Reinforcement Learning Methodology for Electricity Market Simulation

Charles Renshaw-Whitman

Delft University of Technology

TUDelft

Reinforcement Learning Methodology for Electricity Market Simulation

by

Charles Renshaw-Whitman

Student Name	Student Number
Charles Renshaw-Whitman	5513812

Supervisor:	Prof L. de Vries
Supervisor:	Prof J. Cremer
Supervisor:	V. Zobernig

Cover: Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Contents

1

Introduction and Literature Review

1.1. Introduction: Market Design and Reinforcement Learning for the Energy Transition

The design of electricity markets is both increasingly important and increasingly complex as the world undertakes the transition to a low-emissions power system. Increasing penetration of intermittent renewables ensures that secondary markets in, i.a., inertia, balancing, and capacity, play an ever more important role in generators' portfolios. From a societal perspective, more complex markets impose a greater difficulty on market designers wishing to ensure the system's soundness against exploitative bidding strategies - and thus entail a risk that generators will abuse their market power at consumers' expense.

Complex physical constraints and multiple coupled markets make difficult the application of traditional optimization methods for determining the Nash equilibrium bidding strategies [36]. Recent work has applied techniques from single-agent [35] and multi-agent [12][23] reinforcement learning (RL) to address this increase in complexity.

Despite this increased interest in RL techniques, the application to market simulation (and in turn to market design) imposes new constraints on the algorithms used to simulate bidder behavior. In this work, we seek to address the problem of non-stationarity - the effective change in environment faced by an agent due to other agents' learning new strategies - for multi-agent reinforcement learning (MARL) algorithms in the context of electricity markets. This done, we go on to show how MARL methods permit the adversarial design of electricity markets, selecting an approximately optimal market-design from a pre-specified family of designs.

In the first part of the work, we illustrate the problem of non-stationarity in a case of tabular RL (i.e. with discrete, finite action- and state-spaces). This done, we focus our attention on the continuous case, which we accommodate via the use of "Deep Reinforcement Learning" (DRL). We examine the non-stationarity problem on an algorithm analogous to that used in the tabular case, the "Deep Deterministic Policy Gradient" (DDPG). After showing this algorithm can learn to behave strategically in simple environments, we show that it too suffers from the problem of non-stationarity when one tries to train multiple agents simultaneously. Subsequently, we make use of the specially adapted "Multi-Agent Deep Deterministic Policy Gradient" (MADDPG) algorithm; this algorithm uses a unique centralized critic for each agent which accounts for the effect of all agents' behaviors, while the trained agents have access only to their own state-observations - they can thus react intelligently in an environment with other agents at deploy-time without needing to model these other agents. This algorithm has been used with success in, e.g., [18], and in electricity markets specifically in [7]. We demonstrate that for the simple day-ahead market environment created for this work, MADDPG outperforms its decentralized twin, DDPG and addresses the non-stationarity problem; subsequently, we prove conditions under which centralized-decentralized value-learning converges. Finally, we make use of this knowledge to illustrate how DRL can be used not only to evaluate markets, but also to design them, examining the case where power-producing-agents behave strategically while a market-manager simultaneously learns to set a price-cap in order to maximize social-welfare.

This work ultimately seeks to contribute to the field of electricity market design on two fronts. First, we offer a methodological contribution, indicating a principled way to select MARL algorithms in order to simulate economic agents' strategic behavior, and explicate the conditions under which convergence can be expected. Second, we introduce methods to design markets, offering a perspective combining ideas from mechanism design with those of DRL, in the hopes that this new tool will aid computational market designers in more effectively ensuring socially beneficial market outcomes.

1.1.1. Plan of the Work

The plan of the work proceeds as follows: in this Chapter 1 we have offered an introduction to the problems with which we are concerned, which will be followed subsequently by a literature review highlighting different fields from which the present work draws. The succeeding subsections will offer an overview of game theory and reinforcement learning sufficient to acquaint the reader with the appropriate terminology and the notation used in this work.

Chapter 2 begins by offering an illustration of a very simple case of non-stationarity and discussing its origin in MARL. We focus first on the tabular case because this is the domain in which algorithm implementation is simplest. Afterwards, we describe a simple day-ahead electricity market, discretized so that simple tabular RL methods may be used; we introduce the ”optimality-deficit” as a measure of how far an agent's policy is from a best-response (due to non-stationarity). We demonstrate the occurrence of such failures, and also offer a brief discussion of some of the incidental parameters governing the RL algorithm. Finally, the results of the experiments are briefly summarized and discussed.

Chapter 3 extends the work of the previous chapter - where before tabular reinforcement learning was used, we now introduce DDPG and MADDPG for the operation of a continuous environment. After a brief introduction to these algorithms and exposition of some of the incidental parameters governing them, we extend the ”optimality deficit” from the previous chapter to the case of Bertrand competition, along with a suggestion for how it might be further generalized. Subsequently, we proceed to examine the performance of these algorithms in a continuous analogue of the day-ahead market from the previous chapter, using comparison with the Bertrand solution to determine how effectively a learning agent is playing. Subsequently, we demonstrate that, in cases where multiple DDPG learners fail to play an equilibrium, MADDPG does so with greater success. After a discussion of the results and their implications, the chapter closes with a proof of the conditions under which centralized-decentralized algorithms (to which class MADDPG belongs) can be expected to converge, simultaneously illustrating why we should not expect independent learning algorithms like DDPG to reliably find equilibria.

Chapter 4 introduces the idea of adversarial market-design. A brief discussion of mechanism-design, its methods, and its goals is offered in order to illustrate the ways in which this adversarial market-design differs. We then go on to offer a schematic illustration of how this technique could be applied to electricity market design; to the simple day-ahead market from the previous sections, we add an agent which learns to set a price-cap according to some social-welfare function. We present a mathematical formulation of the predicted market-equilibrium as a function of the price cap; subsequently, we compare these predictions to numerical experiments. We discuss the results of the experiments and offer some general remarks on the prospects of adversarial market-design.

Chapter 5 begins with a recapitulation of the results of the prior chapters. After discussing the relationship between the results from different chapters, we conclude with remarks on potential directions for future work.

As this work focuses in part on RL methodology, there will be occasion to employ some mathematics which may at first blush appear somewhat intimidating; a typical example is an expectation with respect to some distributions which require multiple symbols to specify. While this is necessary for precision, we also wish to illustrate the fundamental economics at issue as much as possible - accordingly we have made every effort to ensure that a reader uninterested in disentangling mathematical formulations can comfortably follow the work at the conceptual level.

1.2. Literature Review

Here we present a brief summary of related research which informs the perspective of the present work. We first discuss the problems faced by electricity-market designers with a focus on research in coupled markets and trends related to the energy transition. Subsequently, we offer a broad overview of some notable applications of MARL methods in solving game-theoretic problems, and contrast these with other methods of computational game-theory. Finally, we discuss some recent applications of RL (single- and multi-agent) in the field of electricity market-design.

1.2.1. Energy Transition and Electricity Markets

The classic pedagogic text on electricty markets is [16]. [6] offers an overview of the field, discussing the key issues and the criteria by which success is to be judged - namely, the ability to ”[provide] reliable electricity at least cost to consumers”. In addition to these traditional economic aspects, market designers must increasingly design for an energy system making heavy use of renewables; an excellent discussion of the economic aspects of intermittent renewable energy sources is offered by [14]. Many of the key insights and difficulties of electricity market-design relate to the interaction between the short-term power- and ancillary- markets and the longer-term impact on generation capacity; an explication of the relevant considerations is offered in [1] in the context of forward markets. It is noteworthy that much of the work in this area is reasoned about conceptually, as opposed to mathematically. This seems, in part, a result of the fact that, while phenomena such as e.g., risk aversion, political uncertainty with respect to permitting, effects of short-term markets on long-term investment etc. can be handled mathematically, this may be done only at the cost of introducing complex models which themselves entail many assumptions. Ideally, mathematical frameworks like that of mechanism design would complement and synergize with the intuitions of economists and policymakers.

1.2.2. MARL Methods in Game Theory and Alternative Approaches

One of the core perspectives from which this work draws is that of game theory. In particular, one understanding of MARL methods is as a general solution technique for games. One class of algorithms with which MARL methods bear comparison is direct-optimization models, which seek to explicitly formulate and solve the mathematical conditions governing optimal agent-behavior; an introduction to such equilibrium solution methods in electricity-market contexts is offered in [11]. Broadly speaking, RL methods may be thought of as very general approximate optimization algorithms - typically, bespoke optimization algorithms perform better than RL in within a limited domain, and fail entirely when certain assumptions are not met (e.g. convexity of a reward function); by contrast, RL algorithms suffer from a variety of reliability issues, and rarely have practical guarantees of achieving the true optimal solution, but can be readily employed in complex domains where ”good-enough” solutions can be highly valuable.

Game-theory provides an effective framework for discussing problems in MARL - when agents do not share a common reward function, the language of games offers a useful way to understand the incentives facing an agent. A general discussion of the relationship between MARL and game-theory can be found in [17] - the joint-policy-correlation-matrices introduced by the authors’ offers a different perspective on very similar issues to those which we discuss here, focusing on the performance of independent-learners trained in different experiments. A broad overview of classical MARL methods is offered in [4], and a more specific treatment of the non-stationarity problem with which we are concerned here is offered in [21]. These works introduce many techniques which have been used to coax MARL agents into solving games to Nash-equilibria; the most principled being the use of other-agent-modelling, which inevitably leads to violating the assumption of hypothesis-realizability underlying RL (though, as discussed below, frameworks like Infra-Bayesian probability have been set out to counteract this issue).

As game- and decision-theorists’ models work by making assumptions about agents’ behavior, one of the great problems of multi-agent theory is how to model agents reacting to other agents - exotic problems in this domain include the ”Open-source prisoner’s dilemma” (in which agents are possibly identical AIs with access to each others’ source-code, and must choose to co-operate accordingly), as well as Newcomb’s problem (in which a superintelligent oracle places $1 in the first of two boxes, and $1000 in the second if and only if it predicts the agent will take only the second box, otherwise leaving the second box empty) [34]. As noted above, one approach is to model agents as having models of other agents (having models of other agents having ... *ad infinitum*); such an approach has been combined with modern DRL methods in [9]. This necessarily entails that an agent model something at least as complicated as itself - an impossibility known as ”hypothesis nonrealizability”. A fascinating, if inscrutable, alternative is that of Infra-Bayesian decision-theory [15], which explicitly understands both agents and opponents as physical subsets of the environment, and also provides an extensive mathematical framework for decision-making in non-realizable hypothesis spaces. Given the diversity of the work in this direction, we attempt in this work to deal as much as possible with situations in which agents need not directly respond to one another, in order to highlight the core aspects of the problem.

Recent high-profile applications of MARL methods to game-theoretic problems include [18], in which is introduced the MADDPG algorithm which plays an important role in the present work, as well as [22], which combines deep RL with game-theoretic methods to solve the imperfect-information game of Stratego.

Ultimately, just as economists use game theory to model economic agents as approximately rational, MARL methods in some sense generalize this to offer an account of how rational agents can learn about a multi-agent environement well enough to play effectively. This entails two possible use-cases: MARL methods can be used to simulate the learning procedure of actual actors facing an uncertain strategic environment (this is a relatively rare approach, but see [8] for a relevant discussion, albeit only with mostly implicit connections to RL). Alternatively, they may be used to generate possible (approximate) equilibrium strategies in environments which are too complex to ”solve” with the tools of game theory; it is the latter use-case with which we principally concern ourselves in this work.

1.2.3. RL In Electricity Market Design

Within the field of electricity market-design, the most common use of RL and MARL methods is for the simulation of economic agents' strategic behavior - a common use-case is to build a model of some portion of an electricity market and then allow RL agents to operate one or more of the market-participants, with the aim of determining how vulnerable the market is to strategic behavior.

An overview of the state of RL methods in electricity markets is presented in [37]; the authors discuss a number of single- and multi-agent algorithms for doing RL-based analysis of electricity markets.

More directly antecedent to the present work, [36] compare the performance of bi-level optimization algorithms with a deep RL DDPG agent. A comparison of various bidding strategies, DDPG, and MADDPG is performed in [7], which finds MADDPG to well-approximate the exact solution of a given MPEC, while generally learning more efficiently than DDPG. [23] presents a case study utilizing MARL methods to identify cases in which agents are able to exercise market power thus informing the design of a balancing-market - notably, in this case, the authors augment the state-space of learning agents with past market-clearing data - which data *per se*, by the Markov assumption, *cannot* be decision-relevant for these agents; instead, this augmentation artificially enlarges the state-space, implicitly permitting adaptation to other agents' changing policies (c.f. the theorems proved at the end of Chapter 4 of the present work). While these authors verify their algorithm reproduces the Bertrand solution, an explanation for how or when such artificial augmentations work would be valuable in designing RL setups for similar applications. Finally, [12] provides an analysis of the performance of the DDPG algorithm in a market in which the analytical-solution is known, highlighting, i.a., the issue of non-stationarity which we discuss in the present work.

1.3. Game Theory and Nash Equilibria

One of the most fruitful methods of economic analysis is the theory of games. Game theory describes the strategies of rationally-acting agents seeking to maximize a ”utility-function”, potentially in conflict with other agents seeking to do the same. The application of game theory to economics will be central to this thesis, as rational agents are also the natural comparison point for any trained agents. In this section, we introduce some of the core principles and some of our notation.

For the purposes of this thesis, we discuss the action of energy-producers within an energy-market as rational actors seeking to maximize their profit; here, the modeling assumption of rationality is justified by an appeal to coherence theorems and money-pump arguments, such as [29], to the effect that a failure of rationality (to oversimplify, a failure to act as if an agent was a Bayesian expected-utility maximizer) would amount to behavior which could be exploited to take arbitrary amounts of utility from the agent. This work will not discuss the ways in which this model fails to accurately model reality, even as these are of prime importance in energy-market design, primarily because their mathematical modelling is a complicated and delicate matter which would take us too far afield.

We model a (single-stage, deterministic, simultaneous, perfect-information) game G as a set of agents $A^1 ... A^N$, to each of which is available a set of actions $\mathcal{A}^i$; we denote the Cartesian product of all actions $\mathcal{A} = \bigotimes_{i=1}^{N} \mathcal{A}^i$. Each agent has a utility function $U^i : \mathcal{A} \to \mathbb{R}$ - a function which, for any joint action, assigns the agent's valuation of such an outcome.

A joint action $a^* \in \mathcal{A}$ is said to be an equilibrium if no agent can increase its expected utility by a unilateral deviation:

$$\forall i \in \{1, ..., N\}, \forall a^i \in \mathcal{A}^i,$$
$$U^i(a^{-i*}, a^i) \leq U^i(a^{-i*}, a^{i*})$$

(where the "$-i$" is a compact way of representing all actions other than that of agent i).

In general, not all games have such an equilibrium if we require an agent to choose a specific action in advance (a 'pure strategy') - they may instead choose a probability distribution over possible actions, known as a 'mixed-strategy'. In cases where agents employ a mixed strategy, the equilibrium is known as a "Nash equilibrium". It is a well-known theorem [26] that all games of the type we consider here have at least one mixed-strategy Nash equilibrium.

A mixed strategy for an agent i is a probability distribution over possible actions $\sigma^i \in \Delta\mathcal{A}^i$ (ΔS denoting the set of probability distributions over elements of a set S). The criterion for a set of mixed strategies σ^* (note that throughout this work, unindexed quantities refer to the cartesian product of the relevant quantity over all agents, unless otherwise noted) to be a Nash equilibrium is, similar to the above,

$$\forall i \in \{1, ..., N\}, \forall \sigma^i \in \Delta\mathcal{A}^i,$$
$$\mathbb{E}[U^i(a^{-i*}, a^i)|a^{-i*} \sim \sigma^{-i*}, a^i \sim \sigma^i] \leq \mathbb{E}[U^i(a^{-i*}, a^{i*})|a^{-i*} \sim \sigma^{-i*}, a^{i*} \sim \sigma^{i*}]$$

($\mathbb{E}$ denotes the expectation value, while the tilde may be read as "distributed according to") This is the natural extension of the concept of a Nash Equilibrium as above to the case of probabilistic policies.

Finally, in some games, agents have access to hidden information which may inform their actions. In this case, it is necessary to introduce the concept of a Bayes-Nash equilibrium. In this generalization, each agent i is in possession of some hidden information $\theta^i \in \Theta^i$, distributed - we will assume independently - according to a publicly-known distribution $p^i(\theta^i)$; the agents' utility functions U^i and strategies σ^i are then permitted to be functions of this hidden information as well. In this case, a Bayes-Nash equilibrium is defined as a family of strategies $\sigma^i : \Theta^i \to \Delta\mathcal{A}^i$ [26]

$\forall i \in \{1, ..., N\}, \forall \sigma^i : \Theta^i \to \Delta\mathcal{A}^i,$
$\mathbb{E}[U^i(a^{-i*}, a^i|\theta^i)|a^{-i*} \sim \sigma^{-i*}(\theta^{-i*}), \theta^{-i} \sim p^{-i}, a^i \sim \sigma^i(\theta^i), \theta^i \sim p^i]$
$\leq \mathbb{E}[U^i(a^{-i*}, a^{i*}|\theta^i)|a^{-i*} \sim \sigma^{-i*}(\theta^{-i}), \theta^{-i} \sim p^{-i}, a^{i*} \sim \sigma^{i*}(\theta^i), \theta^i \sim p^i]$

These are the criteria by which we shall evaluate our reinforcement learning agents - these equilibria are the strategies to which a MARL training process should converge.

1.4. Reinforcement Learning

Reinforcement Learning (RL) is a paradigm for machine learning; in contrast to the other major divisions of machine learning, supervised or unsupervised learning, reinforcement learning focuses not on statistical prediction based on data, but instead on the maximization of a 'reward function'. The mathematical framework for understanding RL is that of Markov Decision Processes (MDPs); the most basic type of MDP is defined by four main objects: $\mathcal{S}$, the set of all possible states of the environment, $\mathcal{A}$ the set of actions available, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the reward function (the quantity to be optimized) and $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$, the transition function, which details the probability of arriving at a particular new state given the prior state and an action taken. To these is often appended a time-limit T (such that at most T actions may be taken consecutively) and a discount-factor $0 \leq \gamma \leq 1$, representing the rate at which rewards obtained at later times are discounted compared to those at earlier times (i.e., the reward at time t is discounted by a multiplicative factor γ^t). We refer to ordered lists of state-action pairs $(s_0, a_0, s_1, a_1, \ldots s_T, a_T)$ as 'trajectories' (often denoted by the letter τ). Sometimes agents do not know the true state of the environment s_t, but instead must infer it based on observations o_t which contain incomplete information about the full state; unless otherwise noted, we will generally elide states and observations of states, as the distinction is usually clear from context.

The core of the RL problem is to find a (possibly probabilistic) 'policy' $\pi : \mathcal{S} \to \Delta\mathcal{A}$ which maximizes the expected cumulative reward $R[\pi]$,

$$R[\pi] = \mathbb{E}\left[\sum_{t=0}^{t=T} \gamma^t r(S_t, A_t)|A_t \sim \pi(S_t), S_{t+1} \sim \mathcal{T}(S_{t+1}|S_t, A_t), t \geq 0\right]$$

(for simplicity, we consider the initial state $S_0 = s_0$ to be fixed). Following (the first edition of) [31], we use capital letters to indicate random variables, and lower-case letters to indicate a particular instantiation of the corresponding random variable.

As this notation is quite cumbersome, it is customary to suppress the distributions from which the random variables are drawn, instead indicating that trajectories τ are drawn according to the policy π; for a quantity X, $\mathbb{E}[X(\tau)|\tau \sim \pi]$ or $\mathbb{E}_\pi[X]$. In order to briefly distinguish it from the reward at a particular timestep, $R[\pi]$ is typically called the 'return'. The policy should choose actions which, in expectation, yield trajectories along which the most possible reward is gathered. When the policy is chosen from a family of policies dictated by some parameters θ, the corresponding policy is denoted π_θ, in which case the return may be considered a function $R[\theta]$.

Some algorithms, such as the eponymous REINFORCE [30], directly optimize this policy by estimating $\nabla_\theta R[\theta]$ and using this quantity to perform gradient descent. Such techniques are known as 'policy learning'. Alternatively, many algorithms seek to learn a value function which approximates the return, and to subsequently optimize the policy with respect to this - which algorithms are called 'value-learning'. There are two common value functions, the 'state-value function' $V_t^\pi(s)$ and the 'state-action-value function' $Q_t^\pi(s,a)$. The definitions of these are as follows:

$$V_{t_0}^\pi(s) = \mathbb{E}\left[\sum_{t=t_0}^{t=T} \gamma^t r(S_t, A_t)|\tau \sim \pi, S_t = s\right]$$
$$Q_{t_0}^\pi(s,a) = \mathbb{E}\left[\sum_{t=t_0}^{t=T} \gamma^t r(S_t, A_t)|\tau \sim \pi, S_t = s, A_t = a\right]$$

Thus $Q_t^\pi(s,a)$ may be interpreted as "the expected return of taking action a while in state s at a time t, and subsequently operating according to the policy π thereafter", and analogously for V_t^π.

The functions given above are typically not found by directly computation (not least because this requires the evaluation of $|\mathcal{S}|^{|\mathcal{A}|^t}$ trajectories), but are instead estimated on the basis of their values over a small family of sampled trajectories. The selection of these trajectories requires balancing the desire to consider only a limited set of 'promising' trajectories ('exploitation' according to the possibly inaccurate value function) with the need to calibrate the value function over a wide family of trajectories ('exploration'). This is commonly referred to as the 'explore-exploit' tradeoff. In general, this means that, in sampling trajectories, there is cause to, some fraction of the time, take an action distinct from that which the policy 'guesses' is optimal. The particular techniques used in this work will be discussed in the corresponding sections. [31] discusses a wide variety of exploration techniques, with more created regularly.

What relevance has Reinforcement Learning to economics as a whole, let alone to the design of electricity markets? As noted previously, economics is, at least in part, the theory of so-called 'rational actors', whose rationality is defined, with reference to, e.g., the von-Neumann-Morgenstern axioms [29], as the maximization of some expected-utility function. While this model is generally regarded only as an approximation to the behavior of individuals with potentially conflicting preferences or other irrationalities, the relation to firm-behavior is, if imperfect, much more direct - the nominal purpose of a for-profit firm is the maximization of profit.

Regarding market-design more specifically, there are two principle reasons to speak of RL methods. The first is practical: if market-design is to be regarded as some sort of game between agents, this game must at some point be solved; while traditional optimization solvers (e.g., EPEC models [11]) provide exact answers for a limited class of models, as the complexity of the game increases, possibly violating the assumptions of solvers based on, say, convexity, such direct solution becomes intractable. On the other hand RL algorithms are readily adapted to any environment in which there is a well-defined reward function. While RL will not, in general, determine the optimal solution in finite time, it can often readily arrive at reasonable strategies (and thus, hopefully, approximate the actual behavior of firms with sufficient fidelity to draw conclusions about the suitability of a particular market rule).

The second reason is theoretical: firms in reality exhibit behavior which would be irrational were their objective solely the maximization of expected profit; instead, considerations such as risk-aversion [13], and managerial short-termism can be considered rational only with respect to utility functions more complex than the strict expected-profit; RL methods extend to such models merely by changing the reward-function appropriately. Further, in games with multiple equilibrium solutions, actors must eventually co-ordinate to arrive at one particular solution, a process known in economic theory as ”tâtonnement” [28] - modeling agents as reinforcement-learners provides an account of how, subject to appropriate assumptions, rational agents should learn.

The extent to which learning provides a realistic model of behavior in out-of-equilibrium games is discussed in [8] (though with respect to individuals rather than firms). The celebrated Sonneschein-Mantel-Debreu theorem [28] demonstrates that markets composed of rational actors need not, in general, converge to a unique and stable equilibrium via the tâtonnement; the non-stationarity problem is in many regards analogous to the failure of tâtonnement, except for occurring between learning agents taking multiple actions over time. In this sense, our work with respect to non-stationarity will be an attempt to eliminate all these tâtonnement-failure-analogues which are due solely to the learning process and not the underlying game structure.

2 Theory and Tabular Reinforcement Case

2.1. Non-Stationarity and the Failure of Independent Learning

Traditional RL assumes that the environment within which an agent operates is Markovian - that is, that state-transition and reward probabilities, $\mathbb{P}(S_{t+1}, R_t | S_t, A_t)$ are independent of time [20]. Independent learning (IL) refers to any MARL scheme in which agents are trained using single-agent RL techniques, taking the behavior of other agents to be implicit in the environment. That is, though the state-transitions or rewards may depend on all agents' actions, an IL agent will learn as though its action alone determined these probabilities. In the case where only one agent is learning (others remaining fixed), the Markovian assumption holds, as the other agents' actions may be understood as part of a stochastic environment. On the other hand, when multiple agents are learning simultaneously, the Markovian assumption is, in general, violated. This in turn means that standard guarantees of convergence no longer apply.

Recent work in this vein has focused on applications to cooperative RL, in which agents learn separately but share an overall reward function. In this case, techniques such as decomposition networks [24], coordination graphs [2], and optimality-gap formulations [27] have been used.

By contrast, when agents do not share a reward-function (i.e., in mixed environments), the problem ceases to be one of credit-assignment; in such cases, algorithms such as fictitious play [10], along with techniques like leagues [32] have been used. The algorithm we focus on here, MADDPG, has been used successfully in general MARL environments [18] and in electricity market behavior in particular [7]. At the end of Chapter 3, we discuss general criteria for value-learning convergence.

The problem of non-stationarity in MARL ultimately entails a failure of agents to converge to a Nash-equilibrium, in which case they cannot be used to meaningfully infer the properties of electricity markets with respect to strategic behavior.

2.2. Convergence Criteria and Definition of the Optimality Deficit

We wish to study when RL algorithms converge to an equilibrium solution to the economic dispatch problem; thus we must define some notion which measures the distance of an agent's policy from equilibrium. This section defines such a notion which will be used throughout our experiments, which we call the "optimality deficit".

For simplicity, we will begin by discussing the illustrative, simplified case in which the environment itself has no state, and all actions are selected from a discrete set of actions (the same for all agents). We will restrict ourselves also to the case where there are only two bidding agents.

Let us call the policies played by each agent π_1 and π^2, respectively (here, interpreted as vectors describing probability distributions over the discrete action-space). Then the expected return of an each agent is given by the simple matrix-vector multiplication

$$\pi^{1,T} U^i \pi^2$$

With U^i the utility matrix of player i (whose elements, U^i_{mn} are the profit obtained by agent i when agent i performs action m and their counterpart action n.

The Nash equilibrium condition asserts that each player is playing a best-response to the other. For agent 1, this takes the form

$$\pi^1 \in \text{argmax}_{\pi^1} \pi^{1,T}(U^1\pi^2)$$

(The brackets emphasize that, fixing the counterparty's policy entails an effective single-player utility). The Nash-equilibrium condition corresponds to requiring that this condition attains simultaneously for all bidders.

While it is not necessarily easy to calculate Nash equilibria, it is simple enough to verify whether a given set of policies constitutes an equilibrium (at least in this discrete case). A best-response is one in which the reward is the equal to the reward obtained by playing the deterministic policy which always plays the maximum value of (the vector) $U^i\pi^j$ (this strategy may not itself be the equilibrium, as it would not necessarily satisfy the criterion for the remaining agent).

Thus, we can quantify how far a given policy is from a Nash-equilibrium based on it's loss of potential profits:

$$\lambda^i = \pi^{i,T}U^i\pi^j - ||U^i\pi^j||_\infty$$

Here $||\cdot||_\infty$ denotes the infinity-norm of a vector (equal to the value of the vector's maximum entry). Intuitively, this deficit is the profit which agent i fails to obtain by fact of playing suboptimally, assuming the behavior of the opponents are fixed. This λ^i we call agent i's "optimality deficit".

Having developed a criterion by which to gauge whether a true equilibrium has been reached, we now wish to define a measure that describes when a training procedure has converged (not necessarily to a correct equilibrium). A commonly used measure, the TD-error, felicitously fulfills this function. The TD-error is equal to the difference between the agent's expected reward and its actual reward. When these two are approximately equal for every state-action pair, the value-function has converged; in the following section, we consider stateless environments, where the policy is the argmax of the value function. Under our simplifying assumptions the formula for the TD error δ_t at time t is:

$$\delta_t = Q(a_t) - r_t$$

where a_t is the action taken at time t, r_t the reward received, and $Q(a_t)$ the agent's predicted reward. Note that this definition applies only to a single, independent learner - it does not admit introducing the effects of other agents' actions. Intuitively, when the TD error has gone to zero for all agents over all actions (or at least over all actions which are played), the agent is no longer 'surprised', and will continue to act as it thinks best according to its Q-function.

2.2.1. Example

Following [33], consider the co-operative simple matrix game described by the utilities:

$$U = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 6 \\ 0 & 6 & 7 \end{bmatrix} \tag{2.1}$$

(element U_{ij} corresponds to agent 1 taking action i while agent 2 plays action j). Now, if both agents begin by exploring randomly and independently, each outcome will occur with probability $1/9$. E.g., agent 1 learns that action 3 has value $Q^1(3) = (0+6+7)/9 = 13/9$. Thus, each agent's learned (after training to convergence) Q-values will be $10/9$, $11/9$ and $13/9$ for actions 1, 2, and 3, respectively. Thus supposing the agents implement a greedy policy after training, they will both choose action 3, receiving 7 reward rather than the optimal 10.

This example illustrates the phenomenon of relative overgeneralization for co-operative games; in such cases where there is a shared reward function, there exist a number of techniques which facilitate co-operative learning, as discussed above. These techniques, however, do not readily apply in the case of mixed-sum games such as the economic dispatch problem.

2.2.2. More General Formulation

For reference, we formulate the above-defined optimality-deficit without the simplifying assumptions of statelessness, observability, discrete action space, or having only two agents. Here we let τ^i_t denote the observation-action history of agent i at time t, as opposed to a full trajectory.

The Bellmann-equation defining the optimal state-value function is

$$V^{*,i}(\tau_t^i) = \max_a \mathbb{E}\left[R^i(S_t, A_t^{-i}, a) + \gamma V^{*,i}(\tau_{t+1}^i)\right]$$

As noted before, all capital letters are understood to be random variables over which expectations are taken (albeit with τ being inferred from context in the absence of an easily identifiable capital). As much as possible, we will suppress distributions which are obvious from context. Likewise, the observations of other agents are suppressed for notational simplicity, but must also be integrated over.

The expected profit of an agent obeying strategy π^i is similarly

$$V^{\pi^i,i}(\tau_t^i) = \mathbb{E}\left[R^i(S_t, A_t^{-i}, A_t^i) + \gamma V^{\pi^i,i}(\tau_{t+1}^i)|\pi^i\right]$$

Different generalizations of the optimality-deficit can be obtained by choosing whether the agent subsequently obeys some original policy or its optimal policy. As we do not, in general, have access to the optimal policy or value function, the version for which we will have the most use is the deficit with respect to an agent subsequently playing its original policy; this corresponds to the expected profit suboptimality in a single timestep.

$$\lambda^i(\tau_t^i) = Q^{\pi^i,i}(\tau_t^i, a^{i,*}) - V^{\pi^i,i}(\tau_t^i)$$

In the upcoming section, we will have occasion to refer to the case in which agents bid for a single clearing of a market, and in which each has hidden information about its generation capacity and marginal costs; for this case, the idea is the same but for taking the expectation over the counterparties' policies:

$$\lambda^i(o_t^i) = \max_a \left\{\mathbb{E}\left[R^i(S_t, A^{-i}, a)|A^{-i} \sim \pi^{-i}(O^{-i})\right]\right\} - \mathbb{E}\left[R^i(S_t, A^{-i}, A^i)|A^{-i} \sim \pi^{-i}(O^{-i}), A^i \sim \pi^i(o^i)\right]$$

Intuitively, the deficit represents the expected loss in profit - compared to what it would achieve playing optimally - of an agent with generation parameters τ_t^i, supposing that its opponents bid according to fixed policies, with their own hidden generation parameters drawn from some known distribution.

One important feature of the discrete environment is the ability to directly compute the optimality-deficit: fixing the agent whose optimality-deficit we wish to calculate, the remaining agents may be assumed (once exploration has been disabled) to play a greedy policy which always selects their guessed-optimal action - thus, the given agent's optimal action may be obtained by brute-force search, requiring only $|\mathcal{A}^i|$ (the number of actions available to the agent under consideration) market clearings.

2.3. Experiment Setup

In light of the theory presented above, we wish to investigate the circumstances under which independent learners will converge to an incorrect bidding strategy. To this end, we simulate a simple day-ahead market, described in detail below, with agents trained implementing simple tabular Q-learning. The optimality-deficit is calculated, and examples are shown in which convergence occurs while agents maintain a non-zero optimality deficit.

2.3.1. Description of the Environment

For the sake of clarity, we model a simple day-ahead electricity market. Consumer demand is taken to be linear as a function of the electricity (in MWh) demanded. A single market-clearing is simulated - technical operation constraints are not considered. Agent observations are given by four parameters, each selected from a discrete list of possible candidate values: two parameters describe the demand function (i.e. a slope and an intercept), which are visible to all agents. The remaining two parameters describe an agent's maximum generation capacity (in MWh), and the marginal operating cost of their generator (in USD / MWh) - these generation parameters are visible only to the generator's owner (e.g. agent 1 is not permitted to see agent 2's marginal operating cost). Of course, this is not especially realistic given that firms can reasonably infer such information about their opponents - the point is to provide the grounds to illustrate the optimality-deficit in a game with hidden information.

In each round, agents select a pair of actions (a, b) corresponding to the fraction of their total generation capacity to bid, and the fraction of their marginal cost to bid - i.e., the action selection (a, b)

corresponds to a bid to sell aQ^i_{max} units of electricity at a price bP^i_{MC}. Allowed values of a and b are the same for all agents at all times, and are each drawn from a distinct discrete set. This set is a linear spacing between a minimum and maximum value with a chosen number of sample points. Unless otherwise noted, Q_{max} fractions were permitted to go from 0.5 to 1.0, while P_{MC} fractions were permitted to range from 0.5 to 10; there 3 choices for quantity bidding, and 10 for price-bidding, resulting in 30 total pairs.

The market clears in as-bid merit-order, with a uniform clearing price set to be the minimum price at which sufficient bids are accepted to satisfy demand at that same price level. Ties are resolved in no particular order, and the acceptance of a fraction of a bid's quantity is permitted.

2.3.2. Training Procedure

Agents were initialized with an empty Q-table. After every round, an agent learns from an observation. This learning takes one of two forms; one in which all reward-observations are stored, and their average value used as the estimated Q, and one in which Q was updated such that $Q_{new} = \rho Q_{old} + (1-\rho)r$ (with ρ a "learning-rate" parameter). We call the former the "average reward" scheme, and the latter the "soft-update" scheme.

Different types of exploration were examined - ϵ-exploration, in which agents select the action with the highest Q value with probability $1-\epsilon$, and act randomly with probability ϵ. Thus, the exploration method discussed in the example in the previous section is ϵ-exploration, with $\epsilon = 1$. Alternatively, Boltzmann exploration was also used in some experiments; in this method an action is selected as the softmax probability distribution over all Q-values [5]. Boltzmann exploration employs a parameter β (the "inverse temperature") governing the amount of exploration - $\beta \to \infty$ corresponds to choosing the guessed-optimal action with probability 1, while $\beta = 0$ corresponds to acting uniformly randomly.

For the sake of preventing a profusion of graphs and test-cases, in the experiments below, we employ the soft-update rule with $\rho = 0.99$, and epsilon exploration with $\epsilon = 1$ In the plots shown, exploration was, unless otherwise noted, employed for half of all time-steps. Unless otherwise noted, agents continued to learn (i.e. to update their Q-tables) after exploration was disabled.

2.4. Results

2.4.1. Single Agent

We now present a series of test cases examining the convergence properties of the TD-error and the optimality-deficit. First, in order to test the functionality of the environment, a simple case with no state-information and a single agent is performed. The results are shown in 2.1. As expected, the single agent converges to the monopoly behaviour (or at least as close as it can, given the restrictions of discreteness). Second, we add states to the single-agent case; in this case it is told at each timestep that it has a generator of one of three sizes, and must play accordingly; results shown in figure 2.2. This case converges as expected.

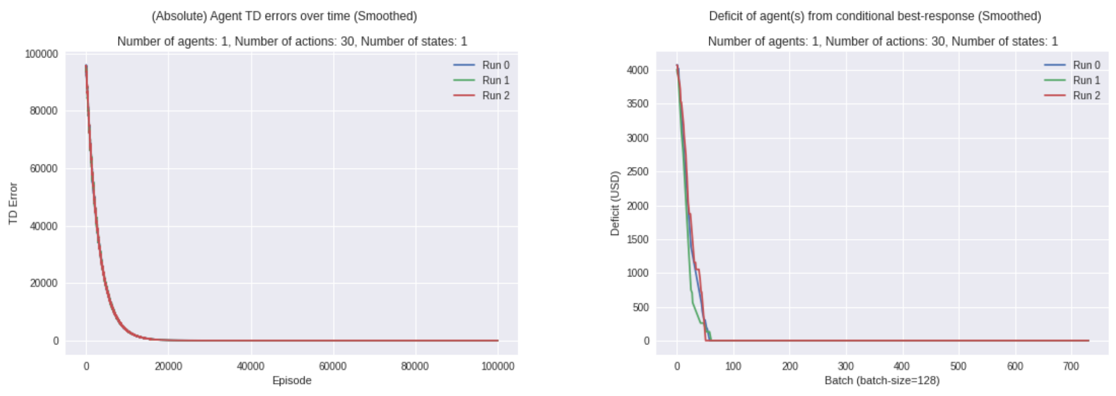


Figure 2.1: Plot of the (smoothed) agent TD-error (left) and conditional deficit (right) for the case with one agent and no state. In the testing stage, the conditional deficit is zero as expected - the equilibrium played is a genuine one.

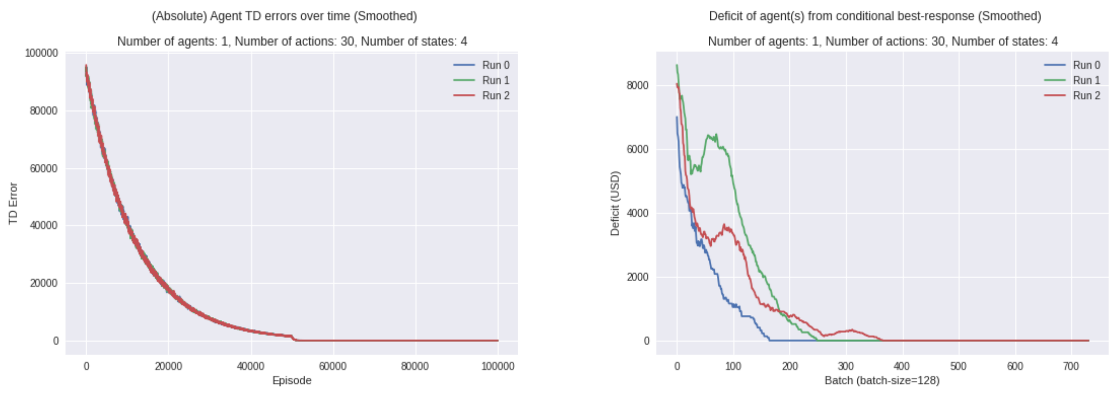


Figure 2.2: Plot of the (smoothed) agent TD-error (left) and conditional deficit (right) for the case with one agent - this time with 4 states (corresponding to different generator sizes and marginal costs); as there is only one agent, there is no hidden information. In the testing stage, the conditional deficit is zero as expected - the equilibrium played is a genuine one.

2.4.2. Three Agents

Next we examine cases with multiple (specifically, four) agents; firstly, a case with no state (i.e., Q_{max} and P_{MC} are the same at all times for all agents), and secondly in a case with four states (Q_{max} and MC each being allowed to take two possible values, observable only to the owning-agent). In both the case with no state and with four states (per agent), we observe convergence of the TD error, while the deficit does not go to zero! Thus, the agents converge to bidding outside a Nash equilibrium, as at least one bidder is failing to play a best response to the other bidders. Results are shown in figure 2.3 (no state), and figure 2.4 (with state), respectively.

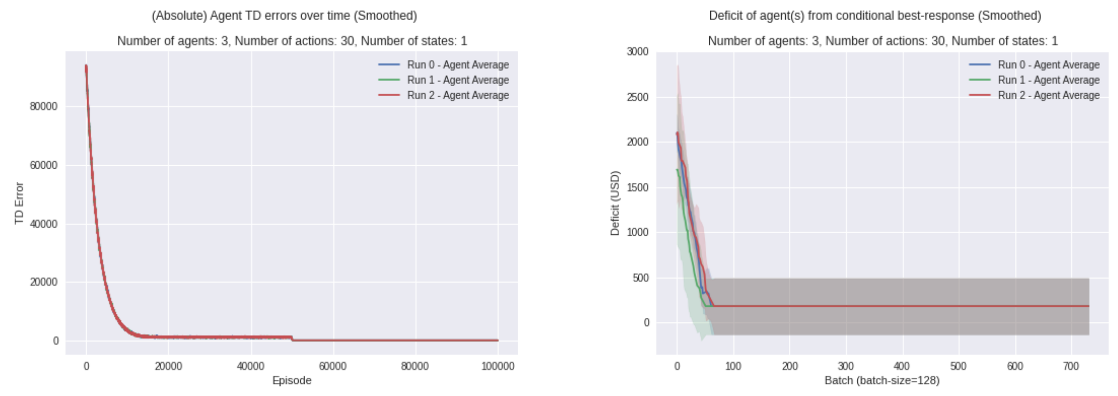


Figure 2.3: Plot of the (smoothed) agent TD-error (left) and conditional deficit (right) for the case with three agents and no state. The highlighted region denotes standard-deviation over agents. The conditional deficit is non-zero (albeit slightly) as expected - the equilibrium played is incorrect.

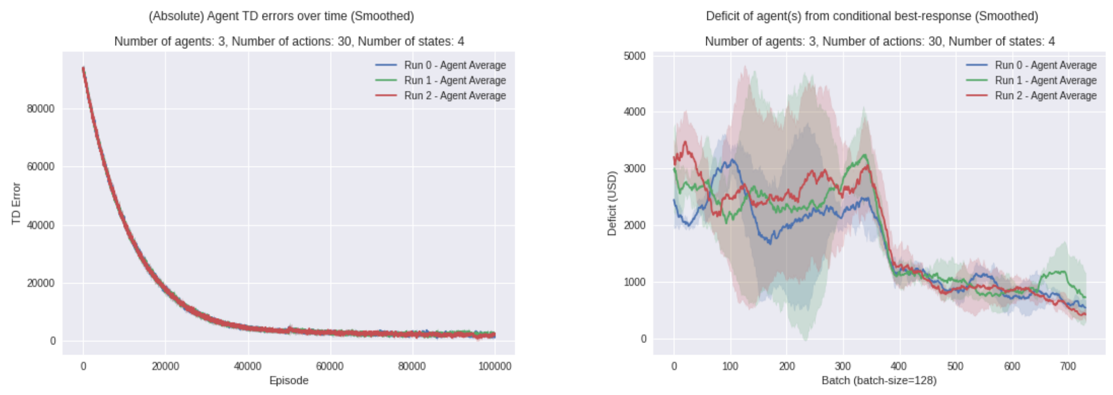


Figure 2.4: Plot of the (smoothed) agent TD-error (left) and conditional deficit (right) for the case with three agents and four states (per-agent, corresponding to generator size). The highlighted region denotes representative deviation. In the testing stage, the conditional deficit is non-zero as expected - the equilibrium played is incorrect.

2.4.3. Discussion

From the preceding experiments, we may make several observations. First, the interaction of multiple agents in even a simple day-ahead market such as that simulated can result in failure to arrive at Nash equilibria. Secondarily, it is clear that the amount of hidden-information increases the deficit - not only must agents suffer the accompanying performance penalty, but also find it harder to learn. Third, the use of analytical solutions greatly facilitates the evaluation of algorithms via the optimality deficit, and can thus bolster our credence that such algorithms will or will not work when applied to analytically intractable situations.

In the next chapter, we discuss this problem when the environment has been made continuous - this is of interest both because it allows for more realistic modelling, but also because the DRL techniques used to simulate learning agents readily extend to more complex environments in which market designers might derive real insight for their use.

3 Extension to the Continuous Case: DDPG and MADDPG

This chapter extends the previous chapter's results for tabular reinforcement learning to the case where the state- and action-spaces take continuous values. This extension is important for three reasons. First, MARL methods in market simulation are likely useful principally in cases where optimization-based models such as MPECs are not applicable, e.g., due to complex physical grid-constraints, participation in multiple markets, etc.; it is in these cases that performant DRL may be of use to policymakers. Second, tabular RL tends to require discretization of typically continuous values, e.g., permitting only certain multiples of Q_{max} as biddable quantities. Third, it permits the use of a centralized-training, decentralized-execution paradigm, while simultaneously highlighting some important caveats of the optimality-deficit as discussed previously.

3.1. Overview of DDPG and MADDPG

The purpose of this section is to briefly introduce the DDPG and MADDPG algorithms as approaches to Q-learning in continuous spaces. Then we discuss the difficulty of extending the optimality-deficit measure to the continuous case, and propose a method for approximating it.

3.1.1. DDPG

The DDPG algorithm is an algorithm for doing deep reinforcement learning that permits one to make use of a deterministic policy - that is, an actor's policy is associated with a function (traditionally denoted μ rather than π) $\mu : \mathcal{S} \to \mathcal{A}$, as compared to probabilistic policies which have the type signature $\pi : \mathcal{S} \to \Delta\mathcal{A}$. We focus our attention on the DDPG algorithm for two principle reasons: first, the fact that the policy is deterministic makes theoretical computations much simpler, as no distribution-sampling is required; second, it forms the basis of the popular centralized-training decentralized-execution algorithm, MADDPG, and thus provides an especially appropriatebasis for comparison. It is worth noting that deterministic policies are ”universal” in the sense that any parameterized probabilistic policy may be modeled as a deterministic policy over the distributions' parameters (e.g., a policy which deterministically chooses the mean and variance of a normal distribution); on the other hand, the standard version of DDPG can only be deployed on problems with a discrete state-space and continuous action-space.

We will describe the DDPG algorithm for the case of a single-agent, as, without the modifications introduced in MADDPG, the simplest extension to the multi-agent case is merely to instantiate an independent DDPG learner for each agent.

The agent is represented by a policy $\mu_\theta : \mathcal{S} \to \mathcal{A}$, a function parameterized by parameters represented by θ. Unlike probabilistic policies, gradient-descent on deterministic policies requires modelling a reward function (sometimes this reward model is called the ”critic”, in which case the policy is known as the ”actor”, and the algorithm is said to be one amongst a class of ”actor-critic” algorithms). The simplest target to model is the Q-function discussed previously, which we associate with parameters ϕ. Q is a function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. In this algorithm, gradient-descent on Q seeks to minimize the difference between the predicted Q for a given state-action pair and the observed reward; the policy μ is optimized to maximize this model-reward function. That is, ideally we would have:

$$Q^{\mu}_{\phi}(s_t, a_t) \to \mathbb{E}\left[R(s_t, a_t) + \Sigma_{t' \geq t+1}\gamma^{t'} R(S_{t'}, \mu(S_{t'}))\right]$$
$$\mu_\theta(s_t) \to \text{argmax}_\mu \mathbb{E}\left[R(s_t, \mu(s_t)) + \Sigma_{t' \geq t+1}\gamma^{t'} R(S_{t'}, \mu(S_{t'}))\right]$$

Accordingly, we can consider the Bellman equations, and take the approximations:

$$Q^{\mu}_{\phi}(s_t, a_t) \approx \mathbb{E}\left[R(s_t, a_t) + \gamma Q^{\mu}_{\phi}(S_{t+1}, \mu(S_{t+1}))\right]$$
$$\mu(s_t) \approx \text{argmax}_a Q^{\mu}_{\phi}(s_t, a)$$

(these are approximations because changes to one function are only gradually propagated to the other) Which in turn allows us to define the appropriate loss functions:

$$\mathcal{L}^Q[\phi] = \mathbb{E}\left[\left(R_t + \gamma Q^{\mu}_{\phi}(S_{t+1}, A_{t+1}) - Q^{\mu}_{\phi_0}(S_t, A_t)\right)^2\right]$$
$$\mathcal{L}^\mu[\theta] = \mathbb{E}\left[Q^{\mu}_{\phi}(S_t, \mu_\theta(S_t))|\right]$$

The subscript ϕ_0 indicates that the function-parameters used are those corresponding to ϕ, but gradients are not taken with respect to them (otherwise gradients flow in an acausal manner which gives rise to instabilities). The quantities shown are all differentiable and may be estimated from environmental data, thus the gradients may be estimated for gradient-descent.

3.1.2. MADDPG

MADDPG, introduced in [18], is an extension of the DDPG algorithm to the multi-agent case. The core problem addressed is that independent learning does not always converge correctly. MADDPG is different from DDPG in that the "critic" functions are permitted access to the data from the whole environment (including data which is hidden from the agent); thus the actor, which we wish to have access only to data available to it at execution time, is still able to learn to behave intelligently in response to other agents' changing policies.

The reason for giving each part of the algorithm access to different data is that we ultimately wish the actor's to behave without knowing any hidden information from their opponent - however, as we have demonstrated, it is necessary to take some account of opponents' behavior in order to achieve an equilibrium. The compromises solution is to allow the critic function Q to act as a "computational middleman" - that is, it is allowed access to data in training which it's agent should not actually know. However, the policy's gradient-step takes an expectation over this hidden information anyways - in the ideal case, the gradient-step would be the same as in DDPG. Because all agents are learning at once, however, the DDPG Q-function will fluctuate as agents change their policies - the MADDPG Q-function will assign disparate actions to different "bins" instead of lumping them together (which bins do *not* depend on agents' policies). Thus, when opponents change their policy, MADDPG must account for this in the policy loss - however it does *not* need to simultaneously re-learn the Q-function.

Mathematically, each agent is defined, as before, by an actor μ_{θ^i} and a critic Q_{ϕ^i} the only change is that the Q_{ϕ^i} now have the type signature $Q_{\phi^i} : \mathcal{S} \times \bigtimes_{j=1}^{N} \mathcal{O}^j \times \mathcal{A}^j$. The appropriately altered loss function for the Q is

$$\mathcal{L}^{Q^i}[\phi_i] = \mathbb{E}\left[\left(R^i_t + \gamma Q_{\phi^i}(S_{t+1}, O^i_{t+1}, A^i_{t+1}, O^{-i}_{t+1}, A^{-i}_{t+1}) - Q_{\phi_0}(S_t, O^i_t, A^i_t, O^{-i}_t, A^{-i}_t)\right)^2\right]$$

(It is of note that this Q-function makes use of all the agents' observations o and the full hidden state)

Of particular interest to the cases we discuss below, when the environment is stateless, the Q function is then totally independent of all agents' policies (because in the general case, Q depends only on agents' policies through the state-transition distribution). That is, for the simple day-ahead clearing market described previously and below, the Q-function has a global minimum independent of all agents' policies - training agents with respect to it, then, if it converges, must converge to a valid Nash-Equilibrium. For a formal statement of the conditions and proof, see Section 3.3.

3.1.3. Extending the Optimality-Deficit

In comparison to the tabular case, calculating the optimality-deficit in continuous spaces can be quite difficult. In particular, one must calculate, for each possible hidden-state, the expected return of an agent's optimal action. This entails, for each candidate for the optimal action, sampling over hidden-states and opponents' possibly stochastic policies - a scaling that quickly becomes intractable. In this section, we describe one approach to approximately compute the optimality-deficit for DDPG using an MADDPG-agent trained in parallel.

Our goal is to know, for a given agent, and fixing the other agents' policies, the expected profit of acting optimally. Supposing that the given agent is trained using DDPG, and thus is associated with both a policy $\mu^i(o^i)$ and a critic function, $Q^i(o^i, a^i)$. We then introduce a "supervisor" counterpart of this agent (trained on the same rounds as the acting agents with appropriately augmented data, and used only for deficit estimation - never acting) with associated $\mu^i_S(o^i)$ and $Q^i_S(o^i, a^i, o^{-i}, a^{-i})$ (the "S" subscript denoting the supervisor). Further, as the supervisor policy is meant to determine the best-response, it be made more accurate by giving it access to other agents' actions and observations, i.e. $\mu^i_S(o^i, o^{-i}, a^{-i})$.

In the limit of "perfectly effective" training, we anticipate that the supervisor should yield the action which maximizes the expected centralized Q-value; that is, $\mu^i_S(o^i) \to \max_{a^i} \mathbb{E}\left[Q^i_M(o^i, a^i, O^{-i}, A^{-i})\right]$. This is precisely the 'optimal action' we would like to use in formulating the deficit (indeed, there is no particular need to use reinforcement-learning to find this action - other stochastic optimization methods may be used to determine the optimal action, if desired).

Then, the approximate value of the deficit is,

$$\lambda^i(o^i, \mu^i_S(o^i, o^{-i}, a^{-i}), o^{-i}, a^{-i}) \approx Q^i_S(o^i, o^{-i}, a^{-i}) - Q^i(o^i, \mu^i(o^i) o^{-i}, a^{-i})$$

Unfortunately, the primary algorithms we discuss here, DDPG and MADDPG are on-policy algorithms - that is, training the supervisor policy based on the actor's actions will not work. We demonstrate this failure in Case 1a below; in the experiments discussed below, comparison is, where possible, instead made to analytical solutions. Future work might test the deficit using an off-policy algorithm like PPO.

3.1.4. Bertrand Analytical Solution

We begin our experiments by seeking to replicate the well-known analytical solution for Bertrand competition; Bertrand competition models firms which produce at the same, constant marginal cost subject to no capacity constraint for a fixed, downward sloping demand-curve. This solution is useful because it is possible to derive both the Nash equilibria and the best-response functions of the firms involved.

Suppose we have a demand function $Q_D(P)$ (with corresponding inverse $P_D(Q)$), along with N firms selecting selling-prices p_i and producing at constant marginal cost c. We assume that all production is awarded to the firm with the lowest bid, with ties being broken evenly. Fix a particular agent i; when any other agent bids at or below the common marginal cost c, the agent's best-response is to likewise bid c. If the minimum of other agents' bids is between c and the monopoly price, the best response is to bid this same minimum price less an arbitrarily small positive number ε (in order to ensure it is awarded the full amount rather than tie). Finally, when all other firms have bid above the monopoly price, the best-response is to bid the monopoly price. Formally, if not illuminatingly, we may write this

$$p^i_{\mathsf{BR}}(p^{-i}) = \mathsf{min}(p_{\mathsf{monopoly}}, \mathsf{max}(c, \underset{i}{\mathsf{min}}(p^{-i}) - \varepsilon))$$

Clearly, for any number of agents $N > 1$, the only Nash-equilibrium is for all players to bid the marginal cost. The principal reason for considering Bertrand competition in this section of the work is that it admits a simple reformulation of the deficit formulated in the previous chapter. The deficit is simply the agent's best-response profit less its realized profit.

$$\lambda^i(p^{-i}) = Q^i_D(p^i_{BR}(p^{-i}), p^{-i})(p^i_{BR}(p^{-i}) - c) - Q^i_D(p^i, p^{-i})(p^i - c)$$

Where the contracted quantity $Q^i_D(p^i, p^{-i})$ is the amount awarded to agent i given other agents' actions - 0 if another agent undercuts it, and otherwise the whole demand $Q_D(p^i)$.

As before, this profit deficit may be considered a measure of how well agents converge to Nash equilibria.

3.2. Experiments

3.2.1. Neural Network Technicalia

For the sake of transparency we note here some of the parameter values used in training. All neural networks used in this work have two hidden layers of 300 nodes, with Layer-Norm and ReLU activations. The Adam optimizer was used. For exploration, the policy-generated act was added to zero-mean Gaussian noise with standard-deviation 0.3. All acts were scaled such that 1.0 corresponded to Q_{max} or MC. Rewards used for learning were scaled down by a factor of 1000. In all cases in this work, exploration occurs for the first half of the run only, while learning occurs throughout. A replay buffer stored the last $\text{num}_{\text{episodes}}/4$ episodes for learning (FIFO); batches of size 128 were drawn at random from this buffer; learning updates occurred every 128 timesteps.

3.2.2. Case 1: Price-Only, Hypercompetitive

For this scenario, we seek to demonstrate that the deep learning algorithms have been implemented correctly and are capable of performing reliably in the simplest of environments. As before, we examine a simple day-ahead clearing market; for this case, we restrict ourselves to price-only bidding in the hypercompetitive regime (in which each agent can unilaterally meet all demand), as this allows comparison of agent performance to the expected analytical solution of the Bertrand market discussed above. All agents' marginal costs are $c = 40$USD/MWh, while demand (in USD) is given by $P_D(Q) = 500 - 2Q$.

We examine three sub-cases in which agents compete only with respect to price (a la Bertrand competition): in the first, the DDPG agent is a monopolist (and so is expected to converge to the monopoly price). In the second, a DDPG agent competes with a naive marginal-cost bidder (the maximum profit is thus zero). In the third, two DDPG agents are pitted against one another to illustrate the failure to converge to a Nash equilibrium, analogous to the discrete case.

Case 1a: DDPG Monopolist

Per the parameters above, the monopoly solution is to sell at a price of $P = 270$USD $= 6.75c$, resulting in a quantity $Q = 115$MWh being sold for a total profit of $\Pi = 26450$USD.

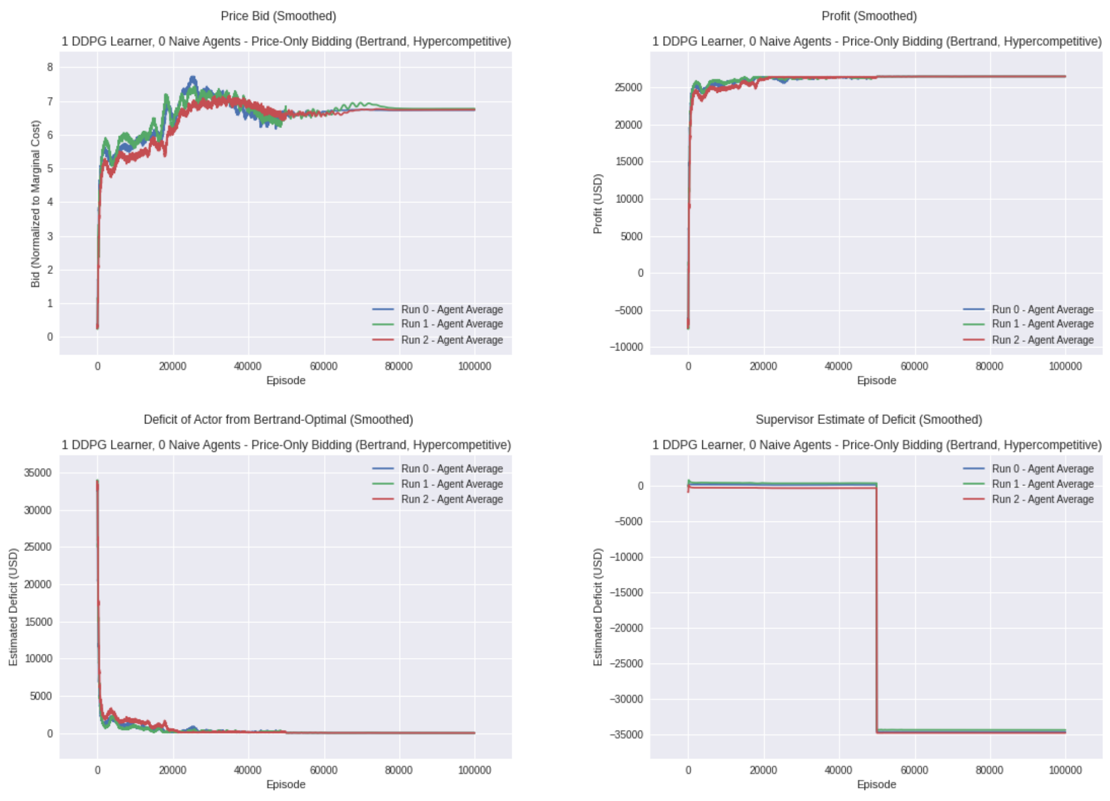


Figure 3.1: (Case 1a) Plots illustrating the performance of the DDPG algorithm for a simple, price-only market; 'Hypercompetitive' indicates that the agent's generator capacity is greater than the zero-price demand. All plots are shown for each of three independent runs of the simulation. Exploration was active for half of the total time, while learning occurred throughout. Top-left: Price bids over training. Top-right: Profit earned. Bottom-left: Deficit with respect to analytical Bertrand solution Bottom-right: Deficit estimated by the supervisor

In these plots, we observe that the DDPG learner converges to approximately the monopoly outcome in all the runs shown. Accordingly, the profit rapidly increases while the deficit with respect to the Bertrand best-response decreases. As noted in section 3.1.3, the supervisor, if trained using the same on-policy DDPG algorithm, fails to provide useful information about the deficit.

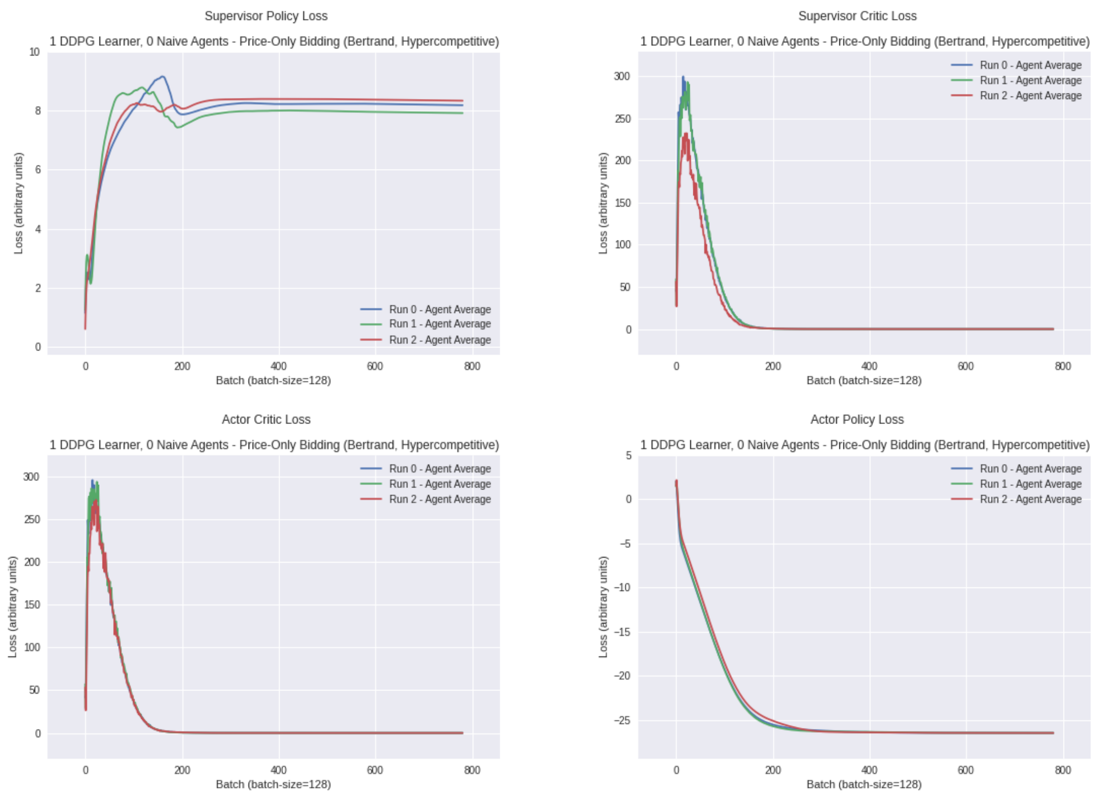


Figure 3.2: (Case 1a) Plots illustrating the performance of the DDPG algorithm for a simple, price-only market; 'Hypercompetitive' indicates that the agent's generator capacity is greater than the zero-price demand. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over all learning-agents (i.e. excluding marginal-cost bidders). Exploration was active for half of the total time, while learning occurred throughout. Top-left: Supervisor policy loss. Top-right: supervisor critic loss. Bottom-left: actor critic loss Bottom-right: actor policy loss.

Here we provide plots illustrating the training progress of the actor and the supervisor (each of which is associated with two neural networks, the policy and the critic). We observe that both critics converge rapidly (indicating that both quickly learn to determine what profit should be expected from a particular action). The policy of the actor converges rapidly as well, while the supervisor policy does not. As noted above, despite learning an effective Q-function, the supervisor fails to adequately estimate the deficit because this estimate relies on the supervisor policy to provide an approximately optimal action. It is worth noting that the Q-function learned by the supervisor works only incidentally due to the fact that the environment has only one timestep - otherwise it would depend on the supervisor policy and accordingly would not be learned adequately by an on-policy algorithm like DDPG.

Case 1b: DDPG vs. Marginal Cost Agent

In this case, we pit a DDPG learner against a naive agent which always bids its marginal cost - as a result, the optimal profit is zero, and the learner need only learn not to underbid.

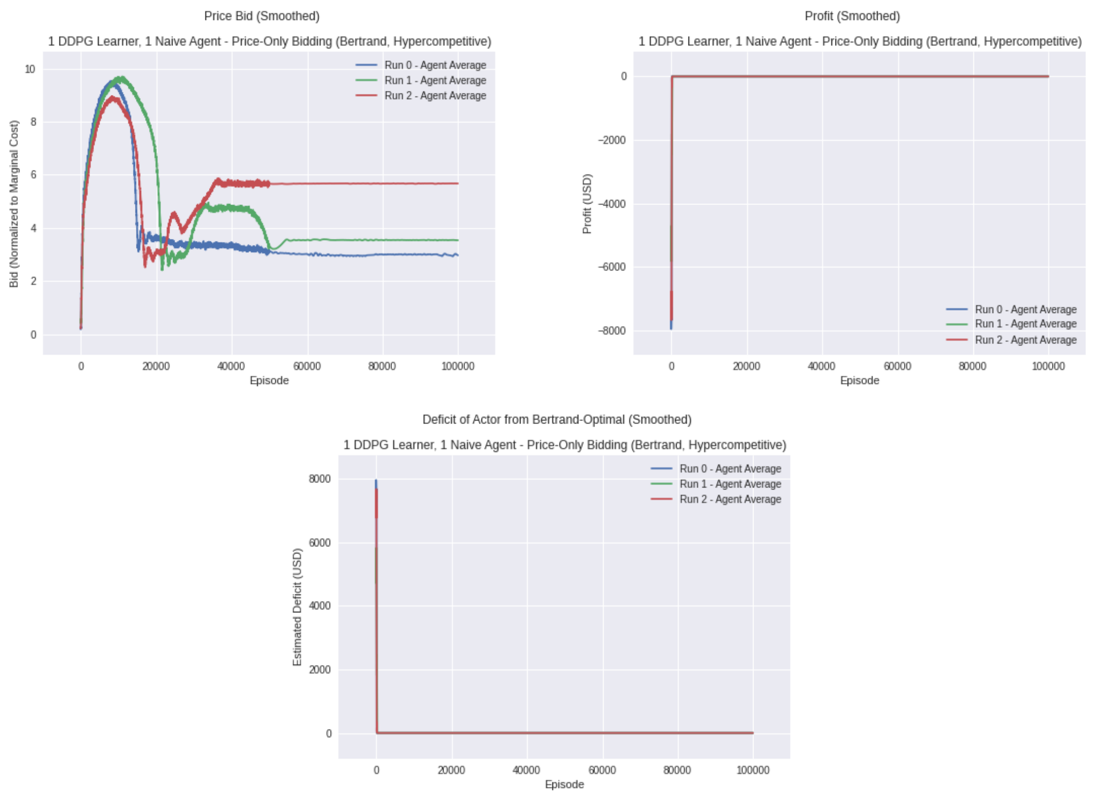


Figure 3.3: (Case 1b) Plots illustrating the performance of the DDPG algorithm for a simple, price-only market, competing against a single other agent which always bids its marginal cost. Top-left: Price bids over training. Top-right: Profit earned. Bottom: Deficit with respect to analytical Bertrand solution

For this case, we observe that the agent learns to bid different prices on each run - the reason for this is, of course, that these bids are never accepted; the marginal-cost-bidding agent ensures no bids above marginal-cost are accepted, so that the agent's profit is zero if it overbids, and negative if it underbids. Accordingly, we observe the profit and deficit reach their ideal values almost immediately.

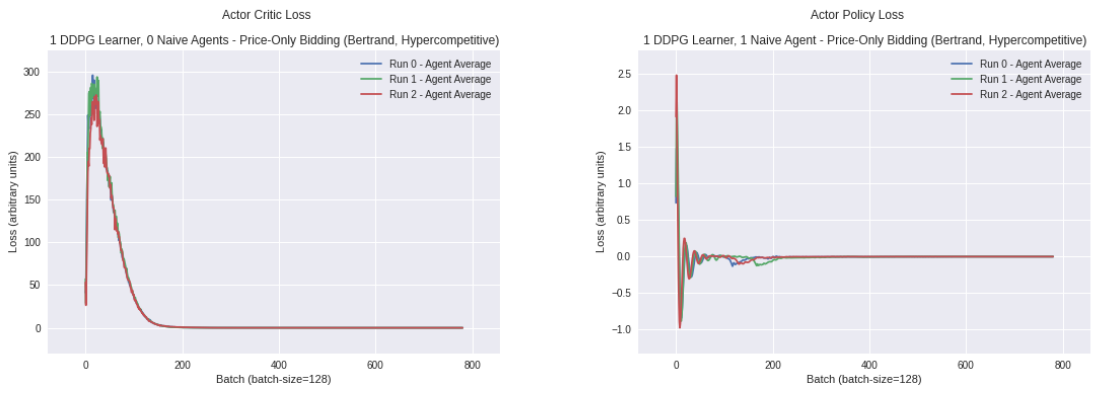
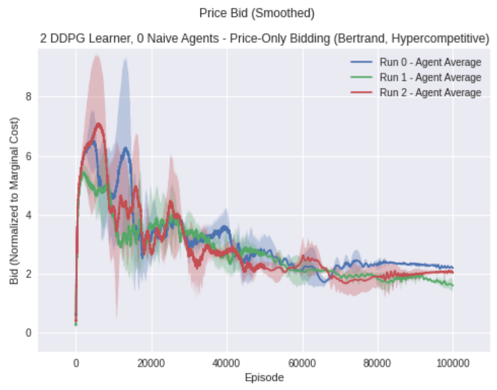
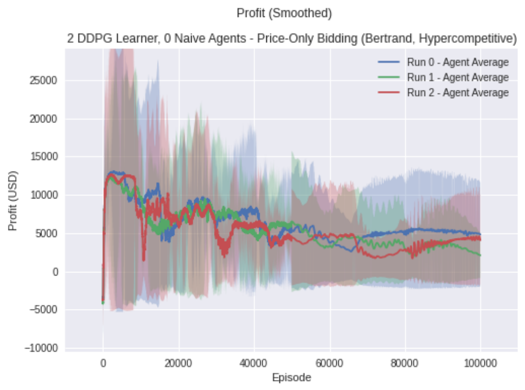


Figure 3.4: (Case 1b) Plots illustrating the performance of the DDPG algorithm for a simple, price-only market, competing against a single other agent which always bids its marginal cost. Left: Learner critic loss. Right: Learner actor loss.

Case 1c: DDPG vs. DDPG





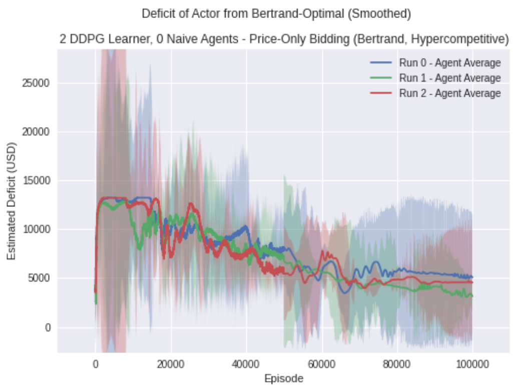
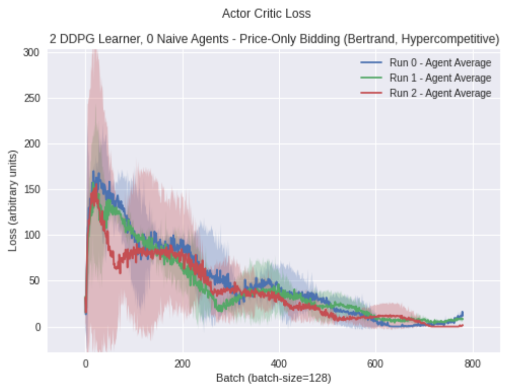


Figure 3.5: (Case 1c) Plots illustrating the performance of two competing DDPG algorithms for a simple, price-only market. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over all learning-agents (i.e. excluding marginal-cost bidders). Top-left: Price bids over training. Top-right: Profit earned. Bottom: Deficit with respect to analytical Bertrand solution

Here we begin to observe the effects of non-stationarity on training the DDPG agents. The correct solution to this game is the competitive outcome in which both agents bid their marginal cost; on the other hand we see the two agents stabilizing to a bid price of roughly double the marginal cost. Likewise both the profit and the deficit fluctuate substantially without clearly converging to 0.



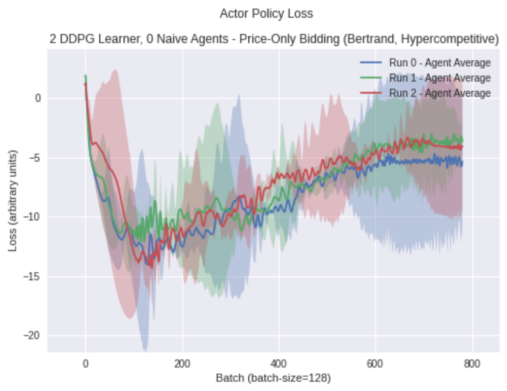


Figure 3.6: (Case 1c) Plots illustrating the learning performance of two DDPG learners pitted against one another in a simple, price-only market. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over all learning-agents. Left: Learner critic loss. Right: Learner actor loss.

Examining the learning performance of the competing DDPG learners, we see that the critics gradually learn correct predictions, but not rapidly enough for the learner to adapt within the simulated timeframe.

Case 1d: MADDPG vs. MADDPG

Here we examine what happens if we change the two DDPG learners to instead use MADDPG.

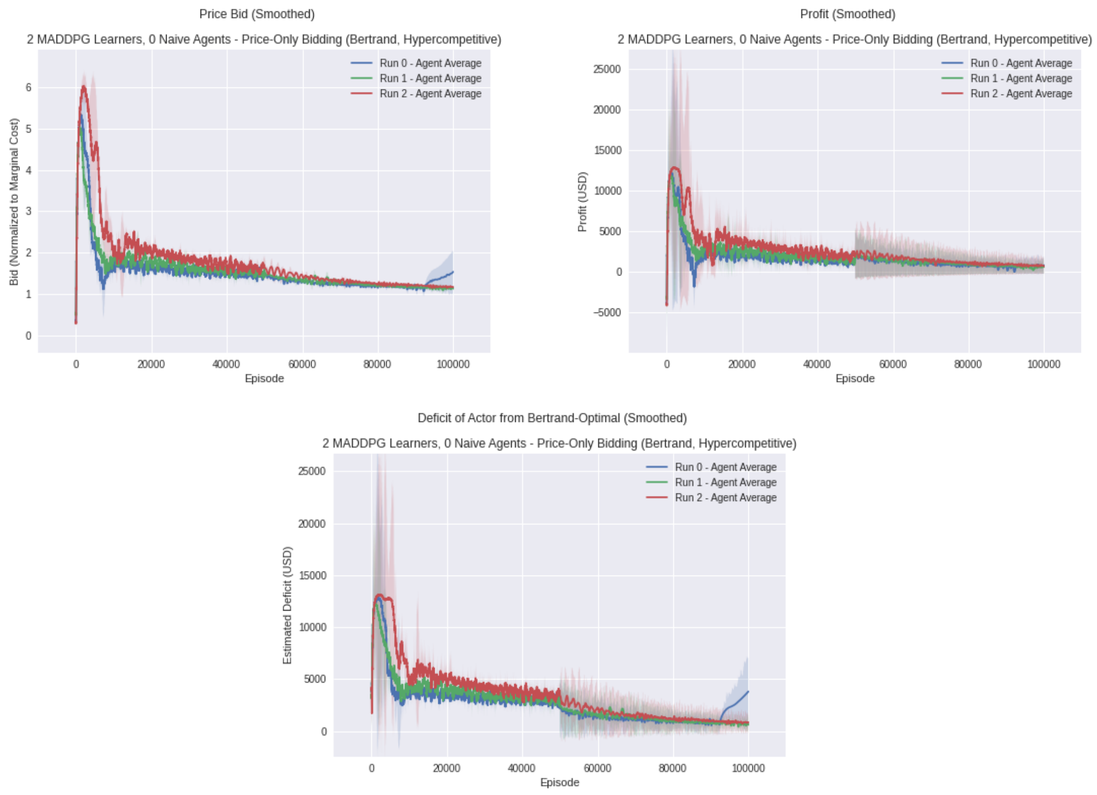


Figure 3.7: (Case 1d) Plots illustrating the performance of two competing DDPG algorithms for a simple, price-only market. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over the two learning-agents. Top-left: Price bids over training. Top-right: Profit earned. Bottom: Deficit with respect to analytical Bertrand solution

Here we observe that MADDPG essentially resolves the problems which DDPG encountered above; the learners approach the competitive outcome. Accordingly, the profit earned, as well as the best response go to zero.

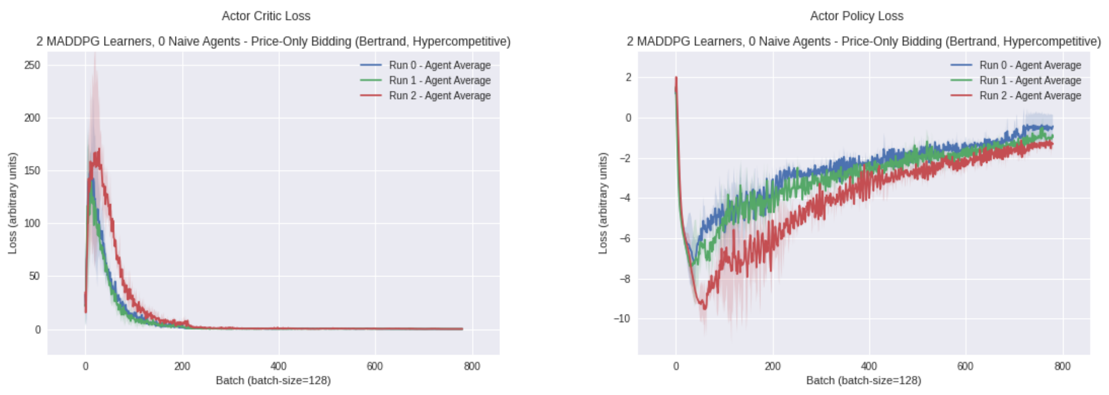


Figure 3.8: (Case 1d) Plots illustrating the learning performance of two MADDPG learners pitted against one another in a simple, price-only market. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over all learning-agents. Left: Learner critic loss. Right: Learner actor loss.

3.2.3. Case 2: Q,P Bidding, Competitive

In this case, the action-space of the agents is extended to permit an agent to choose both the price and quantity of its bid. Further, the capacity of each agent is lowered (to $Q_{\text{max}} = 200MW$ so that no individual agent can meet all demand alone. The analytical solution to this environment is the oligopoly. This increase in complexity is meant to illustrate that DDPG learners are capable of operating in (slightly) more complex environments, but fail when pitted against one another. In such a case, MADDPG is capable of converging where independent-learning algorithms like DDPG fail to do so.

Case 2a: DDPG vs. Marginal Cost Agent

Here we examine the case of a DDPG learner bidding against a naive agent which always bids its marginal cost at full quantity. As the marginal cost agent cannot meet demand on its own, the learner has some latitude to behave strategically.

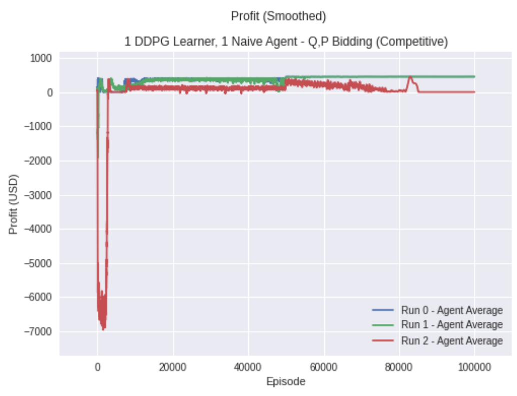


Figure 3.9: (Case 2a) Plot illustrating the profit of the DDPG algorithm for Q,P-bidding against a single marginal-cost agent; 'Competitive' indicates that no agent can unilaterally satisfy all demand, but that all agents together can. Plots are shown for each of three independent runs of the simulation.

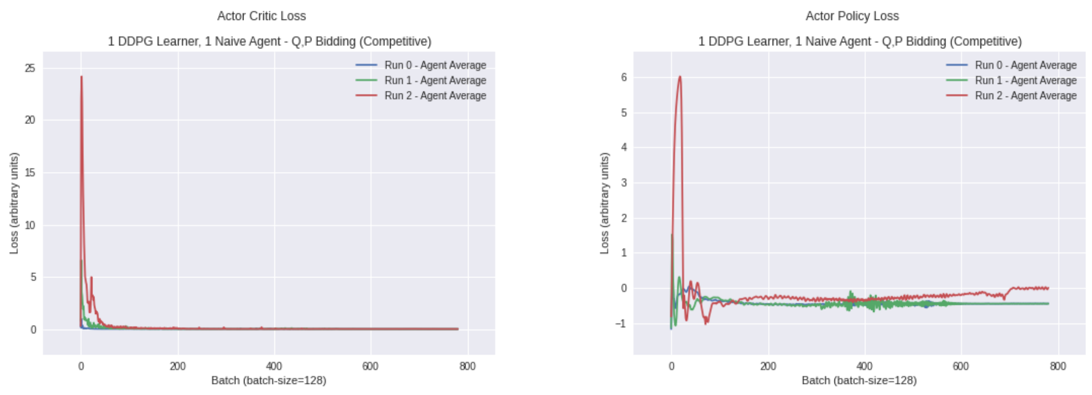


Figure 3.10: (Case 2a) Plots illustrating the performance of the DDPG algorithm for Q,P-bidding against a single marginal-cost agent; 'Competitive' indicates that no agent can unilaterally satisfy all demand, but that all agents together can. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over the two learners. Left: actor critic loss. Right: actor policy loss.

Case 2b: DDPG vs. DDPG

Here we examine the case in which two independent DDPG learners compete with one another.

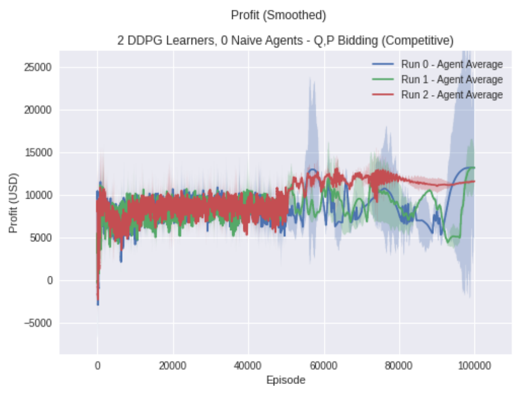


Figure 3.11: (Case 2b) Plot illustrating the profits of two independent DDPG learners for Q,P-bidding. Plots are shown for each of three independent runs of the simulation; curves are averaged over the two learners. Left: price bids over training. Right: profit earned.

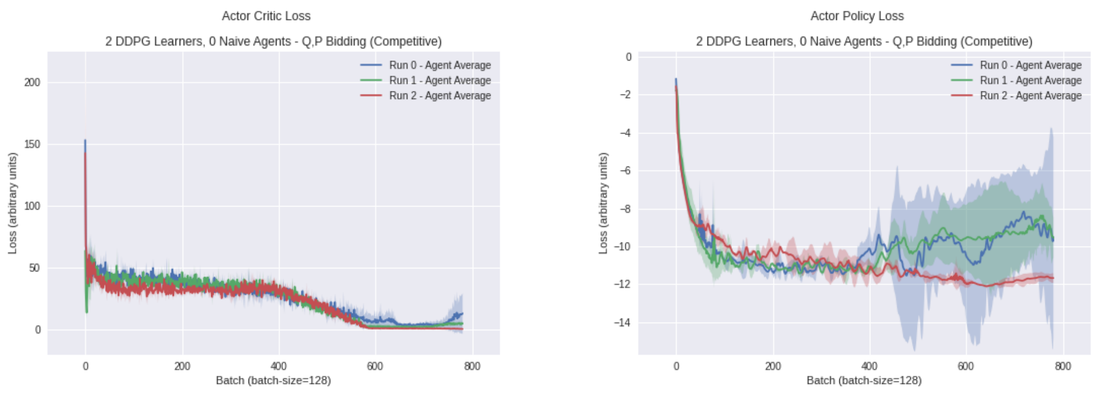


Figure 3.12: (Case 2b) Plots illustrating the performance of two independent DDPG learners for Q,P-bidding. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over the two learners. Left: actor critic loss. Right: actor policy loss.

We observe in Figure 3.11 that in all cases, the learners converge to achieving an appreciable profit, if not without some subtantial fluctuations. In Figure 3.12, we see the critics converge relatively rapidly, while the actors exhibit some fluctuation.

Case 2c: MADDPG vs. MADDPG
Here we examine the case in which two independent MADDPG learners compete with one another.

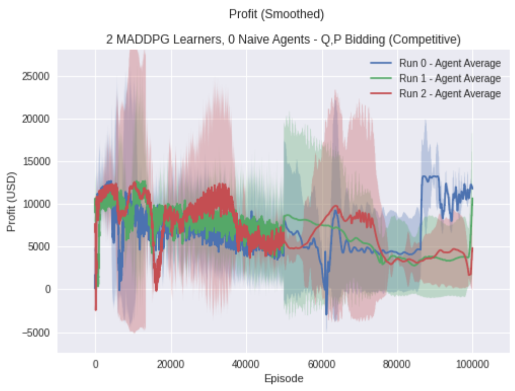


Figure 3.13: (Case 2c) Plots illustrating the performance of two MADDPG learners for Q,P-bidding. All plots are shown for each of three independent runs of the simulation; curves are averaged over the two learners. Left: price bids over training. Right: profit earned.

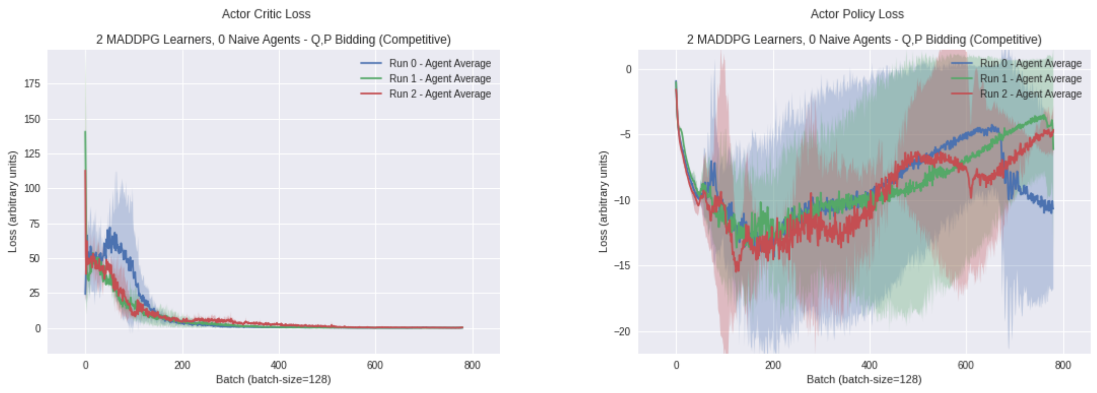


Figure 3.14: (Case 2c) Plots illustrating the performance of two MADDPG learners for Q,P-bidding. All plots are shown for each of three independent runs of the simulation; plotted curves are averaged over the two learners. Left: actor critic loss. Right: actor policy loss.

We observe in Figure 3.13 that in all cases, the learners converge to achieving an appreciable profit, if not without some subtantial fluctuations. In Figure 3.14, we see the critics converge quite rapidly, the actors seem to fluctuate quite dramatically. Surprisingly, the MADDPG learners seem to flucuate much more dramatically than their DDPG counterparts.

3.3. Convergence Proofs

We are now in a position to discuss mathematically this mis-convergence phenomenon. To begin with, we note that the puzzle is that both DDPG (subscript D) and MADDPG (subscript M) policies, should ideally be the same; if everything converges perfectly, we have:

$$\pi^i_M = \text{argmax}_{a^i} Q_M(a^i, \pi^{-i}_M) = \text{argmax}_{a^i} r(a^i, \pi^{-i}_M)$$
$$\pi^i_D = \text{argmax}_{a^i} Q_D(a^i) = \text{argmax}_{a^i} r(a^i, \pi^{-i}_D)$$

That is, the policies are the same. The problem is a combination of the estimation of the Q-functions with the exploration policies.

In what follows, we seek to prove sufficient conditions for convergence to a correct equilibrium; to make this simpler, we consider the discrete case with only two agents; the generalization to many agents is straightforward, while convergence theorems are impractical to prove for the continuous case in which Q-function updates rely heavily on the complicated neural network parameterization and specifics of the optimizer used for gradient-descent.

First we consider the centralized policy: Let Q_t^i be a matrix whose entry i, j represents the agent's Q-value estimate of the reward obtained for itself and its opponent taking actions a_i and a_j, respectively. Upon the action-pair (a_i, a_j) occuring, an update rule dictates that agent k updates as

$$Q_{t+1}^k = f(Q_t^k, R_{ij}^k \delta_{ij})$$

(where δ_{ij} is the matrix which is 1 in entry i, j and 0 in all others) For instance, the soft-update rule has the form $f(A, B) = A(1 - \delta_{ij}) + \delta_{ij}(\rho A + (1 - \rho))B$. This is simply to say that the update rule updates only the entry for the action-pair which is observed, which updates to a weighted average of the estimated and observed value.

The following condition suffices to show that an update rule for Q is non-increasing: if we have for some matrix norm $|| \cdot ||$ that for any observation (a_i, a_j)

$$||Q_{t+1}^k - R^k|| \leq ||Q_t^k - R^k||$$

then the update-rule is non-increasing and bounded above by $||Q_0 - R||$, and bounded below by 0 - thus the sequence converges (not necessarily to 0).

Then, requiring that, for each possible observation-pair, (a_i, a_j), $Q_{t+1,ij}^k = R_{ij}^k$ or that the observation-pair occur with non-zero probability, and upon occuring, cause $|Q_{t+1,ij}^k - R_{ij}^k| \leq \rho |Q_{t,ij}^k - R_{ij}^k|$ for some $0 \leq \rho < 1$. That this criterion is sufficient for convergence follows from considering any observation-pair (a_i, a_j) where $|Q_{t,ij}^k - R_{ij}^k| = c_t > 0$ for some value c_t. Let the stopping-time $\tau_1 = t$ denote the event that t is the first time such that $c_t > c_{t-1}$. By hypothesis, τ_1 is finite with probability one. Then let τ_2 be the stopping-time associated with the second such decrease - conditioning on $t \geq \tau_1$ (a set which is nonempty as τ_1 is finite), τ_2 is likewise finite, etc. Thus, for any integer N, there exists, with probability one, a time t_N such that (a_i, a_j) has been drawn N times; in each case, the mismatch $|Q_{t,ij}^k - R_{ij}^k|$ decreases by at least a factor of $\rho < 1$; as this is true for each pair of indices and for all N, Brouwer's fixed-point theorem [3] implies that the Q-esimtate converges element-wise and almost-surely to the true reward. Thus we have:

Theorem: Let Q_t^k denote agent k's estimated Q-function as a matrix whose entries correspond to possible action pairs of all agents. Similarly, let R^k be a matrix denoting the actual reward obtained, and $f(Q^k, R_{ij}^k \delta_{ij})$ be an update rule which alters the Q-estimate upon observing a reward R_{ij}^k corresponding to the action pair i, j. Then, if there exists a matrix norm $|| \cdot ||$ such that $||Q_{t+1}^k - R^k|| \leq ||Q_t^k - R^k||$, then the error $||Q_t^k - R^k||$ converges to a constant c as $t \to \infty$.

Theorem Suppose the following three conditions on are met:

- Seeing the i, j-action pair, the update rule alters only Q's i, jth entry
- For each i, j there exists a ρ, $0 \leq \rho < 1$, $|Q_{t+1,ij}^k - R_{ij}^k| \leq \rho |Q_{t,ij}^k - R_{ij}^k|$
- All action-pairs (a_i, a_j) for which $Q_{ij}^k \neq R_{ij}^k$ occur with non-zero probability

Then Q_t^k converges element-wise and almost-surely to R^k as $t \to \infty$.

A stateful MDP with multiple timesteps can always be transformed into a multi-agent, stateless, timeless MDP such as the one discussed above; each agents' transformed action is a policy over all partial-trajectories. Thus,

Corollary The convergence theorem stated above applies to stateful MDPs with multiple timesteps if a nonstandard Q-function over trajectories is used.

Let us now turn to the independent case, analogous to DDPG. In this case, the Q-estimate can be written as a vector q_t^k (or as a matrix whose columns are all q_t^k - one for each of the opponents' actions). The analogous update rule is, for an observation of the i, j action-pair,

$$q_{t+1}^k = f(q_t^k, R_{ij}^k \delta_i)$$
$$[q_{t+1}^k, q_{t+1}^k, \ldots] = [f(q_t^k, R_{ij}^k \delta_i), f(q_t^k, R_{ij}^k \delta_i), \ldots]$$

(the second-line is the matrix-representation, for more direct comparison)

It is immediately clear that the theorems above will not apply in general. The contraction property cannot be assumed without further assumptions, and further, such an update rule necessarily entails altering entries of the Q matrix which have nothing to do with the observation. Intuitively, provided that the second agent's actions effect the first's reward, it can cause the first agent's q-function to oscillate merely by consistently playing one action for a long time, and then playing a different action for a long time, and so on.

Whether or not this misgeneralization occurs in practice thus clearly depends on the update rule and the behavior of the other agent.

4 Adversarial Market-Design

Now that we have examined the behaviour of various algorithms in the simple day-ahead market, we turn our attention to analyzing more complex models of the electricity market which allow for the fruitful application of MARL methods. First, we consider once more the day-ahead market, though this time extended with an adversarially selected price-cap. This is relevant as it illustrates directly the utility of MARL methods for market-design.

Second, we extend the work of the previous section to cases in which the environment is stateful - that is, there are multiple timesteps per episode; in particular, we implement a simple physical market with redispatch. This is valuable both because it illustrates an extension of the previous work, as well as highlighting the ability of RL algorithms to operate in more complex coupled-market environments.

The purpose of this section is not, of course, to present policy-advice based on such a simplistic model; instead, it is hoped that the simpleness of the model will render the application and its limitations more legibly than a fully realistic environment. A real application of this technique would entail careful selection of market-designs such that they can be usefully represented as the output of a neural network, as well as a more thoroughgoing consideration of the appropriate state- and action-spaces.

4.1. Adversarial Market Design

In general, market-design is the discipline of translating objectives into market-design parameters. If these objectives can be translated into some generalization of the ”social-welfare” function from economics, and these parameters can be quantified, it is possible to phrase the problem of market design itself in terms of reinforcement learning.

As an example, consider the case of an oligopolistic market with known supply and demand curves; a market-designer might take their goal to be the maximization of the social-welfare function, and aim to set bidding rules accordingly. Note that, many market-design questions entail deciding between two discrete alternatives (e.g., a nodal vs. a zonal pricing-scheme) in such cases, it is possible to employ tabular methods, though this is likely not an effective method for performing such analyses.

In the language of (single-parameter) mechanism-design, a market rule consists of an outcome-rule $x(b)$ mapping a set of bids b to a set of good allocations, and a payment-rule $p(b)$ mapping a set of bids to payments exacted upon each bidder. One must further specify the bidders' utility functions, $u(b)$ [25]. Each bidder values a good differently, modeled as corresponding to a vector of random values V (realized values are known only to their bidder). Now, for a given market, it is probably the case that bidders' utility functions are not subject to market-design; on the other hand, the allocation and pricing rule likely are. The revelation principle [19] implies that for any mechanism, there exists a mechanism such that it is a weakly dominant strategy for all agents to submit their bids as truthful reports of their own values; such mechanisms are known as ”dominant strategy incentive compatible” (DSIC).

In the case of an electricity market, then, bidders are power-producing firms while market-design parameters might include things such as a price-cap and/or floor, or more complicated tiered price-restrictions which set a dynamic price-cap according to demand. Meanwhile, the bids correspond to agents' submitted (Q, P) pairs. A market with fixed demand-function $P_D(Q)$, according to the revelation principle, has some clearing-rule such that truthfully reporting (Q_{max}, MC) is a dominant strategy for the producing firms - this cannot be the standard merit-order clearing-rule, as this rule in general allows agents to profit by e.g., strategically withholding capacity.

In what follows, we will consider a market designer attempting to optimize the social-welfare function

$$\text{SW} = 2\int_0^{Q^*} dq(\alpha P_D(q) - (1-\alpha)P_S(q)) + (1-2\alpha)P^*(Q^*)$$

(a social welfare-function which (dis-)privileges consumer welfare relative to producer welfare according to a factor α ($0 \leq \alpha \leq 1$)).

For ease of reference, we call α the "adversariality", as it dictates how strongly 'adversarial' the market agent is to the firm agents.

In contrast to the typical mechanism-design approach which seeks DSIC payment rules from truthful bids, one may wish to find the optimal parameter-value for a given family of market-designs. This might be of use in, say, setting an optimal price-cap, as in the example we will consider; it again bears noting that the present example is meant as an illustration of the technique rather than an attempt to solve any particular market-design problem.

The fact which makes this example interesting is that, rather than requiring truthful bidding, MARL methods allow us to train a market-designer seeking to maximize social-welfare while *simultaneously* training agents to simulate strategic behavior under a given market rule! The work in previous sections allows us to be comfortable that the centralized-decentralized training (a la MADDPG) regimen should allow the correct derivation of equilibria, at least in principle.

4.1.1. Experiment Setup and Analytical Solution

The setup is as follows: N producer firms are in posession of identical generators with $MC = 40$USD. For the sake of crisply highlighting the effect on strategic behavior, we consider a Cournot model in which agents compete solely on quantity without capacity constraints, with the corresponding price determined by the market. The demand is modelled as a simple linear function $P_D(Q) = b - mQ = 500 - 2Q$. When demand exceeds supply, the price is set to the demand's price offer for the supplied quantity.

The price-cap to be set by the market-designer agent, $\bar{p}$, implemented by replacing the consumer-demand with an effective demand-curve $P_{D,\text{eff}}$

$$P_{D,\text{eff}}(Q) = \begin{cases} \bar{p} & P_D(Q) \geq \bar{p} \\ P_D(Q) & P_D(Q) \leq \bar{p} \end{cases}$$

This said, we may fix the price-cap in order to find the oligopoly bids (as a function of $\bar{p}$); then, we may derive an analytical expression for the optimal price cap from the social-welfare function.

Denote the total quantity bid by Q, and individual agents' bids by q^i. Now for a given agent, the profit obtained is

$$\Pi^i = (P_D(Q) - c)q^i$$

In the absence of any price-cap, the agent's best response to a set of other agents' bids Q^{-i} would be to bid

$$q^i_{\text{BR}}(Q^{-i}) = \frac{1}{2m}(b - c - mQ^{-i})$$

Which, applied to all agents, yields the equilibrium bid:

$$q^i_{\text{olig}} = \frac{b-c}{m(N+1)}$$

with corresponding price and individual firm profit

$$p_{\text{olig}} = \frac{b+Nc}{N+1}$$

$$\Pi^i_{\text{olig}} = \frac{1}{m}\left(\frac{b-c}{N+1}\right)^2$$

Once the price cap is in place (assuming it is above the marginal cost, as the optimal bid is otherwise trivially zero), the above solution is no longer necessarily an equilibrium - where a decreasing demand would have offset the gains from increasing quantity, now this demand is, for some range of quantities, fixed. If the price-cap is above the oligopoly-price, the oligopoly solution maintains, as it remains the point at which any unilateral change in a quantity is penalized.

On the other hand, if the price-cap is below the oligopoly-price, the 'crossover-point' (the quantity demanded by consumers at the price-level set by the price-cap, $P_D(Q) = \bar{p}$) is the equilibrium solution. To see this, note that, while below both the oligopoly price the price-cap, any unilateral restriction in quantity increases a firm's profit. This profit increases until the price-cap is hit, at which point a decrease in quantity no longer changes demand, so that decreasing quantity is now penalized.

The best-response bid can be calculated by examining the sign of the cap-modified profit (again considering only the case where the cap is above the marginal cost). The derivative of the profit in the price-capped region is positive by assumption; at the crossover point, this either becomes negative, so that the crossover point is optimal, or remains positive, so that the capless best-response maintains. The post-crossover profit derivative is positive as long as the capless best-response is greater than the crossover quantity.

Then

$$q^i_{\text{BR}}(Q^{-i}, \bar{p}) = \begin{cases} \frac{b-\bar{p}}{m} - Q^{-i}, Q^{-i} < \frac{b+c-2\bar{p}}{m} \\ \frac{b-c}{2m} - \frac{1}{2}Q^{-i}, \text{else} \end{cases}$$

The equilibria are now slightly more diverse - when the oligopoly price is realizable, this is the sole equilibrium. However, if the price-cap prevents this, then *any* set of bids which sum to $\frac{b-\bar{p}}{m}$ is an equilibrium. The producer surplus is

$$\text{PS} = \begin{cases} \frac{N}{m}\left(\frac{b-c}{N+1}\right)^2, \frac{b+Nc}{N+1} \leq \bar{p} \\ \frac{1}{m}(b-\bar{p})(\bar{p}-c), \text{else} \end{cases}$$

The consumer surplus is then

$$\text{CS} = \begin{cases} \left(\frac{N}{N+1}\right)^2 \frac{(b-c)^2}{m}, \frac{b+Nc}{N+1} \leq \bar{p} \\ \frac{1}{2m}(b-\bar{p})^2, \text{else} \end{cases}$$

Thus, for our market designer (who weighs consumer surplus by a factor 2α and producer surplus by a factor $2(1-\alpha)$), the social welfare function is

$$\text{SW} = \begin{cases} 2\frac{(b-c)^2}{m(N+1)^2}(\alpha N + (1-\alpha)N^2), \frac{b+Nc}{N+1} \leq \bar{p} \\ \frac{b-\bar{p}}{m}(\bar{p}(3\alpha-1) + (b(1-\alpha) - 2c\alpha)), \text{else} \end{cases}$$

Because of the many overlapping constraints, theinversion to obtain the optimal price cap is not readily attainable or simply expressed; instead, we not a few points to guide the reader's intuition in interpreting the experimental results below.

First, in general, to the extent that they are free to choose their prices, the strategically acting firms should cause a deadweight loss. Clearly, when $\alpha = 1$, the optimal price cap is the marginal cost (plus an arbitrarily small real number) - this maximizes not only the consumer welfare, but the total social welfare for any $\alpha > 1/2$ - since lowering the cap in this regime converts producer welfare into a strictly greater amount of consumer welfare (which is also valued more). When $\alpha < 1/2$, the price cap can force the oligopolists to produce a quantity greater than the oligopoly quantity, but not less - thus the optimal cap in this region involves trading off (relatively more valuable) producer welfare to consumer welfare in an attempt to recoup by decreasing the deadweight loss.

4.2. Experiment

Here we present the results of the experiment detailed above. We begin by presenting some example plots showing the training-progress over time. After these example plots have been shown, we present a "sweep" of parameter-space, in which we attempt to determine the optimal price-cap as a function of α and of N.

4.2.1. Example Case 1: One Firm, $\alpha = 0.25$

For our first example case, we run the experiment with a single producing firm, and a market-agent with adversariality $\alpha = 0.25$ - meaning that it should prefer to improve producer welfare rather than consumer welfare. Based on our earlier reasoning, we should expect to see little effect from the price-cap, the most producer-surplus can is to be had at the oligopoly point.

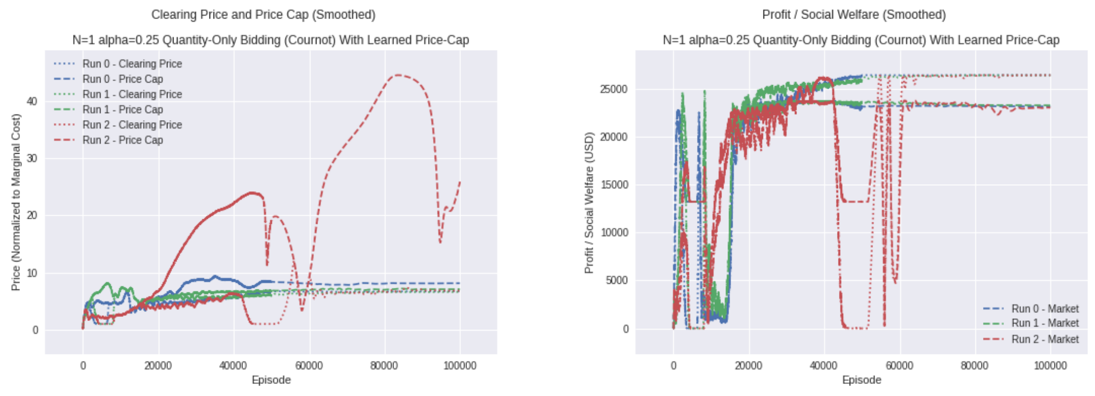


Figure 4.1: (Example Case 1) Plots illustrating the learning progress of one firm agent and the market agent, with adversariality $\alpha = 0.25$. Left: price bid and price cap. Right: profit and social welfare

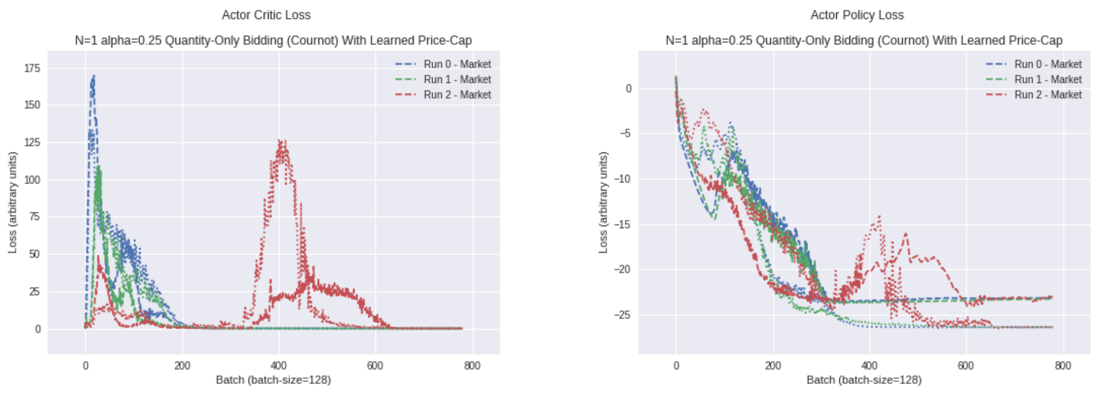


Figure 4.2: (Example Case 1) Plots illustrating the learning progress of one firm agent and the market agent, with adversariality $\alpha = 0.25$. Left: critic loss. Right: policy loss

And indeed - in spite of some erratic behavior (attributable to the fact that very high price-caps are equi-welfare and thus leave the neural-network optimizer with some 'momentum', causing outputs to continue to change), we observe in Figure 4.1 that in all three runs, the price cap tends to be very high - almost always above the monopoly price; further, the social-welfare function illustrates that the market-agent's welfare is practically the same as the agent's surplus.

With respect to the learning shown in Figure 4.2, it is noteworthy that even in this fairly simple case, both the actor and the critic exhibit fairly erratic behavior, though they do ultimately converge to a reasonable solution

4.2.2. Example Case 2: Five Agents, $\alpha = 0.75$

We now perform the analogous experiment, though in this case, we have five power producing firms, and an adversariality of $\alpha = 0.75$ - generally favoring consumers. Since consumers can be favored without creating any deadweight loss, we should expect the price-cap to approach the generators' marginal costs.

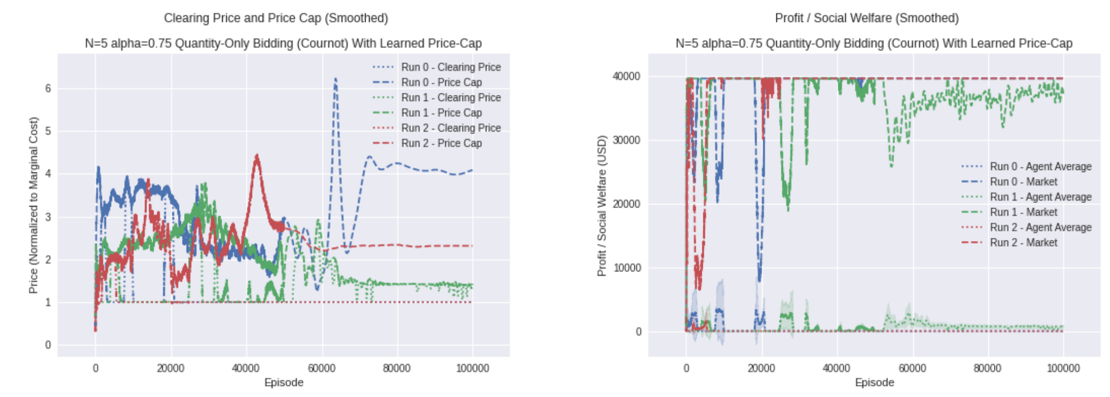


Figure 4.3: (Example Case 2) Plots illustrating the learning progress of five firm agents and the market agent, with adversariality $\alpha = 0.75$. Quantities for the firm-agents are averaged over all five agents, with the shaded region indicating the standard deviation over agents. Left: price bid and price cap. Right: profit and social welfare

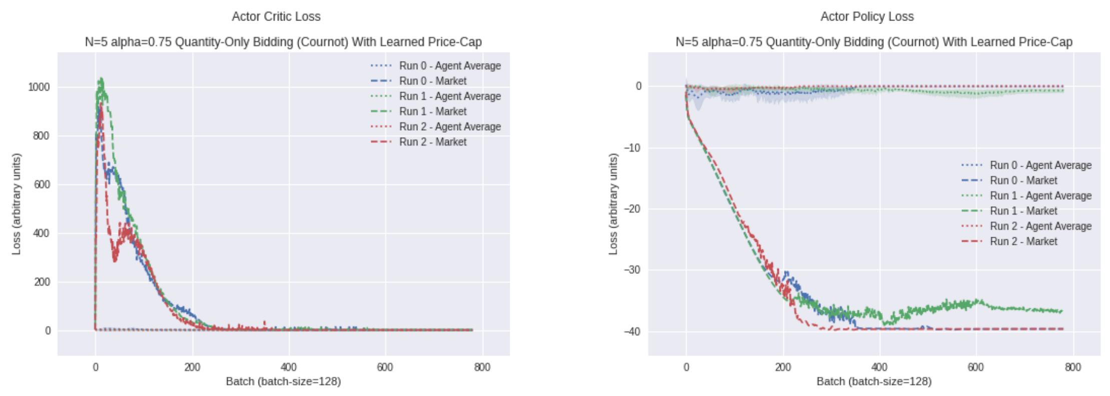


Figure 4.4: (Example Case 2) Plots illustrating the learning progress of one firm agent and the market agent, with adversariality $\alpha = 0.75$. Quantities for the firm-agents are averaged over all five agents, with the shaded region indicating the standard deviation over agents. Left: critic loss. Right: policy loss

Again, in spite of some erratic learning behavior, we see in Figure 4.3 that the clearing prices tend to be low, while the market-agent's welfare tends to be high. It seems that much of the odd fluctuations of the price-cap coincide with periods of low clearing price - which is often, given the number of agents; as noted above, the market-agent when the price-cap is inactive, may learn erratically, though this is not a problem provided this does not interfere with the reward.

Perhaps surprisingly, we observe in Figure 4.4 that the learning proceeds with relatively few hiccups - though there is a marked difference in the scales of the producers' loss compared to the market, this is not especially important provided convergence is achieved; this scale difference is due in part to the fact that the producers' rewards must be split among the five agents, while the market-agent keeps its own reward undivided.

4.2.3. Varying the Number of Firms and the Adversariality

Now that we have presented these two example cases, we may move on to the final part of this section. Here we present results of a "parameter sweep", showing what the final clearing prices and price caps were found to be after running the experiment above for each possible combination of (the number of agents) $N = 1, 3, 5$ and $\alpha = 0.0, 0.25, 0.50, 0.75, 1.0$ with three runs per data-point. The "final" values used are the average over the previous 1000 timesteps. We plot for reference the oligopoly prices for each N

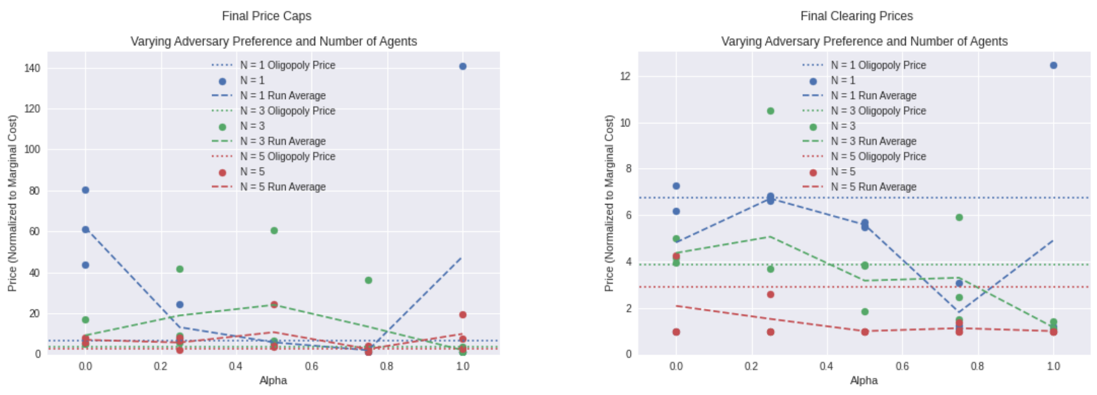


Figure 4.5: (Parameter-Space Sweep) Plots illustrating the behavior of the final clearing price and price cap. Left: learned price-cap. Right: Clearing-Price

Looking at Figure 4.5, we see that, generally, the actual clearing prices more closely attend the scale of the oligopoly prices - that is, while it seems that the learned price-caps tend to be extremely high for the reasons noted previously, this ultimately cashes out as achieving clearing-prices which are much more reasonable.

As α goes from 0 to 1, we observe that the clearing prices broadly tend to decrease as expected, though this trend is bucked by some outliers. At $\alpha = 0$, the clearing price is appreciably close to that which would be achieved in the oligopoly case, and when $\alpha = 1$, most of the clearing-prices tend to be very close to the marginal cost, as expected. This lends some credibility to the points of middling α which we cannot readily determine analytically, at least insofar as the follow the predicted trends.

Thus, we see that this 'adversarial market design' is capable of providing a reasonably good guess at the optimal price cap for reward functions of varying adversariality. While this result is encouraging, it is clear that the learning behavior of the algorithm is unstable, if only in cases where it doesn't much effect the reward - it is unclear to the author if this is a feature principally of the family of designs considered (i.e. of the price-cap mechanism itself), or of the learning-procedure.

5 Conclusion

5.1. Recapitulation

The present work has focused on offering two kinds of contribution, the first to the RL methodology for energy markets, the second for the introduction of adversarial market design as a tool for market designers. The overarching goal, of course, is the effective design of energy markets, with an aim towards a more effective energy transition.

Chapter 1 offered an introduction to the problem along with a review of the literature informing the present work; it also presented brief expositions of both reinforcement learning and game theory, outlining the core ideas and introducing the notation used in the rest of the work.

Chapter 2 introduced the idea of non-stationarity, its relationship to RL, and to economics. We give a simple example of the phenomenon, before going on to discuss the TD error and the optimality deficits which we use to gauge agents' performance. We describe a discretized day-ahead market environment. Subsequently, we examine several permutations of this scenario, altering the number of agents and the number of available hidden-states. In each case, we examine the behavior of the learning agents and discuss their convergence properties.

In Chapter 3, we extend the ideas of Chapter 2 to the continuous-regime; we introduce a new framework for formulating the optimality deficit, as well as the DDPG and MADDPG algorithms. Adapting the day-ahead market environment from Chapter 2, we examine different combinations of Naive marginal-cost, DDPG, and MADDPG agents bidding in either price-only or (Q, P)-bidding environments. In the price-only case, we verify the operation of the DDPG algorithm by comparison with the known Bertrand solution, and show that DDPG agents competing against one another indeed fail to converge to the solution due to non-stationarity, while MADDPG agents do. We perform analogous tests in the (Q, P)-bidding environment, generally observing a greater amount of fluctuation in the learning-process. Finally, we present a mathematical discussion of non-stationarity and prove conditions under which centralized-decentralized algorithms like MADDPG will converge (simultaneously highlighting why independent-learners like DDPG cannot).

In Chapter 4, we introduce the idea of adversarial market design. After a brief comparison to the field of mechanism design, we outline the Cournot solution to the day-ahead market and analyze the effect of a price-cap. Subsequently, we perform experiments in which a market-agent attempts to learn the optimal value for the price-cap, given a particular set of preferences, in the teeth of strategic behavior from the producing agents. Two examples of such experiments are presented and briefly discussed, after which we present the results of a parameter-space sweep. In this sweep, we vary the number of agents N and the adversariality α. We find that consumer-preferring market-agents tend to achieve clearing-prices near the marginal cost, while producer-preferring market-agents practically refuse to implement a price-cap, achieving roughly oligopolistic prices.

5.2. Future Work and Conclusion

Much of the methodological work done here focuses on the restricted case of RL with no state, as we attempted to minimize the number of complicating factors. However, it is clear that the convergence-proofs presented do not straightforwardly extend to cases with multiple-timesteps (except for the weakened version applying over whole trajectories). Thus, one productive direction of future work would be the examination of algorithms which can overcome this problem - of particular interest from the game-theory / economics perspective is the possibility that agents learn models of each others' behavior.

While there is a large variety of directions in which the methodological aspect of this work could be taken, the more germane to our purpose is what could be done with adversarial market-design. In particular, it is clear that new techniques are likely needed to allow market-agents to sufficient freedom to design a market at more than just the level of a single parameter as here; there is also certainly much practical work to be done creating stabler versions of the present algorithms, inventing extensions to more complex environments, experimenting with different learning architectures, and interfacing more thoroughly with the ideas of (multi-parameter) mechanism-design.

In conclusion, it bears emphasizing that the overarching purpose of the present work is to aid in applying the tools of MARL to electricity market design, and ultimately to the acceleration of the energy transition. The leitmotif of outer-loop optimization in the face of inner-loop strategic behavior plays out at variegated scales throughout life - as for electricity market designers, so too for legislators, AI-aligners, and chess-players. The author hopes that his small contribution may advance both the knowledge of the field and, to however small a degree, the order and well-being of the world.

References

[1] Lawrence M. Ausubel and Peter Cramton. “Using forward markets to improve electricity market design”. en. In: *Utilities Policy* 18.4 (Dec. 2010), pp. 195–200. ISSN: 09571787. DOI: 10.1016/j.jup.2010.05.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957178710000366 (visited on 02/07/2023).

[2] Wendelin Boehmer, Vitaly Kurin, and Shimon Whiteson. “Deep Coordination Graphs”. en. In: *Proceedings of the 37th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, Nov. 2020, pp. 980–991. URL: https://proceedings.mlr.press/v119/boehmer20a.html (visited on 07/18/2023).

[3] *Brouwer theorem - Encyclopedia of Mathematics*. URL: https://encyclopediaofmath.org/index.php?title=Brouwer_theorem (visited on 07/18/2023).

[4] Lucian Busoniu, Robert Babuska, and Bart De Schutter. “A Comprehensive Survey of Multiagent Reinforcement Learning”. en. In: *IEEE Trans. Syst., Man, Cybern. C* 38.2 (Mar. 2008), pp. 156–172. ISSN: 1094-6977, 1558-2442. DOI: 10.1109/TSMCC.2007.913919. URL: https://ieeexplore.ieee.org/document/4445757/ (visited on 12/22/2022).

[5] Nicolò Cesa-Bianchi et al. “Boltzmann Exploration Done Right”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/b299ad862b6f12cb57679f0538eca514-Abstract.html (visited on 07/18/2023).

[6] Peter Cramton. “Electricity market design”. en. In: *Oxford Review of Economic Policy* 33.4 (Nov. 2017), pp. 589–612. ISSN: 0266-903X, 1460-2121. DOI: 10.1093/oxrep/grx041. URL: http://academic.oup.com/oxrep/article/33/4/589/4587939 (visited on 02/07/2023).

[7] Yan Du et al. “Approximating Nash Equilibrium in Day-ahead Electricity Market Bidding with Multi-agent Deep Reinforcement Learning”. en. In: *Journal of Modern Power Systems and Clean Energy* 9.3 (2021), pp. 534–544. ISSN: 2196-5625. DOI: 10.35833/MPCE.2020.000502. URL: https://ieeexplore.ieee.org/document/9406572 (visited on 02/07/2023).

[8] Ido Erev et al. “Learning and equilibrium as useful approximations: Accuracy of prediction on randomly selected constant sum games”. en. In: *Economic Theory* 33.1 (July 2007), pp. 29–51. ISSN: 0938-2259, 1432-0479. DOI: 10.1007/s00199-007-0214-y. URL: http://link.springer.com/10.1007/s00199-007-0214-y (visited on 02/14/2023).

[9] Richard Everett. “Learning Against Non-Stationary Agents with Opponent Modelling & Deep Reinforcement Learning”. en. In: (), p. 8.

[10] Drew Fudenberg and David K. Levine. “Learning and Equilibrium”. en. In: *Annu. Rev. Econ.* 1.1 (Sept. 2009), pp. 385–420. ISSN: 1941-1383, 1941-1391. DOI: 10.1146/annurev.economics.050708.142930. URL: https://www.annualreviews.org/doi/10.1146/annurev.economics.050708.142930 (visited on 12/22/2022).

[11] Steven A. Gabriel et al. *Complementarity Modeling in Energy Markets*. en. Google-Books-ID: Lu1L5wUea8IC. Springer Science & Business Media, July 2012. ISBN: 978-1-4419-6123-5.

[12] Christoph Graf et al. “Computational Performance of Deep Reinforcement Learning to Find Nash Equilibria”. en. In: *Comput Econ* (Jan. 2023). ISSN: 1572-9974. DOI: 10.1007/s10614-022-10351-6. URL: https://doi.org/10.1007/s10614-022-10351-6 (visited on 04/21/2023).

[13] Sven Ove Hansson. “Risk”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Summer 2023. Metaphysics Research Lab, Stanford University, 2023. URL: https://plato.stanford.edu/archives/sum2023/entries/risk/ (visited on 07/18/2023).

[14] Lion Hirth. “The market value of variable renewables”. en. In: *Energy Economics* 38 (July 2013), pp. 218–236. ISSN: 01409883. DOI: 10.1016/j.eneco.2013.02.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0140988313000285 (visited on 02/14/2023).

[15] *Infra-Bayesianism - AI Alignment Forum*. en. URL: https://www.alignmentforum.org/s/CmrW8fCmSLK7E25sa (visited on 07/18/2023).

[16] Daniel S. Kirschen and Goran Strbac. *Fundamentals of Power System Economics*. New York, UNITED KINGDOM: John Wiley & Sons, Incorporated, 2004. ISBN: 978-0-470-02058-6. URL: http://ebookcentral.proquest.com/lib/delft/detail.action?docID=219775 (visited on 02/14/2023).

[17] Marc Lanctot et al. “A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/3323fe11e9595c09af38fe67567a9394-Abstract.html (visited on 06/03/2023).

[18] Ryan Lowe et al. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. en. arXiv:1706.02275 [cs]. Mar. 2020. URL: http://arxiv.org/abs/1706.02275 (visited on 11/23/2022).

[19] “Mechanism Design and the Revelation Principle”. en. In: ().

[20] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. en. SpringerBriefs in Intelligent Systems. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-28927-4 978-3-319-28929-8. DOI: 10.1007/978-3-319-28929-8. URL: http://link.springer.com/10.1007/978-3-319-28929-8 (visited on 11/29/2022).

[21] Georgios Papoudakis et al. *Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning*. en. arXiv:1906.04737 [cs, stat]. June 2019. URL: http://arxiv.org/abs/1906.04737 (visited on 11/23/2022).

[22] Julien Perolat et al. *Mastering the Game of Stratego with Model-Free Multiagent Reinforcement Learning*. en. arXiv:2206.15378 [cs]. June 2022. URL: http://arxiv.org/abs/2206.15378 (visited on 11/29/2022).

[23] Ksenia Poplavskaya, Jesus Lago, and Laurens de Vries. “Effect of market design on strategic bidding behavior: Model-based analysis of European electricity balancing markets”. en. In: *Applied Energy* 270 (July 2020), p. 115130. ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.115130. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261920306425 (visited on 11/23/2022).

[24] Tabish Rashid et al. “Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 10199–10210. URL: https://proceedings.neurips.cc/paper/2020/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html (visited on 07/18/2023).

[25] Tim Roughgarden. “CS364A: Algorithmic Game Theory Lecture #2: Mechanism Design Basics”. en. In: ().

[26] Yoav Shoham. “Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations”. en. In: ().

[27] Kyunghwan Son et al. “QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning”. en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, May 2019, pp. 5887–5896. URL: https://proceedings.mlr.press/v97/son19a.html (visited on 07/18/2023).

[28] Hugo Sonnenschein. “Market Excess Demand Functions”. In: *Econometrica* 40.3 (May 1972), p. 549. ISSN: 00129682. DOI: 10.2307/1913184. URL: https://www.jstor.org/stable/1913184?origin=crossref (visited on 07/18/2023).

[29] Katie Steele and H. Orri Stefánsson. “Decision Theory”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2020. Metaphysics Research Lab, Stanford University, 2020. URL: https://plato.stanford.edu/archives/win2020/entries/decision-theory/ (visited on 07/18/2023).

[30] Richard S Sutton et al. “Policy Gradient Methods for Reinforcement Learning with Function Approximation”. en. In: (), p. 7.

[31] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. en. Second edition. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018. ISBN: 978-0-262-03924-6.

[32] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". en. In: *Nature* 575.7782 (Nov. 2019). Number: 7782 Publisher: Nature Publishing Group, pp. 350–354. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1724-z. URL: https://www.nature.com/articles/s41586-019-1724-z (visited on 07/18/2023).

[33] Ermo Wei and Sean Luke. "Lenient Learning in Independent-Learner Stochastic Cooperative Games". en. In: ().

[34] Paul Weirich. "Causal Decision Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2020. Metaphysics Research Lab, Stanford University, 2020. URL: https://plato.stanford.edu/archives/win2020/entries/decision-causal/ (visited on 07/18/2023).

[35] Thomas Wolgast, Eric MSP Veith, and Astrid Nieße. "Towards reinforcement learning for vulnerability analysis in power-economic systems". en. In: *Energy Inform* 4.S3 (Sept. 2021), p. 21. ISSN: 2520-8942. DOI: 10.1186/s42162-021-00181-5. URL: https://energyinformatics.springeropen.com/articles/10.1186/s42162-021-00181-5 (visited on 12/02/2022).

[36] Yujian Ye et al. "Deep Reinforcement Learning for Strategic Bidding in Electricity Markets". en. In: *IEEE Trans. Smart Grid* 11.2 (Mar. 2020), pp. 1343–1355. ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2019.2936142. URL: https://ieeexplore.ieee.org/document/8805177/ (visited on 02/07/2023).

[37] Ziqing Zhu et al. "Reinforcement learning in deregulated energy market: A comprehensive review". en. In: *Applied Energy* 329 (Jan. 2023), p. 120212. ISSN: 03062619. DOI: 10.1016/j.apenergy.2022.120212. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261922014696 (visited on 12/01/2022).