (and may not conform to external objectives) due to agent clustering [12]. The problem encapsulates *spectrum sharing problems in wireless communications* [1] and models spatio-economics problems such as firms locating their supply with dynamic demand processes and taxi-fleets. To handle large strategic populations, we use an RL mean field game (MFG) framework [12].

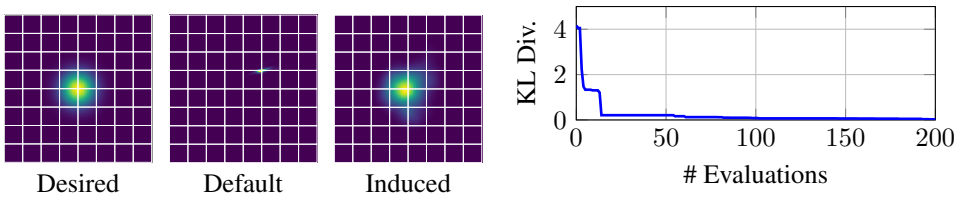

Figure 1: One shot case. (Top) Heat maps represent the MA's preferred distribution $M^\star$, the default agents' behaviour, and the influence of the MA's in the agents' distribution. (Bottom) Average KL divergences for each evaluation of the MA's BO outer loop.

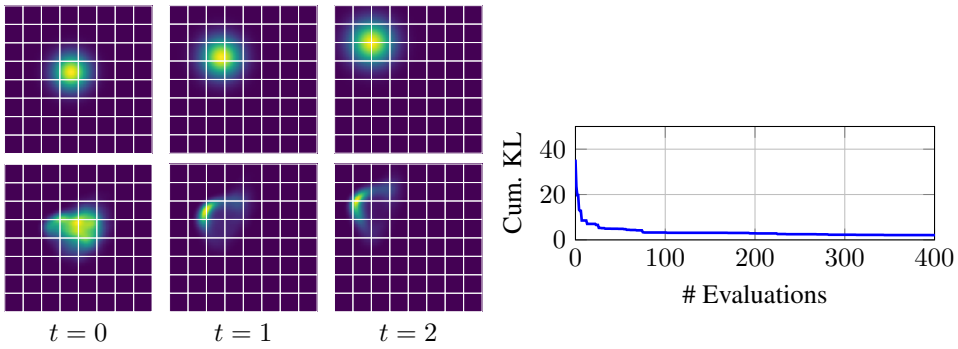

Figure 2: Dynamic case. (Top) Heat maps represent (first row) the MA's preferred distribution $M_t^\star$, (second row) the induced agent distribution M_t^a at time-steps $t = 0, 1, 2$. (Bottom) Average episodic cumulative KL divergences for each evaluation of the MA's BO outer loop (averaged over 100 independent tests per evaluation for 4 independent runs). Without the influence of the MA, the agents behave similar to the default behaviour displayed in Figure 1-Top middle.

We test our method in a *one-shot* game and in a *dynamic* game. Unlike current methods, our method does not require knowledge of the gradients and/or the reward functions. We observe, in Figure 1 and Figure 2 respectively, that in accordance with the theory, the agents learn to select policies that produce a distribution that matches $M^\star$ over the horizon of the problem.

In the **one-shot game** the MA seeks to induce a single agent distribution (as shown by the left heat map in Figure 1) - this is different from the distribution obtained when agents' behaviour is driven by their intrinsic reward function (central heat map in Figure 1). When the modifier function Θ in added to the agents' reward function, the average KL divergence converges almost to zero, i.e., the agents' distribution obtained with the MA framework (right heat-map in Figure 1) is almost the desired one. In the **dynamic game** the MA's desired distribution changes over time. In our experiment, $M_t^\star$ for $t = 0, 1, 2$ are as shown by the heat maps in the top row of Figure 2(left), while the bottom row presents the agents' distributions achieved with the MA framework. For the one-shot game, in Figure 2(right), we observe the average episodic cumulative KL divergences converge almost to zero.

Conclusion

In this paper, we introduce a meta-agent framework - a technique that enables self-interested adaptive learners to converge to efficient Nash equilibria in Markov games. By adding a modifier function to the agents' rewards, our method learns how to affect the rewards of self-interested agents to induce efficient system outcomes and thus producing convergence to desirable M-NE. We prove a continuity property in the meta-agent's modifications to the game which permits a broad range of black-box optimisation techniques to be applied. We demonstrated how the technique can be used to tackle problems in which the meta-agent can dramatically induce efficient outcomes in MARL.

References

[1] Ahmad, S.; Tekin, C.; Liu, M.; Southwell, R.; and Huang, J. 2010. Spectrum sharing as spatial congestion games. *Preprint arXiv:1011.5384*.

[2] Babes, M.; De Cote, E. M.; and Littman, M. L. 2008. Social reward shaping in the prisoner's dilemma. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, 1389–1392. International Foundation for Autonomous Agents and Multiagent Systems.

[3] Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research* 153(1):235–256.

[4] de Palma, A., and Lindsey, R. 2011. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies* 19(6):1377–1399.

[5] Devlin, S., and Kudenko, D. 2011. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 225–232. International Foundation for Autonomous Agents and Multiagent Systems.

[6] Devlin, S., and Kudenko, D. 2012. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 433–440. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Dubey, P. 1986. Inefficiency of nash equilibria. *Mathematics of Operations Research* 11(1):1–8.

[8] Dütting, P.; Feng, Z.; Narasimhan, H.; and Parkes, D. C. 2017. Optimal auctions through deep learning. *arXiv preprint arXiv:1706.03459*.

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

[10] Littman, M. L. 2001. Friend-or-foe Q-learning in general-sum games. In *Proceedings of the 18th International Conference On Machine Learning*, 322–328. Morgan Kaufmann, San Francisco, CA.

[11] Liu, M., and Wu, Y. 2008. Spectrum sharing as congestion games. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 1146–1153. IEEE.

[12] Mguni, D.; Jennings, J.; and de Cote, E. M. 2018. Decentralised learning in systems with many, many strategic agents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-18)*.

[13] Mohsenian-Rad, A.-H.; Wong, V. W.; Jatskevich, J.; Schober, R.; and Leon-Garcia, A. 2010. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE transactions on Smart Grid* 1(3):320–331.

[14] Nisan, N., and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior* 35(1-2):166–196.

[15] Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. MIT Press.

[16] Peysakhovich, A., and Lerer, A. 2017. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv:1709.02865*.

[17] Quang Duy, L.; Yong Huat, C.; and Boon-Hee, S. 2016. *Potential Game Theory: Applications in Radio Resource Allocation*. Springer Publishing Company, Incorporated, 1st edition.

[18] Roughgarden, T., and Tardos, E. 2007. Introduction to the inefficiency of equilibria. *Algorithmic Game Theory* 17:443–459.

[19] Satterthwaite, M. A. 1975. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10(2):187 – 217.

[20] Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and de Freitas, N. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 104(1):148–175.

[21] Slade, M. E. 1994. What does an oligopoly maximize? *The Journal of Industrial Economics* 45–61.

[22] Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; and Vicente, R. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PloS one* 12(4):e0172395.

[23] Tang, P. 2017. Reinforcement mechanism design. In *Early Carrer Highlights at Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI, pages 5146–5150*.

[24] Tharakunnel, K., and Bhattacharyya, S. 2007. Leader-follower semi-markov decision problems: theoretical framework and approximate solution. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, 111–118. IEEE.

[25] Valcarcel Macua, S.; Zazo, J.; and Zazo, S. 2018. Learning parametric closed-loop policies for markov potential games. In *To appear in Proceedings of the Sixth International Conference on Learning Representations (ICLR)*.