Efficient Reinforcement Dynamic Mechanism Design

David Mguni
PROWLER.io
Cambridge, UK
davidmg@prowler.io

ABSTRACT

Combining reinforcement learning (RL) with mechanism design (MD) holds the potential for efficient optimisation of mechanisms within online ad auctions, smart grid pricing and automated tolling among others. However, current attempts to integrate RL with MD have failed to produce scalable methods that are both sample efficient and, incorporate strategic reasoning among agents. Existing methods require incorporating equilibrium solutions to the agents' decision problem before a mechanism choice can be evaluated. Furthermore, optimisation programs with equilibrium constraints are generally intractable. To this end, we introduce a powerful method that combines stochastic optimisation and RL within MD to efficiently compute optimal, incentive compatible (IC) mechanisms. This combined framework exploits previously seen mechanisms to guide exploration of mechanism parameters leading to a highly sample efficient approach which speeds up optimisation. Our method incorporates a model that uses computerised learning agents to simulate equilibrium response to the mechanism parameters which in turn, are updated. This, as we show, guarantees convergence to incentive-compatible mechanisms. In contrast to classical MD which agents are assumed to perform optimal decision-making, we embed cognitive constraints faced by real-world agents using information-theoretic principles. We prove theoretical results that give convergence guarantees.

1 INTRODUCTION

Mechanism design (MD) is a mathematical formalism which studies how to induce desirable outcomes in systems with self-interested agents who are privately informed about their own preferences [17]. Over the past decade there has been a surge in interest in understanding how to construct mechanisms that maximise revenue in auction-based settings. Driving this increase is the application of MD to optimise sponsored search auctions which enable (online) platforms (e.g. Facebook, Baidu) to monetise advertising space [8]. The role of the mechanism, which takes the form of an auction, is to allocate the impressions for the site to advertisers that pay for the advertising space. Such mechanisms are capable of being applied to compute pricing in smart grids [21] and ride-sharing in *Uber-like* scenarios [7].

We propose a sample efficient technique that converges optimal mechanisms in unknown environments. We tackle *dynamic* MD so that the mechanism designer and agents are faced with future uncertainty and in which information about the agents' preferences and/or environment changes over time [19]. Our framework therefore captures problems in which agents can acquire knowledge of their preferences through repeated interactions (*learning-by-doing*) and dynamic settings in which preferences change after interactions with the mechanism for example, an agent's demand for a good changing after a sequence of purchases. This permits application to a broad range of problems with complex and unknown reward functions.

Using reinforcement learning (RL), our framework learns a model of agent behaviour in response to a chosen mechanism, the mechanism parameters are then updated *concurrently* which, as our theory demonstrates, converges to an optimal mechanism. In our framework the mechanism designer (**M**) makes adjustments to a space of of mechanisms in a *simulated environment*. Concurrently, the set of adaptive agents learn optimal best-response policies for the chosen mechanism. The simulated feedback avoids the need for costly acquisition of data from real-world environments but however ensures the generated agent behaviour is consistent with real-world outcomes. In our framework, **M** need not have *a priori* knowledge of its reward function but can simply observe its realised rewards after its decisions.Note that in the setting, the effect of changes to the mechanism on the outcome is also a priori unknown to **M**.

Designing mechanisms that maximise some objective[1] within *unknown environments* is currently a formidable challenge. In general the MD problem is $\mathcal{NP}$ hard [5]. A notable example is revenue maximisation in auctions - in settings such as online ad-auctions, bidders' valuations are typically drawn from a set of unknown distributions. Moreover, bidders may not know their own valuations up-front but may learn them only after repeated participation. Finding optimal mechanisms in these settings requires tuning numerous parameters to achieve optimal outcomes which ensure that agents are appropriately incentivised to participate (IR) and, that agents truthfully announce their preferences.

Additionally, finding an optimal mechanism requires imputing agents' (equilibrium) behaviour in response to a given mechanism to determine the mechanism performance. Consequently, determining optimal mechanisms in a sample efficient way is a pivotal challenge. This need is deepened when costs such as *menu costs* or the cost of evaluating many mechanisms in real-world scenarios are taken into account.

We show that by combining methods from stochastic approximation with RL, we can construct a sample efficient technique that produces fast convergence to an optimal mechanism for which IR and IC conditions are satisfied in unknown environments. A central result of the paper shows that modifications to the mechanism (performed by **M**) produce a *continuous family* of mechanism outcomes. This is a crucial property that permits *stochastic optimisation* methods to find the optimal mechanism parameters. This leads to a sample efficient approach which converges rapidly to the solution. Using RL, we then show that the set of adaptive agents converge to equilibrium policies which, as we show, provides convergence

[1] As in [10] we use the term *optimal mechanisms* to define a mechanism that maximises any given objective.

guarantees for both incentive compatible and individual rational mechanisms using our method.

To accurately model decision-making in real-world scenarios, using information-theoretic principles we for the first time, incorporate into the MD problem computational and cognitive limitations facing real-word agents. This departs from classical MD and allows us to accurately simulate the behaviour of real-world agents.

BACKGROUND

In classical MD, it is assumed that agents are fully rational, face no computational constraints and have perfect knowledge of their individual preferences and environment up-front. It is also assumed that agents' preferences and the information available to them do not change over time. Under these assumptions the celebrated Vickrey-Clarke-Groves (VCG) mechanism ensures constraints known as *incentive compatibility* (IC) and *individual rationality* (IR) [25] — IC ensures rational agents truthfully reveal their private information to the mechanism (e.g. bidders reveal their valuations to an auctioneer). IR ensures agents are suitably incentivised to participate (i.e. receive net positive returns in expectation). are satisfied. However, in many practical applications, the idealisations of classical MD are often violated, an example is the case when agents act in dynamic environments with changing information about their preferences. In these environments, the prescribed solutions of classical MD are rendered suboptimal [4].

Optimal MD and algorithmic MD seek to find the choice of mechanism that maximises some predefined mechanism objective [17]. Each of these problems has a *bilevel optimisation structure* in which given some choice of mechanism, the agents each choose a policy that maximises their given objective, given the chosen mechanism. **M** then optimises over some parameterised set of mechanisms. It is well known that for such problems, analytic solutions are notoriously difficult to obtain.

Learning methods tackling MD in unknown environments require acquiring performance feedback from agents in response to a mechanism choice. This is obtained either through direct interaction with real-world agents or by simulating the agents' responses for each choice of mechanism. Agents are therefore tasked with finding solutions to a decision problem, the outcomes of which are affected by the decisions of other agents. Consequently, generating a data point of the MD problem requires solving a decision problem (e.g. Markov game). In the absence of a relationship between the performance of each mechanism, current algorithmic methods require *pointwise-evaluation* of each mechanism to determine the optimal parameter. This leads to poor sample efficiency and slow convergence.

In [4], it is assumed that agents employ a type of *no-regret learning*, in which each agent's objective is to minimise its long-term regret. In addition to omitting the strategic reasoning by agents, constraints such as IR are neglected. In these models, the non-strategic aspect simplifies the analysis since the problem facing the agents is not strategic. However, with this idealisation, vital features of the problem such as coordination and in the influence of other bidders in auctions are neglected.

Recently, RL methods have been adopted to tackle the problem of optimal MD. In particular, *reinforcement MD* (RMD) applies learning algorithms drawn from RL to systematically compute mechanism parameters. These approaches have several deficiencies which render the solutions impracticable in various settings. RMD approaches such as [23] use responses generated by real-world agents to a choice of mechanism to evaluate the performance of a selected mechanism. This generates valuable data that describes the *actual* agents' response to a selected mechanism. This requires a costly collection of data from large numbers of mechanisms that have been implemented in real-world settings. Moreover, this approach does not ensure *strategy-proofness* - robustness against strategic reasoning by the agents and/or collusive behaviour. Therefore, agents may communicate false statements about their preferences to the mechanism.

A central challenge facing RMD is to produce methods that make use of data generated by previous actions to extract information about the performance of mechanisms with similar parameters [24]. This feature is crucial for data efficient exploration of mechanism parameters. We address this challenge in the paper with an alternative approach that is data efficient and fast to converge.

Contributions

We introduce a data-efficient approach that enables **M** to design an optimal mechanism from *simulated strategic equilibrium responses* from a set of agents. We also address a number of limitations of existing methods:

1. Computational efficiency. Our approach leads to a vast increase in convergence speed by allowing concurrent updates of both the agents' policies and the mechanism parameters. In particular, we use a *two timescales method* of stochastic approximation [2] to adjust mechanism parameters whilst updating the agents' polices that determine their announcements for a given mechanism.

2. Strategic behaviour. Our method uses a strategic formulation in which agents reason about other agents in their environment. To account for cognitive constraints, we introduce a form of BR derived from information-theoretic principles. By including an IR constraint directly to the agents' game, we construct mechanisms in which IR constraints are respected.

3. Sample efficiency. Central to our method is a result that demonstrates continuity in the mechanism outcome w.r.t. changes in the MD specification. This enables data generated by exploratory actions to be exploited, yielding an optimisation method that is highly data efficient.

Outline: First, we provide a formal description of the problem and extend the setting to include bounded rationality. We then prove the main theoretical results which are followed by some concluding remarks.

2 PRELIMINARIES

The system is comprised of self-interested agents that are each endowed with private information. The agents jointly make individual announcements to a mechanism over a series of rounds. After making their announcements, the agents then immediately receive an allocation and a payment (which constitute an *outcome*) which is determined by the mechanism. The dependence of the outcome on the agents' joint announcement results in a game being played by the agents. Formally, let $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$ denote a set of agents for some $N \in \mathbb{N}$. At each time $t \leq T$ where $T \in \mathbb{N} \cup \{\infty\}$,

the global state $s_t \in S = X \times \Theta$ is defined by $s_t = (s_{i,t})_{i\in\mathcal{N}}$ where $s_{i,t} = (x_{i,t}, \theta_{i,t}) \in S_i$, where $x_{i,t} \in X \subset \mathbb{R}^p$ for some $p \in \mathcal{N}$ is an *allocation* bequeathed to each agent $i \in \mathcal{N}$ and $\theta_{i,t} \in \Theta \subset \mathbb{R}^q$ for $q \in \mathbb{N}$ is the *type* for agent i at time t. The state space is $S = \times_{i=1}^{N} S_i$. An agent's type can encapsulate its preferences, experiences or its beliefs about the state of the world which can evolve with time. Examples are an agent's wealth after multiple interactions with the mechanism or how much of a good it possesses (for exhaustive discussions on types within MD, see for example [13]).

The variable $\theta_{i,t}$ evolves according to $\theta_{i,t} \sim F(\cdot|\theta_{i,t-1}, x_{i,t-1})$ where $F : \Theta \times X \to \Delta\Theta$ denotes the distribution of the random variable $\theta_{i,t}$. We denote by $\theta_t = (\theta_{i,t})_{i\in\mathcal{N}} \in \boldsymbol{\Theta}$ and $x_t = (x_{i,t})_{i\in\mathcal{N}} \in \boldsymbol{X}_t$. At each time t, each agent $i \in \mathcal{N}$ makes an *announcement* $\hat{a}_{i,t}$, where each possible announcement is an element within the type space Θ, the agent then receives an *allocation* $x_{i,t} \in X$ which is determined by an *allocation rule* $g_{w_1} : \boldsymbol{\Theta} \times \boldsymbol{X}_t \to \Delta\boldsymbol{X}_{t+1}$ where $w_1 \in W \subset \mathbb{R}^l (l \in \mathbb{N})$ is some parameter that is chosen by $\mathbf{M}$. Additionally, upon making its announcement, the agent receives *payment* which is determined by a *transfer rule* $p_{w_2} : \boldsymbol{\Theta} \to \mathbb{R}^{N+1}$ which performs transfers between agents (and the mechanism). Hence, the mechanism $\mathcal{M}$ has inputs $\hat{a}_t = (a_{1,t}, \ldots, \hat{a}_{n,t})$ which is the joint set of announcements made by the agents and, $\boldsymbol{X}_t$ which is the current allocation for the agents. Let us denote by $\boldsymbol{w} = (w_1, w_2) \in \boldsymbol{W}$, a mechanism $\mathcal{M}(\boldsymbol{w})$ is the tuple $\mathcal{M} = \langle g_{w_1}, p_{w_2}\rangle$. Note that the choice of mechanism is over a functional space which captures a large class of mechanism rules for which the transfer rule and allocation rule are measurable functions. We assume both p_w and g_w are drawn from parametric spaces; by the *universal approximation theorem* solutions to g_w and p_w that are close to some measurable functions can be obtained.

Agent i has a policy $\pi_i : [0, T] \times S_i \to \Delta\Theta$ which is a map from the time interval and states to announcements. We denote by Π_i the (non-empty and compact) set of stochastic policies for agent i and by $\boldsymbol{\Pi}$, the joint set of policies for all agents i.e. $\boldsymbol{\Pi} \triangleq \times_{j\in\mathcal{N}}\Pi_j$. We denote by $\Pi_{-i} \triangleq \times_{j\in\mathcal{N}\setminus\{i\}}\Pi_j$, the Cartesian product of the policy sets for all agents except agent $i \in \mathcal{N}$. Agent $i \in \mathcal{N}$ receives a reward determined by a function $R_i : \boldsymbol{W} \times S \times \boldsymbol{\Theta} \to \mathbb{R}$ which the agent seeks to maximise. Let us now define the following transition function $P : S \times S \to [0, 1]$ given by $P(s_{t+1}; \mathcal{M}(\hat{a}_t, x_t), F(\theta_t, x_t))$, since the outcome associated to each agent's choice of a announcement is affected by other agents' announcements, the agents *strategically interact*, hence, the above setup gives rise to a Markov game $\mathcal{G}(\boldsymbol{w}) = \langle \mathcal{N}, P, S, R_i(\boldsymbol{w}, \cdot), \gamma\rangle$, where $\gamma \in]0, 1]$ is the agents' (common) discount factor and the value function for each agent i is given by:

$$v_i^{\pi_i(\theta_i),\pi_{-i}(\theta_{-i})}(\boldsymbol{w}, s, \hat{a}) = \mathbb{E}_{\hat{a}_t\sim\boldsymbol{\pi}, s_t\sim P}\Big[\sum_{t\geq 0}\gamma^t R_i(\boldsymbol{w}, s_t, \hat{a}_t)|s = s_0\Big],\ \forall i \in \mathcal{N}.$$

We will refer to the game the agents play as the agents' subgame which we denote by $\mathcal{G}(\boldsymbol{w})$.

With a slight abuse of notation, we omit the dependence of the policy on x for the value functions. Similarly, though R depends on g_w, we express only its direct dependence on $\boldsymbol{w}$.

The agent model

In this framework, we construct a model of the agents' behaviour to simulate their response to choices of mechanisms. Each agent seeks to maximise their own expected cumulative reward. Each agent's problem is therefore given by:

$$\pi_i \in \arg\max_{\pi_i'\in\Pi_i} v_i^{\pi_i'(\theta_i),\pi_{-i}(\theta_{-i})}, \forall(\theta_i, \theta_{-i}) \in \Theta, \forall\pi_{-i} \in \Pi_{-i}. \quad (1)$$

The agent's optimisation implies that whenever the agents report their type using some reporting strategy $\pi_i \in \Pi$, the agents must be playing a strategy that maximises their reward. When (1) is satisfied for all $i \in \mathcal{N}$, no other available strategy increases their reward given the strategies of other agents. The resulting strategy profile $\boldsymbol{\pi}$ is a Nash equilibrium (NE) strategy. We denote the set of NE policies by $\mathcal{J}(\boldsymbol{w})$ for some $\boldsymbol{w} \in \boldsymbol{W}$. A mechanism $\mathcal{M}$ is said to be **incentive compatible (IC)** whenever (1) is satisfied.

The framework of this paper covers the following cases:
Dynamic MD: The agents' types evolve over time. The agents repeatedly interact with the mechanism and seek to maximise their expected cumulative rewards, for example agents may have a fixed budget over multiple interactions or acquire amounts of a divisible good over multiple rounds.
Learning by doing: Agents learn about their types and update their behaviour after repeated interactions. The agents base their current announcement on the allocation received and their announcement in the previous time step. An example of this type of interaction is e-commerce websites [15].
(Static) MD: This is a degenerate case in which the time horizon of the problem is one step e.g. a one-off auction. Here, agents make a single decision.

Extensions of the VCG mechanism to the dynamic case do not ensure IC - this is because the agents' intertemporal rewards depend on expected future announcements and mechanism payments so both deviations that are contingent on observed information and sequences of deviations are available to agents [12]. Moreover, in the VCG setup, agents are assumed to have immediate full knowledge of their own type and can compute the corresponding optimal behaviour. To relax these restrictions, we will formulate the problem facing $\mathbf{M}$ as a *(black-box) stochastic optimisation problem* over a parametric space of mechanisms and allow the agents to *learn* their optimal behaviour through a sequence of interactions.

We are now in a position to state the problem facing $\mathbf{M}$:
The Optimal Mechanism Design problem In this setup, the goal of the mechanism designer is to find a mechanism that maximises the mechanism deisgner's objective. In particular, the mechanism designer seeks to construct a mechanism which is composed of a parameterised pair of functions $g_{\boldsymbol{w}_1}$ and $p_{\boldsymbol{w}_2}$ which are the allocations rule and transfer rule. Hence, the optimal mechanism design problem is given by the following:

Find $\boldsymbol{w}^\star \in \boldsymbol{W}$ s.th.

$$\boldsymbol{w}^\star \in \text{argmax}\,\mathbb{E}_{\boldsymbol{\pi}}\big[G(\boldsymbol{\pi}, \boldsymbol{w})\big],\ \text{s.t. } \boldsymbol{\pi}^\star \in \mathcal{J}(\boldsymbol{w}), \quad (2)$$

Note that the choice of $\boldsymbol{w} \in \boldsymbol{W}$ fixes a component of the subgame played by the agents.

We refer to the function $G : S \times \boldsymbol{\Pi} \times W \to \mathbb{R}$ as the **mechanism objective function**. This function can be an external objective (that depends on the announcements and agents' allocations) e.g.

revenue in an auction or firm profit or the *joint welfare* of all agents i.e., $G = \sum_{i \in \mathcal{N}} v_i^{(\cdot)}$.

Therefore the task facing $\mathbf{M}$ is to find the parameter $\boldsymbol{w}$ that maximises G given that the agents play their NE strategies. In the setting we consider, $\mathbf{M}$ need not know it objective functions up front but we assume a sufficiently accurate proxy to R_i is available. We demonstrate that $\mathbf{M}$ can learn the parameter $\boldsymbol{w}^\star$ whilst the agents learn their NE strategies.

2.1 The Mechanism Design constraints

In MD, certain properties are prescribed to the mechanism to ensure satisfactory outcomes. A key mechanism objective is to induce *truthful revelations* so that the agents announce to the mechanism their private information. Truthful announcements are generally desirable since if the agents announce false preferences, then using their announcements, the mechanism may then enact suboptimal outcomes. Since the agents are rational, they act to maximise their self-interest.

As is standard, we focus on direct-revelation IC mechanisms - mechanisms for which the only actions available to the agents are to communicate claims about their types to $\mathbf{M}$. By the *Revelation Principle* [14], the same G found by solely studying direct-revelation IC mechanisms can be implemented by an arbitrary mechanism.

Recall that a mechanism $\mathscr{M} = \langle g_{w_1}, p_{w_2} \rangle$ is defined by a tuple that consists of an allocation rule $g_{w_1} : \boldsymbol{\Theta} \times \boldsymbol{X}_t \to \boldsymbol{X}_{t+1}$ and a transfer rule $p_{w_2} : \boldsymbol{\Theta} \to \mathbb{R}^{N+1}$ where $\boldsymbol{w} = (w_1, w_2) \in \boldsymbol{W}$ is a chosen by $\mathbf{M}$.

The following is a non-exhaustive set of properties for direct mechanisms $\forall \boldsymbol{w} \in \boldsymbol{W}, \forall (\pi_i, \pi_{-i}) = \boldsymbol{\pi} \in \mathcal{I}(\boldsymbol{w})$:

(1) Individual rationality (IR): $v_i^{\pi_i(\theta_i), \pi_{-i}(\theta_{-i})} \geq 0$.
(2) Implementability:
$\langle g_{w_1}(\boldsymbol{\pi}(\boldsymbol{\theta})), p_{w_2}(\boldsymbol{\pi}(\boldsymbol{\theta})) \rangle = \langle g_{w_1}(\boldsymbol{\theta}), p_{w_2}(\boldsymbol{\theta}) \rangle$.

IR implies that for each agent, the expected cumulative reward after participating is weakly positive hence, entering is beneficial. Implementability implies truthfully reporting their type is a solution to the agents' problem. In particular, when implementability is satisfied, the agents find it optimal to make announcements that truthfully reveal their types and, by the revelation principle, maximise the objective G.

We prove a series of theoretical results: first, we prove that the mechanism objective (for example, social welfare) is continuous w.r.t. changes in the mechanism design parameters - this enables the use gradient-based techniques for expedient computation of the optimal mechanism. We construct a method which uses RL to compute the agents' joint *equilibrium responses* to a chosen mechanism. We then show that this framework induces IC mechanisms. We then use a two-timescales method of stochastic approximation to show that how adaptive agents can learn their equilibrium strategies in response to mechanisms *whilst* the mechanism parameters are being updated. This yields a method by which the optimal mechanism parameters can be computed expediently using a simulated model of agents' responses, yielding mechanisms that ensure the IC and IR constraints are satisfied.

In our method, the mechanism selection occurs over a large space of functions that determine the allocation function and transfers for the mechanism. This enables $\mathbf{M}$ to make selections over a large range of mechanisms leading to outcomes that maximise $\mathbf{M}$'s objectives.

As remarked earlier, the IC condition in classical MD imposes strong assumptions on the ability of agents to reason about their environment and compute optimal solutions to complex decision problems with arbitrary accuracy. This assumption is often violated in practice. To this end, we now introduce an additional feature that enables more accurate descriptions of real-world multi-agents systems. We incorporate a form of cognitive constraints faced by real-world agents, this in turn enables us to simulate *boundedly rational* behaviour for any given choice of mechanism.

2.2 Bounded rationality

It is widely observed that the decisions made by agents within economic settings are affected by cognitive constraints which vastly alter system outcomes. This results in significant deviations in agent behaviour from that predicted within classical MD. To account for this, we embed cognitive constraints or *bounded rationality* (BR) within the agent's problem. We endow each agent's reward with a regulariser that acts to inhibit large immediate changes in agents' decision policies from some initial policy. This naturally embeds *decision inertia* in our model of the agents' responses. BR has been studied in RL literature [3] and, recently in multi-agent RL systems [9].

The following is a description of the agents problem when the agent exhibits bounded rationality:

Agent's BR objective: : Let $\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots,$ be a sequence of parameters in $\boldsymbol{W}$ s.th. $\boldsymbol{w}_k \to \boldsymbol{w}^\star$ as $k \to \infty$, then the agent's BR objective is given by the following $\forall i \in \mathcal{N}, \forall s_t \in S$:

$$\begin{aligned} &\max_{\pi_i \in \Pi} v_i^{\pi_i(\theta_i), \pi_{-i}(\theta_{-i})}(\cdot, (\hat{a}_i, \hat{a}_{-i})), \quad \hat{a}_i \sim \pi_i \\ &\text{s.t.} \qquad \sum_{t \geq 0} \mathbb{E}\Big[\gamma^t \mathrm{KL}(\pi_i(\cdot|s_t) \| \pi_0(\cdot|s_t))\Big] \leq \beta, \end{aligned} \tag{3}$$

for a given $\beta \in \mathbb{R}_{>0}$ where $(\pi_0)_{i \in \mathcal{N}} \in \mathcal{I}(\boldsymbol{w}_0)$ and where KL denotes the Kullback Leibler divergence, which is a distance measure between two statistical distributions. Hence, π_0 is the best-response policy for the agent against the first mechanism encountered. The constraint (3) captures the observed phenomenon of *anchoring* in which agents excessively rely on initial information presented [18].

In the limit $\beta \to \infty$, we recover the case of full rationality as a specific case, when $\beta = 0$ the agents fix their policy after their first interaction. The Lagrangian $\mathscr{L}$ corresponding to the problem for agent i with BR is given by:

$$\mathscr{L}(s, \hat{a}, \boldsymbol{w}) = \sum_{t \geq 0} \mathbb{E}_{\pi, s_t \sim P}\Big[\gamma^t R_i^\beta(\boldsymbol{w}, s_t, \hat{a}_t)\Big],$$

where $R_i^\beta(\boldsymbol{w}, s_t, \hat{a}_t) \triangleq R_i(\boldsymbol{w}, s_t, \hat{a}_t) - \frac{1}{\beta}\mathrm{KL}(\pi(\hat{a}_{i,t}|s_t) \| \pi(\hat{a}_{i,1}|s_1))$ for some $\beta \in \mathbb{R}$.

With IR, the Lagrangian $\mathscr{L}_{II}$ is: $\mathscr{L}_{II} = (1 + \lambda)\mathscr{L}$.

3 THEORETICAL ANALYSIS

In this section, we prove the convergence of our method to an optimal mechanism in which the agents enact best response policies. We prove that given dominant strategy implementation, the agents' subgame is reducible to a Markov decision process (MDP) and the

game exhibits a continuity property in the MD's parameters. The results enable stochastic gradient methods to compute the optimal choice of $\boldsymbol{w}$. The results are built under the following assumption:

Assumption 1. *The functions G and R_i are Lipschitz continuous in $\boldsymbol{w}$. i.e $\exists c_G, c_{R_i} > 0$ s.th. $\forall \boldsymbol{w}, \boldsymbol{w'} \in \boldsymbol{W}, \forall i \in \mathcal{N}$*

$$\|R_i(\boldsymbol{w}, \cdot) - R_i(\boldsymbol{w'}, \cdot)\| \leq C_{R_i} \|\boldsymbol{w} - \boldsymbol{w'}\|, \tag{4}$$

$$\|G(\boldsymbol{w}, \cdot) - G(\boldsymbol{w'}, \cdot)\| \leq C_G \|\boldsymbol{w} - \boldsymbol{w'}\|. \tag{5}$$

First, let us introduce a useful concept:

Definition 3.1. A policy $\pi_i^\star \in \Pi$ is a **dominant strategy** if the following holds $\forall i \in \mathcal{N}, \forall \theta_i \in \Theta, \forall (\pi_i, \pi_{-i}) \in \boldsymbol{\Pi}$:

$$v_i^{(\pi_i^\star(\theta_i), \pi_{-i})}(\cdot, (\pi_i^\star(\theta_i), \pi_{-i})) \geq v_i^{\pi_i, \pi_{-i}}(\cdot, (\pi_i(\theta_i), \pi_{-i})).$$

A dominant strategy is therefore a best-response strategy irrespective of the actions of other agents. Implementation in dominant strategies endows mechanisms with robust equilibrium outcomes. Dominant strategies do not require **M** to have knowledge of the agents' assumptions about each other's actions or *rationality*, nor is **M** required to make distributional assumptions about the agents' types. For these reasons, dominant strategy implementation is frequently used within MD [16]. For these reasons, in our analysis we assume for each agent, there exists a dominant strategy.

The following theorem is a key result:

Theorem 3.2 (Reducibility). *Each agent's problem is reducible to an MDP, that is, for any $i \in \mathcal{N}$ we have that:*

$$\left\{\pi_i^\star \in \arg\max v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, (s_{i,t}, \cdot), \theta'_{i,t})\right\} \subseteq \mathcal{F}(\boldsymbol{w}), \tag{6}$$

Theorem 3.2 establishes that the agents' NE policy for the subgame is obtainable from the solution to an MDP. This vastly reduces the difficulty of the problem since we can now safely circumvent analysis of the strategic component whilst ensuring we generate a solution to the strategic game.

We denote the set of policies $\pi \in \Pi$ that satisfy (6) by $\bar{\mathcal{F}}(w)$ for some $\boldsymbol{w} \in \boldsymbol{W}$. Using (2) and (6) we additionally observe that the MD problem now becomes:

Optimal Mechanism design problem II (MP II)

Find $\boldsymbol{w}^\star \in \boldsymbol{W}$ s.th.

$$\boldsymbol{w}^\star \in \arg\max \mathbb{E}_{\boldsymbol{\pi}}\left[G(\boldsymbol{\pi}, \boldsymbol{w})\right] \text{ s.t. } \pi^\star \in \bar{\mathcal{F}}(\boldsymbol{w}^\star),\ \forall i \in \mathcal{N}.$$

The MP II captures that the NE policies $\boldsymbol{\pi}^\star = (\pi_i^\star)_{i \in \mathcal{N}}$ that enter **M**'s problem are optimal policies of an MDP.

We now study the effect of modifying $\boldsymbol{w}$ on $\mathcal{F}(\boldsymbol{w})$. To use of gradient techniques, it is necessary to establish a form of continuity of the game $\mathcal{G}(\boldsymbol{w})$ in the parameter $\boldsymbol{w}$. To this end, we introduce a formal notion of continuity of $\mathcal{F}(\boldsymbol{w})$ w.r.t $\boldsymbol{w}$:

Definition 3.3. Given metric space X, let $B_\alpha(x) \triangleq \{\boldsymbol{y} \in X : \|x - y\| < \alpha\}$ denote the open ball with radius $\alpha > 0$ around $x \in X$. Then $x \in \mathcal{F}(\boldsymbol{w})$ is **essential** in $\boldsymbol{w}$ if for any $\epsilon > 0$, $\exists \delta > 0$: $w' \in B_\epsilon(\boldsymbol{w}) \implies x' \in B_\delta(x)\ \forall x' \in \mathcal{F}(\boldsymbol{w'})$.

The essentiality condition states that small changes in the parameter $\boldsymbol{w}$ lead to a small change in the set of IC policies.

Having formalised a notion of continuity, we are now in position to state the continuity result:

Theorem 3.4 (Essentiality). *The agents' subgame is essential in $\boldsymbol{w}$.*

Theorem 3.4 establishes that small changes in the mechanism parameter $\boldsymbol{w}$, lead to small changes in the equilibrium outcome of the agents' game. This provides information about the behaviour of the game in some neighbourhood for each chosen $\boldsymbol{w}$. Crucially, as we later show, this property is inherited by the mechanism objective function itself. Theorem 3.4 underscores the data efficiency of the method.

The following result is required for proving Theorem 3.4:

Proposition 3.5. *The MDP itself is Lipschitz continuous, in particular, there exists a constant $c > 0$ s.th.*

$$\|v_i^{\pi_1^\star(\theta)}(\boldsymbol{w}, \cdot) - v_i^{\pi_2^\star(\theta)}(\boldsymbol{w'}, \cdot)\| \leq c\|\boldsymbol{w} - \boldsymbol{w'}\|, \tag{7}$$
$$\forall i \in \mathcal{N}, \forall \boldsymbol{w}, \boldsymbol{w'} \in \boldsymbol{W}, \forall \pi_1^\star \in \bar{\mathcal{F}}(\boldsymbol{w}), \pi_2^\star \in \bar{\mathcal{F}}(\boldsymbol{w'}).$$

The Lipschitz continuity establishes that small changes in the parameter $\boldsymbol{w} \in \boldsymbol{W}$ lead to small changes in the value function of the game $\mathcal{G}(\boldsymbol{w})$.

Proof of Proposition 1

$$\begin{aligned}
&\left|v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, s, \cdot) - v_i^{\pi_i(\theta_i)}(\boldsymbol{w'}, s, \cdot)\right| \\
&= \left|\mathbb{E}\left[\max_{\pi \in \Pi}[R_i(\boldsymbol{w}, s, \cdot) + \gamma \sum_{s' \in S} p(s'|s, \cdot) v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, s', \cdot)\right]\right| \\
&\quad - \left|\mathbb{E}\left[\max_{\pi \in \Pi}[R_i(\boldsymbol{w'}, s, \cdot) + \gamma \sum_{s' \in S} p(s'|s, \cdot) v_i^{\pi_i(\theta_i)}(\boldsymbol{w'}, s', \cdot)\right]\right| \\
&\leq \max_{\pi \in \Pi} \left|R_i(\boldsymbol{w}, s, \cdot) - R_i(\boldsymbol{w'}, s, \cdot)\right| \\
&\quad + \gamma \sum_{s' \in S} p(s'|s, \cdot) \left|v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, s', \cdot) - v_i^{\pi_i(\theta_i)}(\boldsymbol{w'}, s', \cdot)\right|
\end{aligned}$$

Recall that $\gamma < 1$, we therefore find that

$$\begin{aligned}
&\left|v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, s, \cdot) - v_i^{\pi_i(\theta_i)}(\boldsymbol{w'}, s, \cdot)\right| \\
&\leq (1 - \gamma)^{-1} \max_{\pi \in \Pi} \left|R_i(\boldsymbol{w}, s, \cdot) - R_i(\boldsymbol{w'}, s, \cdot)\right| \\
&\leq c \left\|\boldsymbol{w} - \boldsymbol{w'}\right\|,
\end{aligned}$$

where $c \triangleq C_{R_i}(1 + \gamma)^{-1}$, which proves that v_i is Lipschtzian in $\boldsymbol{w}$. Therefore, v_i is uniformly continuous w.r.t. $\boldsymbol{w}$ so that $\forall \epsilon > 0,\ \exists \delta > 0$ s.th $\|\boldsymbol{w} - \boldsymbol{w'}\| < \epsilon \implies |v_i^{\pi_i(\theta_i)}(\boldsymbol{w}, s, \cdot) - v_i^{\pi_i(\theta_i)}(\boldsymbol{w'}, s, \cdot)| < \delta$, since this holds for any $\pi \in \Pi$, we fix $\pi_i \in \mathcal{F}(\boldsymbol{w})$ from which we deduce the result.

Lemma 3.6. *For each type, there is a unique optimal value function for any agent.*

The result follows from the uniqueness of the value function of the MDP induced by the agents' game and the *contraction mapping theorem* (see [1]).

We immediately deduce the following result:

Corollary 3.7. *For any $\boldsymbol{w} \in \boldsymbol{W}$, the set $\mathcal{F}(\boldsymbol{w})$ is a singleton.*

Further to reducing to an MDP, each agent's problem admits a closed form solution. We now turn to the solutions of the agents' problem. The following results provide closed expressions for the optimal strategies for the agents' subgame:

Proposition 3.8. *The optimal policy for the agents' problem is given by the following expression:*

$$\pi_i^\star(\cdot|s_t) = \frac{\pi_0(\cdot|s_t)\exp\left(\upsilon_i^{\pi_i,\pi_{-i}}(\boldsymbol{w},(\hat{a}_i,\hat{a}_{-i}))\right)}{\mathbb{E}_{\pi_0}[\exp \upsilon_i^{\pi_i,\pi_{-i}}\left((\boldsymbol{w},(\hat{a}_i,\hat{a}_{-i}))\right)]}. \tag{8}$$

The optimal policy for the agents' problem with IR is an analogous expression.

Proof of Proposition 2. *By the Bellman equation corresponding to the agent's reduced problem (MDP) we have:*

$$\begin{aligned}
\gamma + \upsilon_i^{\pi_i,\pi_{-i}} &= R_i(s_t,\cdot) + \mathbb{E}_{\boldsymbol{\pi},\cdot}[\upsilon_i^{\pi_i,\pi_{-i}}] - \mathrm{KL}(\pi_i\|\pi_0(s_t)) \\
&= R_i(s_t,\cdot) + \mathbb{E}_{\boldsymbol{\pi},\cdot}\left[\log\left(\frac{\pi}{\pi_0(s_t)}\right) + \upsilon_i^{\pi_i,\pi_{-i}}\right] \qquad (9) \\
&= R_i(s_t,\cdot) - \log\left(\mathbb{E}_{\pi_0}[\exp -\upsilon_i^{\pi_i,\pi_{-i}}]\right) \\
&\quad - \mathrm{KL}\left(\pi_i\middle\|\frac{\pi_0\exp\{-\upsilon_i^{\pi_i,\pi_{-i}}\}}{\mathbb{E}_{\pi_0}\left[\exp\{-\upsilon_i^{\pi_i,\pi_i}\}\right]\pi_0(s_t)}\right) \qquad (10)
\end{aligned}$$

The result follows after observing the policy π_i maximises the RHS of (10) i.e. minimises the KL divergence is that in (8). ■

Lemma 3.9. *G is Lipschitz continuous and almost everywhere (a.e.) differentiable in $\boldsymbol{w}$.*

Lemma 3.9 follows since the composite function $g_1\circ(\ldots\circ(g_n(\cdot)\ldots))$ of $n<\infty$ Lipshitzian functions $g_1,\ldots,g_n$ is Lipschitzian then applying Rademacher's lemma.

Proof of Lemma 3

Fix $\boldsymbol{w'}\in\boldsymbol{W}$ and $\boldsymbol{w}\in\boldsymbol{W}$ and define $y(\boldsymbol{w})\triangleq \upsilon_i^{\pi_i}(\boldsymbol{w},\cdot)$ and $y(\boldsymbol{w'})\triangleq \upsilon_i^{\pi_i}(\boldsymbol{w'},\cdot)$. By the Lipchiztianity of G,R, $\exists c_G,c_R>0$ s.th.

$$\begin{aligned}
\left|y(\boldsymbol{w})-y(\boldsymbol{w'})\right| &= \left|\mathbb{E}\left[\sum_{t\geq 0}\gamma^t\{R_i(\boldsymbol{w},\cdot)-R_i(\boldsymbol{w'},\cdot)\}\right]\right| \\
&\leq \mathbb{E}\left[\sum_{t\geq 0}\gamma^t\left|R_i(\boldsymbol{w},\cdot)-R_i(\boldsymbol{w'},\cdot)\right|\right] \\
&\leq c_R\sum_{t\geq 0}\gamma^t\left\|\boldsymbol{w}-\boldsymbol{w'}\right\| \\
&= \frac{c_R}{1-\gamma}\left\|\boldsymbol{w}-\boldsymbol{w'}\right\|.
\end{aligned}$$

Moreover, we can find constants $c_G,c>0$ s.th.

$$\begin{aligned}
&|G(y(\boldsymbol{w}))-G(y(\boldsymbol{w'}))| \\
&\leq c_G|y(\boldsymbol{w})-y(\boldsymbol{w'})| \\
&\leq c\|\boldsymbol{w}-\boldsymbol{w'}\|,
\end{aligned}$$

from which we deduce the thesis. ■

We now establish the existence of an optimal value $\boldsymbol{w}^\star\in\boldsymbol{W}$ that solves **M**'s problem.

Lemma 3.10. *There exists a maximum for the MD problem.*

Lemma 3.10 follows from the extreme value theorem. Crucially, Lemmas 3.9 and 3.10 imply that gradient methods can be applied to obtain $\boldsymbol{w}^\star\in\boldsymbol{W}$. As we discuss in the next section, Lemma 3.10 permits the application of a two timescales method which ensures convergence to an optimal policy while the mechanism parameters are updated toward $\boldsymbol{w}^\star$.

4 SOLUTION METHOD

Our method involves concurrent computation of the mechanism parameter $\boldsymbol{w}$ and the agents' policy parameters for the simulated game. In order to ensure the agents are playing IC strategies, it is necessary for the agents' policies to converge to their equilibrium strategies. In accordance with Theorem 3.2, the agents' equilibrium policies can be found by finding an optimal policy for an MDP, hence we seek the pair $(\pi^\star,\boldsymbol{w}^\star)$ where $\pi^\star\in\mathcal{F}(\boldsymbol{w}^\star)$ and $\boldsymbol{w}^\star\in\arg\max G(\boldsymbol{\pi},\boldsymbol{w})$. The iterative process $(\pi_1,\pi_2,\ldots;\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots)\rightarrow(\pi^\star,\boldsymbol{w}^\star)$ generates two stochastic approximation processes for the policy updates and updates for $\boldsymbol{w}_{k\geq 1}$ (see Algorithm 1). The algorithm involves

Inputs: **M**'s learning rate ξ, agents' learning rate $\eta\ll\xi$.

1: Initialise agents' strategy profile $\boldsymbol{\pi}_0$.
2: Initialise reward modifier parameter $\boldsymbol{w}_0$.
3: **while** true **do**
4: Each agent i updates its policy π_i using some RL algorithm with learning rate η.
5: **M** estimates $\widehat{\nabla}G(\boldsymbol{w},\boldsymbol{\pi})$.
6: **M** updates $\boldsymbol{w}$ using $\widehat{\nabla}G(\boldsymbol{w},\boldsymbol{\pi})$ with learning rate ξ.
7: **end while**
8: Return $\boldsymbol{w}$.

Algorithm 1: Two timescales RMD.

estimating the gradient $\widehat{\nabla}G(\boldsymbol{w},\boldsymbol{\pi})$ since both G and its gradients are unknown. Though standard approximation techniques (e.g. Kiefer-Wolfowitz) can be used, these require evaluation of two points of the objective to compute an estimate. We use a *one-point gradient estimation method* that requires that only a single random point of the objective function be evaluated. Using single *random point* evaluation is sufficient to approximate gradient descent [6].

To achieve convergence with concurrent updates, we use a *two timescales method* in which updates to $\boldsymbol{w}_k$ are performed with a lower learning rate than the policy updates, this generates a *quasi-static* appearance w.r.t. the policy updates. Under these conditions, the two update processes converge [2]. Two timescales methods have been applied to tackle convergence problems in actor-critic methods [11] and multi-agent RL [20]. To apply the method, we observe that by Lemma 3.6, for any $\boldsymbol{w}\in\boldsymbol{W}$, the MDP associated with the agents' problem has an asymptotically stable point.

Theorem 4.1. *[2] The two timescales method converges.*

The theorem involves a choice for the learning rates of the process - for complete details see [2].

To ensure the algorithm converges to an optimal solution for **M** both the updates to the policy π and $\boldsymbol{w}$ (namely $(\xi_j)_{j\in\mathbb{N}}$ and $(w_k)_{k\in\mathbb{N}}$) are required to converge. Lemma 3.10 guarantees the existence of a solution for $\boldsymbol{w}^\star$. Convergence of the RL algorithm is guaranteed using standard algorithms given Lemma 3.6 [1].

The following proposition provides this guarantee:

Proposition 4.2 (Convergence). *Algorithm 1 converges to a (local) maximum of G.*

5 CONCLUSION

In this paper, we showed that computing optimal mechanisms when agents' preferences and their distributions are unknown to

the agents and the mechanism. By proving continuity results, we showed that optimal mechanism parameters can be computed efficiently using stochastic approximation.

REFERENCES

[1] Bertsekas, D. P., and Tsitsiklis, J. N. 1995. Neuro-dynamic programming: an overview. In *Proc. 34th IEEE Conf. Decision and Control*, volume 1, 560–564. IEEE Publ. Piscataway, NJ.

[2] Borkar, V. S. 2009. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer.

[3] Braun, D. A.; Ortega, P. A.; Theodorou, E.; and Schaal, S. 2011. Path integral control and bounded rationality. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, 202–209. IEEE.

[4] Cai, Q.; Filos-Ratsikas, A.; Tang, P.; and Zhang, Y. 2018. Reinforcement mechanism design for e-commerce. In *Proc. 2018 World Wide Web Conf. World Wide Web*, 1339–1348. International World Wide Web Conferences Steering Committee.

[5] Conitzer, V., and Sandholm, T. 2002. Complexity of mechanism design. In *Proc. 18th Conf. Uncertainty in Artificial Intelligence*, 103–110. Morgan Kaufmann Publishers Inc.

[6] Flaxman, A. D.; Kalai, A. T.; and McMahan, H. B. 2005. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proc. 16th annual ACM-SIAM Symp. Discrete algorithms*, 385–394. Society for Industrial and Applied Mathematics.

[7] Furuhata, M.; Dessouky, M.; Ordóñez, F.; Brunet, M.-E.; Wang, X.; and Koenig, S. 2013. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological* 57:28–46.

[8] Gatti, N.; Lazaric, A.; Rocco, M.; and Trovò, F. 2015. Truthful learning mechanisms for multi-slot sponsored search auctions with externalities. *Artificial Intelligence* 227:93–139.

[9] Grau-Moya, J.; Leibfried, F.; and Bou-Ammar, H. 2018. Balancing two-player stochastic games with soft q-learning. In *IJCAI*.

[10] Hartline, J. D. 2013. Mechanism design and approximation. *Book draft. October* 122.

[11] Konda, V. R., and Tsitsiklis, J. N. 2000. Actor-critic algorithms. In *Advances in neural information processing systems*, 1008–1014.

[12] Ma, K., and Kumar, P. 2018. The strategic lqg system: A dynamic stochastic vcg framework for optimal coordination. *arXiv preprint arXiv:1803.06734*.

[13] Morris, S., and Shin, H. S. 2001. Global games: Theory and applications.

[14] Myerson, R. B. 1981. Optimal auction design. *Mathematics of operations research* 6(1):58–73.

[15] Nekipelov, D.; Syrgkanis, V.; and Tardos, E. 2015. Econometrics for learning agents. In *Proc. 16th ACM Conf. Economics and Computation*, 1–18. ACM.

[16] Nisan, N., and Ronen, A. 1999. Algorithmic mechanism design. In *Proc. 31st Annual ACM Symp. Th. Comp.*, 129–140. ACM.

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

[18] Northcraft, G. B., and Neale, M. A. 1987. Experts, amateurs, and real estate: An anchoring-and-adjustment perspective on property pricing decisions. *Organizational behavior and human decision processes* 39(1):84–97.

[19] Pavan, A.; Segal, I.; and Toikka, J. 2014. Dynamic mechanism design: A myersonian approach. *Econometrica* 82(2):601–653.

[20] Perkins, S., and Leslie, D. S. 2012. Asynchronous stochastic approximation with differential inclusions. *Stoch. Syst.* 2(2):409–446.

[21] Samadi, P.; Mohsenian-Rad, H.; Schober, R.; and Wong, V. W. 2012. Advanced demand side management for the future smart grid using mechanism design. *IEEE Transactions on Smart Grid* 3(3):1170–1180.

[22] Selten, R. 1990. Bounded rationality. *Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft* 146(4):649–658.

[23] Tang, P. 2017a. Reinforcement mechanism design. In *Early Carrer Highlights at Proc. 26th Int. Joint Conf. AI (IJCAI, pages 5146–5150*.

[24] Tang, P. 2017b. Reinforcement mechanism design. In *Proc. 26th Int. Joint Conf. AI, IJCAI-17*, 5146–5150.

[25] Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* 16(1):8–37.