

---

# MULTI-AGENT DEEP DETERMINISTIC POLICY GRADIENT ALGORITHM FOR PEER-TO-PEER ENERGY TRADING CONSIDERING DISTRIBUTION NETWORK CONSTRAINTS

---

A PREPRINT

**Cephas Samende\***  
Keele University  
United Kingdom

**Jun Cao**  
Keele University  
United Kingdom

**Zhong Fan**  
Keele University  
United Kingdom

## ABSTRACT

In this paper, we investigate an energy cost minimization problem for prosumers participating in peer-to-peer energy trading. Due to (i) uncertainties caused by renewable energy generation and consumption, (ii) difficulties in developing an accurate and efficient energy trading model, and (iii) the need to satisfy distribution network constraints, it is challenging for prosumers to obtain optimal energy trading decisions that minimize their individual energy costs. To address the challenge, we first formulate the above problem as a Markov decision process and propose a multi-agent deep deterministic policy gradient algorithm to learn optimal energy trading decisions. To satisfy the distribution network constraints, we propose distribution network tariffs which we incorporate in the algorithm as incentives to incentivize energy trading decisions that help to satisfy the constraints and penalize the decisions that violate them. The proposed algorithm is model-free and allows the agents to learn the optimal energy trading decisions without having prior information about other agents in the network. Simulation results based on real-world datasets show the effectiveness and robustness of the proposed algorithm.

**Keywords** Multi-agent · deep deterministic policy gradient · peer-to-peer energy trading · renewable generation · Markov decision process.

## 1 Introduction

Peer-to-peer (P2P) energy trading is a promising approach for addressing the world's 'energy trilemma' (i.e. environmental sustainability, energy equity, and energy security) facing human society today [1]. Its emergence is as a result of rapid deployment and connectivity of distributed energy resources (DERs) to the power system [2]. Conventionally, power systems were dominated by centralized generators situated in strategic locations [3]. The generated power was transmitted over long distances to consumers for consumption. As the result, power flow was unidirectional (flowing from generators to consumers) and control of the power flow was easy due to centralized structures (i.e. generation, transmission, distribution and consumption) [3].

With digitization and the emergence of DERs such as battery energy storage systems, rooftop solar photovoltaic (PV) installations and smart home appliances, power systems are no longer passive but active with DERs actively involved in the electricity system [2, 4]. As DERs can generate energy at point of consumption, power flow in today's power system is bidirectional, posing significant challenges in terms of planning, operation, control and protection of the power system [2, 5]. At the same time, the emergence of DERs along with digitization have created new opportunities which can be used to solve most of the challenges caused by DERs through development of local energy markets such as P2P energy trading schemes [6].

With P2P energy trading, customers with DERs (called 'prosumers' as they are able to generate and consume energy) can locally trade and share energy with each other. As many DERs are stochastic in nature, any surplus energy can be

---

\*All the authors are with the School of Computing and Mathematics, Keele University

sold to neighbouring prosumers with deficit energy via a cloud-based P2P platform. The main role of the P2P platform is to set the P2P energy selling and buying price.

To encourage prosumer participation in P2P energy trading, the selling and buying price must be higher and lower than the export and import prices imposed by the energy service provider (ESP) respectively [7]. Thus, prosumers acting as energy producers benefit more from individual profits and prosumers acting as energy consumers benefit more from cheap energy when they trade with each other via the P2P platform than when they trade directly with the ESP. This creates a win-win situation among the prosumers thereby encouraging adoption and investments in DERs [8]. At the same time, local energy sharing through P2P energy trading reduces peak demand on the main grid thereby reducing investments and operational costs [9].

Although P2P energy trading has many advantages and is the promising next generation energy management technique for smart grids, the following challenges must be addressed. Firstly, it is generally intractable to develop a P2P energy trading model that is accurate and efficient enough for optimal energy trading decision making [10]. Secondly, it is hard to implement P2P energy trading at a large-scale and in real time when conventional and model-based optimization techniques are used [11–13]. Thirdly, P2P energy trading has so many uncertainties caused by the stochastic nature of renewable generation, power consumption and electricity price [14, 15]. Fourthly, as the distribution network acts as a medium for energy exchange during energy trading, its own hard technical constraints including voltage limits and power balance constraints must be satisfied [2]. Finally, as prosumers do not have access to information about others, it is difficult to make optimal energy trading decisions. Many of the existing methods e.g. in [16–20] are model-based approaches which require domain expert knowledge to model P2P energy trading, making them difficult to apply.

To address the above challenges, many studies are focusing on the use of deep reinforcement learning (DRL), which is an artificial intelligence framework with proven success in playing Atari and Go games [21]. DRL is a combination of deep learning and reinforcement learning [21, 22]. Compared to model-based methods, DRL-based methods have the following advantages: (i) they are model-free and the agents learn optimal energy trading policies by interacting with the energy trading environment [23, 24]. Thus, they can operate without explicit knowledge and rigorous mathematical models of the environment, (ii) they have self-adaptability and can operate in an on-line way without requiring forecast information about the energy trading environment [25, 26], and (iii) they are data-driven and capable of determining optimal control actions in real-time even in complex energy trading environments [14].

In [12, 23], Chen *et al.* proposed a DRL-based algorithm to maximize trading profits while also minimizing the dependence on the power plant. In [24] Kim and Lee proposed a DRL-based automatic trading algorithm originally designed for stock trading to maximize profits of prosumers participating in P2P energy trading. In [11] Lu *et al.* presented a DRL-based microgrid energy trading scheme to determine optimal energy trading policy according to predicted energy generation, power consumption and battery energy level. In [14], Gao *et al.* proposed a multi-agent DRL-based approach for minimizing energy costs for P2P energy trading prosumers in a microgrid. Although some DRL-based methods have been proposed in above-mentioned studies, none of them considers the distribution network constraints and loss. As the distribution network acts a medium of exchange for the traded energy, failing to consider its underlying electrical network constraints may cause the outcome of the studied DRL-based energy trading schemes to be impractical.

In this paper, we propose a multi-agent DRL algorithm for determining optimal energy trading policies that minimize energy costs while satisfying distribution network constraints. Each prosumer is modelled as an agent. The energy trading environment is modelled as a multi-agent environment where an action of one agent affects the actions of others, making the entire energy trading environment to be non-stationary from an agent's perspective. As most DRL-based methods such as deep Q-networks [21] perform poorly in multi-agent settings because they do not use information of other agents during training, we adopt a multi-agent deep deterministic gradient policy (MADDPG) [27] based framework to design the proposed algorithm. With the proposed MADDPG-based algorithm, training is centralized with each agent using states and actions of other agents. This makes the environment to be stationary during training even as the agent actions change. Meanwhile execution is decentralized with each agent using only local information to make actions without knowing others' information. The main contributions of this paper are summarized as follows:

- Propose a local P2P energy trading market which enables a distribution system operator (DSO) to leverage prosumers' battery energy storage system as a flexible asset to satisfy the distribution network constraints.
- Propose a MADDPG-based algorithm to learn optimal energy trading policies for each prosumer to minimize the energy costs while satisfying the distribution network constraints.
- Design actor and critic networks for each agent to ensure that training of the agents is stable and that the output from the actor network is optimal.
- Introduce a novel strategy using distribution network tariffs (DNT) to incentivize the prosumers to provide the flexibility required to satisfy the network constraints.

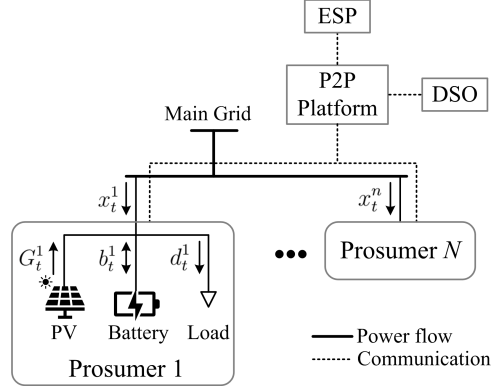


Figure 1: Distribution network model with P2P platform.

The rest of the paper is organised as follows. Section 2 presents the proposed P2P energy trading model and the energy cost minimization problem considered in this paper. Section 3 presents the proposed algorithm. Simulation results that verify the effectiveness of the proposed algorithm are given in Section 4. Section 5 concludes the paper.

## 2 System Model

Fig. 1 shows a simplified distribution network and P2P energy trading model studied in this paper. The distribution network can be described as a connected graph,  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  where  $\mathcal{N} = \{0, 1, \dots, N\}$  is a set of buses and  $\mathcal{E}$  is a set of distribution lines. Thus, each bus,  $n \in \mathcal{N}$  corresponds to a prosumer (e.g. a household, business or commercial building) connected to the distribution network. Bus  $n = 0$  represents a substation bus, which links the distribution network to the main grid, and no consumer or prosumer is connected to it. For easy of reference, we use the term prosumer to refer to either a consumer or prosumer in the remaining part of the paper, unless explicitly stated. It should be noted that not all prosumers connected to the distribution network are willing to participate in the P2P energy trading. We denote the set of prosumers participating in P2P energy trading as  $\mathcal{P} = \{1, \dots, P\}$ . Each prosumer,  $p \in \mathcal{P}$  consists of a solar PV system, battery and/or load. In addition, each prosumer is equipped with an energy management system (EMS) to: (i) collect and send to the P2P platform data such as PV generation, energy consumption, battery charge and discharge power, (ii) receive the price signal from the P2P platform and (iii) optimally schedule the battery energy storage system.

As energy buying and selling prices via the P2P platform must be lower and higher than the import and export tariffs set by the ESP respectively [7], the P2P platform communicates with the ESP in achieving this. To avoid overloading the distribution network, the energy sharing between the prosumers must satisfy distribution network constraints such as voltage limits and losses. In many countries including the UK, management and operation of the distribution network is a responsibility of the DSO, who operates independently from the P2P platform [2, 28]. To satisfy the network constraints at every transaction, the intended power exchange between the prosumers must be approved by the DSO either through penalties or monetary incentives [17, 18]. Bi-directional communication links are required for communication between the prosumer EMS, P2P platform, DSO and the ESP. Further, we assume that the prosumer EMS, P2P platform, ESP and DSO operate on a common time horizon,  $\mathcal{T} = \{1, \dots, T\}$  with equal time slots,  $\Delta t$ .

### 2.1 Prosumer Model

The PV generation profile for prosumer  $p$  during the operation horizon  $\mathcal{T}$  is defined as follows

$$\mathbf{G}^p = \{G_1^p, G_2^p, \dots, G_T^p\}, \quad p \in \mathcal{P} \quad (1)$$

We assume that the PV is operated in maximum power point tracking (MPPT) mode [29] and thus,  $\mathbf{G}^p$  is maximum power output.

The total power consumption profile of prosumer  $p$  during time horizon  $\mathcal{T}$  can be defined as follows

$$\mathbf{D}^p = \{d_1^p, d_2^p, \dots, d_T^p\}, \quad p \in \mathcal{P} \quad (2)$$

We consider loads that do not have a certain amount of flexibility. Consideration of flexible loads is beyond the scope of this paper and considered as future work.

For prosumer  $p \in \mathcal{P}$ , let  $SoC_t^p$  be the battery state of charge (SoC), which indicates the amount of energy remaining in the battery after a charge or discharge operation. Let  $b_t^p$  be the battery power output (positive  $b_t^p$  to denote discharging and negative  $b_t^p$  to denote charging),  $\eta_t^p$  be the battery charge or discharge efficiency and let  $E_b^p$  be the battery energy capacity. The dynamics of the SoC can be modelled as follows [25]

$$SoC_{t+\Delta t}^p = SoC_t^p - \frac{\eta_t^p b_t^p \Delta t}{E_b^p}, \quad p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (3)$$

It should be noted that the value of  $\eta_t^p$  is calculated differently based on whether the battery is charging or discharging [25]. To prolong the battery lifetime,  $b_t^p$  must be restricted within a certain range as follows

$$\frac{E_b^p (SoC_t^p - SoC_{max}^p)}{\eta_t^p \Delta t} \leq b_t^p \leq \frac{E_b^p (SoC_t^p - SoC_{min}^p)}{\eta_t^p \Delta t} \quad (4)$$

$$p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (5)$$

where  $SoC_{min}^p$  and  $SoC_{max}^p$  are predetermined SoC limits to indicate a fully discharged and charged battery respectively.

Also,  $b_t^p$  must satisfy the power limits of the inverter to which the battery is connected as follows

$$b_{min}^p \leq b_t^p \leq b_{max}^p, \quad p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (6)$$

where  $b_{min}^p$  and  $b_{max}^p$  are minimum and maximum inverter power limits respectively.

Practically, the lifetime of the battery is shorter than any other asset in the distribution network. Thus, its wear cost has great impact on the economics of the energy trading strategies of the prosumers. The empirical wear cost  $\varpi^p$  of the battery can be expressed as [30]

$$\varpi^p = \frac{C_b^p}{ACC \times 2 \times DoD \times E_b^p \times \mu_b^2} \quad (7)$$

where  $C_b^p$  is battery price per kWh,  $DoD$  is depth of discharge at which the battery is cycled,  $\mu_b$  is round trip efficiency and  $ACC$  is life cycle at a specific  $DoD$ .  $ACC$  is multiplied by two as one cycle consists of charge and discharge phases.

As profiles of power generation and consumption are different from each other, prosumer  $p$  can assume the role of an energy buyer or seller at any time  $t \in \mathcal{T}$  based on the net power  $x_t^p$  which is defined as follows

$$x_t^p = d_t^p - (G_t^p + b_t^p), \quad p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (8)$$

That is, if  $x_t^p \geq 0$ , the prosumer is an energy buyer, buying energy from others or the grid to meet its power deficit. The prosumer is an energy seller if  $x_t^p < 0$ , selling the excess energy to others or the grid.

## 2.2 P2P Pricing Mechanism

Energy buying and selling all happens through the P2P platform as shown in Fig. 1. To set the energy buying/selling price in the platform, we adopt the supply-to-demand ratio (SDR) based pricing mechanism [7, 10, 31], mainly for two reasons: (i) it is simple in principle and easy to obtain, and can be updated in real time and (ii) it satisfies the basics of modern economics, i.e., price is inversely proportional to SDR as demonstrated in the following paragraphs.

SDR is defined as the ratio of the total power supply to the total demand in the energy sharing community, i.e.,

$$SDR^t = \frac{\sum_{p \in \mathcal{P}} (G_t^p + b_t^p)}{\sum_{p \in \mathcal{P}} d_t^p}, \quad t \in \mathcal{T} \quad (9)$$

The SDR varies with time because of the volatility of solar generation and power consumption. This means that the energy buying/selling price is also not constant but fluctuating according to the SDR. Let the ESP's import and export prices as received by the P2P platform be  $\lambda_b^t$  and  $\lambda_s^t$  respectively, where  $\lambda_b^t \geq \lambda_s^t$ . Let the prosumer's buying and selling prices through the P2P platform be  $\pi_b^t$  and  $\pi_s^t$  respectively, where  $\pi_b^t \leq \lambda_b^t$  and  $\pi_s^t \geq \lambda_s^t$ . The buying/selling price vector set by the P2P platform can be defined as follows

$$\pi = \{\pi_b^1, \pi_b^2, \dots, \pi_b^T : \pi_s^1, \pi_s^2, \dots, \pi_s^T\} \quad (10)$$

The  $\pi_b^t$  and  $\pi_s^t$  can be obtained as a function of  $SDR^t$ ,  $\lambda_b^t$  and  $\lambda_s^t$  as follows [7]

$$\pi_s^t = \begin{cases} \frac{(\lambda_s^t + \lambda)\lambda_b^t}{(\lambda_b^t - \lambda_s^t - \lambda)SDR^t + \lambda_s^t + \lambda}, & 0 \leq SDR^t \leq 1 \\ \lambda_s^t + \frac{\lambda}{SDR^t} & SDR^t > 1 \end{cases} \quad (11)$$

$$\pi_b^t = \begin{cases} \pi_s^t SDR^t + \lambda_b^t (1 - SDR^t), & 0 \leq SDR^t \leq 1 \\ \lambda_s^t + \lambda & SDR^t > 1 \end{cases} \quad (12)$$

where  $\{\lambda | 0 \leq \lambda \leq (\lambda_b^t - \lambda_s^t)\}$  is a compensation price which is used to incentivize prosumers to continue participating in P2P energy trading when  $SDR^t > 1$  (the situation which happens when prosumers have more power supply than demand). Without the compensation price, buying price would be equal to selling price when  $SDR^t > 1$ , a situation that would favour prosumers who are buyers and not sellers. This may discourage the sellers from participating in P2P energy trading especially during periods of high PV generation.

The  $SDR^t$  given by (9) mainly depends on the adjusted power consumption and battery charge and discharge power from all the prosumers. This means that  $\pi_s^t$  and  $\pi_b^t$  largely depend on the choice of  $b_t^p$ . Decreasing  $b_t^p$  (when charging the battery) drives  $SDR^t$  towards zero and  $\pi_s^t$  or  $\pi_b^t$  towards  $\lambda_b^t$ . Conversely, increasing  $b_t^p$  (when discharging the battery) drives  $SDR^t$  towards 1 and  $\pi_s^t$  or  $\pi_b^t$  towards  $\lambda_s^t$ . Thus, price is inversely proportional to the SDR. In both cases the following relationship is satisfied

$$\begin{cases} \pi_b^t \leq \lambda_b^t \\ \pi_s^t \geq \lambda_s^t \end{cases} \quad (13)$$

That is, prosumers are better off buying and selling their energy via the P2P platform because of lower buying prices and higher selling prices compared to the prices,  $\lambda_b^t$  and  $\lambda_s^t$  offered by the ESP.

### 2.3 Distribution Network Tariffs

The selling and buying price given by (11) and (12) respectively do not take distribution network constraints such as line congestion and voltage limits into account. If not controlled, the net power obtained for each energy trading transaction may overload the distribution network. We introduce the use of a distribution network tariff (DNT) to incentivize transactions that do not violate the distribution network constraints and penalize those that violate the constraints.

Prosumer's contribution towards violation of network constraints depends on its location on the distribution network and operational time [32]. Thus, we derive the DNTs from distribution locational marginal pricing (DLMP), a temporal-spatio pricing mechanism which exposes prosumers to the true cost of energy delivery in the distribution network [32, 33]. The DLMP can be decomposed into four constituent components; marginal cost of energy demand, marginal cost of network loss, marginal cost of congestion and marginal cost of bus voltage [18, 32]. As the energy buying/selling price (10) is set by the P2P platform, the proposed DNTs are determined from the other three components of the DLMP, i.e. the marginal cost of network losses, marginal cost of congestion and marginal cost of voltage. Calculation of the DLMP is detailed in [32].

Let the DLMP obtained at time  $t$ ,  $t \in \mathcal{T}$  for the substation bus (i.e.,  $n = 0$ ,  $n \in \mathcal{N}$ ) be  $\lambda_0^t$  and the DLMP for prosumer  $p$  be  $\lambda_p^t$ . The DNT  $\delta_p^t$  for prosumer  $p$  can be calculated as follows

$$\delta_p^t = \lambda_p^t - \lambda_0^t, \quad p \in \mathcal{P}, \quad t \in \mathcal{T} \quad (14)$$

As  $\lambda_0^t$  only accounts for the marginal cost of energy delivery at the substation bus (which is the same at every bus  $n \in \mathcal{N}$ ), subtracting it from  $\lambda_p^t$  gives  $\delta_p^t$ , which is the sum of marginal cost of line losses, congestion and voltage.  $\delta_p^t$  is zero when the net power injected by prosumer  $p$  at  $t \in \mathcal{T}$  does not cause network losses, congestion and/or violate voltage limits. Otherwise,  $\delta_p^t$  is not equal zero due to either network loss, congestion and/or voltage limit violation.

As  $\delta_p^t$  reflects the condition of the entire distribution network considering both location and time, it is therefore a suitable tariff to manage the distribution network constraints.  $\delta_p^t$  can be used as an incentive to incentivize a prosumer whose net power transfer helps to satisfy the network constraints and penalize the one whose power transfer violates the constraints.

### 2.4 Problem Formulation

Each prosumer  $p$  that can schedule the operation of its energy assets can be considered to be an agent. Thus, P2P energy trading can be described as a multi-agent system. Each agent's energy trading decision at a given time slot,

$t$  depends on the current information it receives from its assets (e.g. battery energy level) and the P2P platform (e.g. energy buying/selling price and DNT), and not on the prior history. Thus, energy trading and the subsequent scheduling of the energy assets can be formulated as a Markov decision process (MDP) [34] with continuous action spaces (i.e. assuming that operation of the assets e.g., battery is continuous).

Let the set of agents be the same as that of prosumers (i.e.  $p \in \mathcal{P}$ ). The MDP for each agent  $p$  proceeds as follows: Given a local agent state  $s_p^t \in \mathcal{S}$  at time slot  $t$ , where  $s_p^t = (G_t^p, d_t^p, SoC_t^p)$ , the agent selects an action  $a_p^t \in \mathcal{A}$ , where  $a_p^t = (b_t^p)$  based on a stochastic policy,  $\pi_{\theta_p}$ . The taken action takes the agent into a next local state  $s_p^{t'} \in \mathcal{S}$  according to a state transition probability function,  $\mathcal{F}$ . At the end of the time slot, the agent receives a reward,  $r_p^t$  as a function of the current state and action as follows

$$r_p^t = - \sum_{t \in \mathcal{T}} [(\pi + \delta_p^t) x_t^p + \varpi^p |b_t^p|] \Delta t \quad (15)$$

$$\pi = \begin{cases} \pi_b^t, & \text{if } x_t^p \geq 0 \\ \pi_s^t, & \text{Otherwise} \end{cases}$$

where the first term is the cost for both purchasing energy in the P2P platform and using the distribution network. The second term is the cost of using the battery.

It is important to note that  $\delta_p^t$  is equal to zero when the net power  $x_t^p$  does not contribute to network losses, congestion and/or voltage limit violations, otherwise  $\delta_p^t$  is non-zero. Furthermore, through  $\pi$  and  $\delta_p^t$ , the reward is a function of all agent states and actions in the network (i.e. to determine  $\pi$  and  $\delta_p^t$ , the P2P platform and the DSO needs access to all the states and actions of all agents in the network). In other words, an action of one agent affects the rewards of all other agents in the system.

The goal of each agent is to maximize its own expected reward  $R_p = \sum_{t=0}^T \gamma^t r_p^t$  where  $\gamma$  is a discount factor. As market prices, generation and demand are volatile in nature, it is generally impossible to obtain with certainty the state transition probability function  $\mathcal{F}$  required to derive an optimal policy  $\pi_{\theta_p}$  needed to maximize  $R_p$ . To circumvent this difficulty, we propose to use an artificial intelligence-based approach which is data-driven and model-free as discussed in Section 3.

### 3 Proposed Learning Algorithm

#### 3.1 Deep Reinforcement Learning

Reinforcement learning (RL) is the process in which agents learn for themselves through trial and error [22] the optimal policy  $\pi_{\theta_p}$  to achieve optimal actions that maximize the cumulative reward  $R_p$ . Like a human, agents need to construct and learn their own knowledge directly from raw data such as a historic solar PV generation, demand and market prices. This can be achieved by DRL. DRL has given rise to several value-based algorithms such as Deep Q-networks (DQN) [21, 25] and policy-based algorithms such as deep deterministic policy gradient (DDPG) [35, 36].

As each agent's reward as given by (15) depends on actions from other agents, the interaction between the agents during energy trading can be described as a mixed cooperative-competitive. Naive application of DQN and policy gradient algorithms to such multi-agent settings performs poorly because they do not use information of other agents during training. We propose to use an MADDPG-based algorithm which overcomes this difficulty by using states and actions of other agents during training.

#### 3.2 Multi-Agent Deep Deterministic Policy Gradient Algorithm

Fig. 2 shows the architecture and workflow of the proposed MADDPG algorithm. Each agent is modelled as a DDPG agent, where, however, states and actions are shared between the agents during training. In particular, each agent consists of two networks: an actor network and a critic network. Both actor and the critic networks are created from dense layers with hidden layers having ReLU activations. An actor-network maps the local state of an agent to optimal actions using a Tanh activation function in the output layer. A critic network evaluates the actions received from the actor network to improve the performance of the actor network. The output layer of the critic network is activated by a linear function.

During training, the actor network uses only the local state to calculate the actions while the critic network uses states and actions of all agents in the system in evaluating the local action. As actions of all agents are known by each agent's critic network, the entire environment is stationary during training. During execution, critic networks are removed and

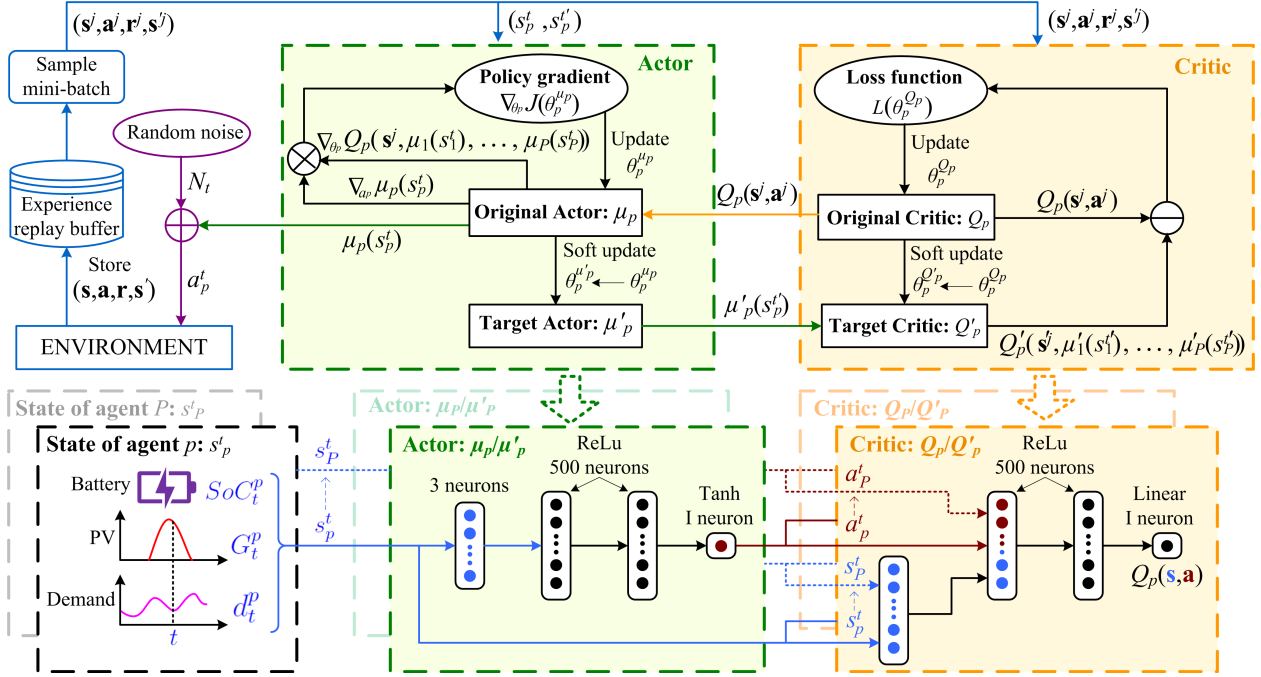


Figure 2: Architecture and workflow of the proposed MADDPG algorithm. Each agent  $p, p = 1, 2, \dots, P$  consists of an original actor network  $\mu_p$  (and target actor network  $\mu'_p$ ) and original critic network  $Q_p$  (and target critic network  $Q'_p$ ).

only actor networks are used. This means that with MADDPG, training is centralized while execution is decentralized. As DNTs are obtained independently by the DSO, we leverage the centralized training provision of the MADDPG to incorporate the DNTs in the algorithm during training.

### 3.3 Proposed Algorithm

The details of the proposed MADDPG algorithm which are illustrated in Fig. 2 are given by Algorithm 1. Let the actor and critic network of agent  $p$  be denoted as  $\mu_p$  and  $Q_p$ , and the associated network weights as  $\theta_p^{\mu_p}$  and  $\theta_p^{Q_p}$  respectively. Before training starts,  $\mu_p$  and  $Q_p$  (which we refer to as original networks) are created and their weights  $\theta_p^{\mu_p}$  and  $\theta_p^{Q_p}$  are randomly initialized. To add stability to the training, target actor  $\mu'_p$  and target critic  $Q'_p$  networks which are identical to the original networks  $\mu_p$  and  $Q_p$  are also created and their weights are initialized as  $\theta_p^{\mu'_p} \leftarrow \theta_p^{\mu_p}$  and  $\theta_p^{Q'_p} \leftarrow \theta_p^{Q_p}$ .

For each agent  $p$ , a replay buffer  $\mathcal{D}$  is created and initialized to store list of tuples  $(s, \mathbf{a}, \mathbf{r}, s')$  known as experiences, where  $s = (s_1^t, \dots, s_P^t)$ ,  $\mathbf{a} = (a_1^t, \dots, a_P^t)$ ,  $\mathbf{r} = (r_1^t, \dots, r_P^t)$  and  $s' = (s_1^{t'}, \dots, s_P^{t'})$ . The replay buffer adds stability to the training as agents learn by sampling mini-batches from all of the accumulated experiences during training.

For each training episode, a random process for action exploration and an initial state  $s$  are initialized. We use Ornstein-Uhlenbeck process [37] for generating the noise  $\mathcal{N}_t$  for action exploration. With the received state  $s_p^t, s_p^t \in s$ , and noise  $\mathcal{N}_t$ , each agent makes an action given by

$$a_p^t = \mu_p(s_p^t) + \mathcal{N}_t \quad (16)$$

where  $\mu_p(s_p^t)$  is output (action) of the actor network  $\mu_p$ .

The actions from the agents together with their states at time slot,  $t$  are used to simulate the energy trading mechanism including the calculation of selling/buying price and the DNTs. At the end of the time slot, each agent calculates its own reward  $r_p^t, r_p^t \in \mathbf{r}$  given by (15) and observes a new state  $s_p^{t'}, s_p^{t'} \in s'$ . The experience  $(s, \mathbf{a}, \mathbf{r}, s')$  is stored in the replay buffer  $\mathcal{D}$  and the initial state  $s$  gets updated;  $s \leftarrow s'$ .

For each agent  $p$ , the actor  $\mu_p$  and critic  $Q_p$  networks are trained by (random) sampling  $S$  number of transitions from the replay buffer  $\mathcal{D}$ . The transitions are used to update the network weights for both original and target actor and critic networks. Let  $(\mathbf{s}^j, \mathbf{a}^j, \mathbf{r}^j, \mathbf{s}'^j)$  be an experience for each transition  $j$ ,  $j \in S$ . Each agent  $p$  updates the weights of its original critic network (i.e.  $\theta_p^{Q_p}$ ) by minimizing the loss

$$L(\theta_p^{Q_p}) = \frac{1}{S} \sum_{j=1}^S (y_p^j - Q_p(\mathbf{s}^j, \mathbf{a}^j))^2 \quad (17)$$

where  $Q_p(\mathbf{s}^j, \mathbf{a}^j)$  is the predicted output of the original critic network and  $y_p^j$  is its target value which is given by

$$y_p^j = r_p^{t^j} + \gamma Q'_p(\mathbf{s}'^j, a_1^{t'}, \dots, a_P^{t'}) \Big|_{a_p^{t'} = \mu'_p(s_p^{t'})}, p \in \mathcal{P} \quad (18)$$

where  $a_p^{t'} = \mu'_p(s_p^{t'})$  is the predicted action by the target actor network and  $Q'_p(\mathbf{s}'^j, a_1^{t'}, \dots, a_P^{t'})$  is the predicted value by the target critic network.

Weights for the original actor network (i.e.  $\theta_p^{\mu_p}$ ) are updated using sampled policy gradient

$$\nabla_{\theta_p^{\mu_p}} J(\theta_p^{\mu_p}) = \nabla_{\theta_p^{\mu_p}} \mu_p(s_p^t) \nabla_{a_p^t} Q_p(\mathbf{s}^j, a) \quad (19)$$

where  $a = (\mu_1(s_1^t), \dots, \mu_P(s_P^t))$ .

Weights for both target actor and critic network (i.e.  $\theta_p^{\mu'_p}$  and  $\theta_p^{Q'_p}$ ) are updated as follows

$$\begin{cases} \theta_p^{Q'_p} \leftarrow \tau \theta_p^{Q_p} + (1 - \tau) \theta_p^{Q'_p} \\ \theta_p^{\mu'_p} \leftarrow \tau \theta_p^{\mu_p} + (1 - \tau) \theta_p^{\mu'_p} \end{cases} \quad (20)$$

where  $\tau$  is the learning rate.

After training, the trained critic network and the replay buffer are removed from each agent. Let the trained actor network for agent  $p$  be  $\mu_p^*$ . At every time slot  $t$ , each agent  $p$  only requires to make an observation of the local state  $s_p^t$  to obtain optimal actions  $a_p^t = \mu_p^*(s_p^t)$ . The obtained actions are considered to be optimal for energy trading and for satisfying the distribution network constraints.

---

**Algorithm 1** MADDPG Algorithm for P2P Energy Trading.

- 1: Randomly initialize (original) actor and critic networks
  - 2: Initialize (target) actor and critic networks
  - 3: Initialize replay buffer  $\mathcal{D}$
  - 4: **for** episode = 1 to  $M$  **do**
  - 5:   Initialize a random process  $\mathcal{N}_t$  for action exploration
  - 6:   Observe initial state  $\mathbf{s}$
  - 7:   **for**  $t = 1$  to  $T$  **do**
  - 8:     For each agent  $p$ , make an action according to (16)
  - 9:     Execute the actions  $\mathbf{a}$ , calculate the reward  $\mathbf{r}$  using (15) and observe next states  $\mathbf{s}'$
  - 10:    Store  $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$  in  $\mathcal{D}$
  - 11:    Update  $\mathbf{s} \leftarrow \mathbf{s}'$
  - 12:    **for** agent  $p = 1$  to  $P$  **do**
  - 13:     Randomly sample  $S$  from  $\mathcal{D}$
  - 14:     Update (original) critic network by minimizing (17)
  - 15:     Update (original) actor network using policy gradient (19)
  - 16:     Update (target) actor and critic network by (20)
  - 17:    **end for**
  - 18: **end for**
  - 19: **end for**
-



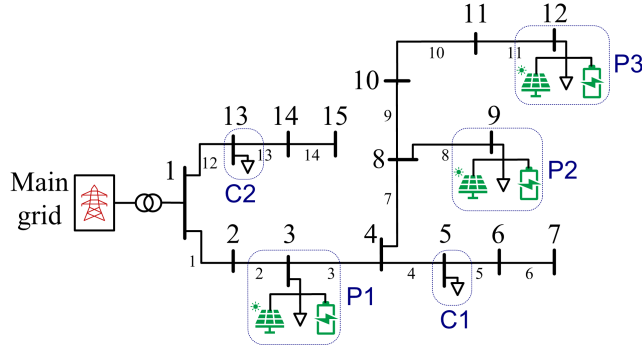


Figure 3: A low voltage 15-bus radial distribution network with C1 and C2 as consumers and P1, P2 and P3 as prosumers participating in a P2P energy trading scheme.

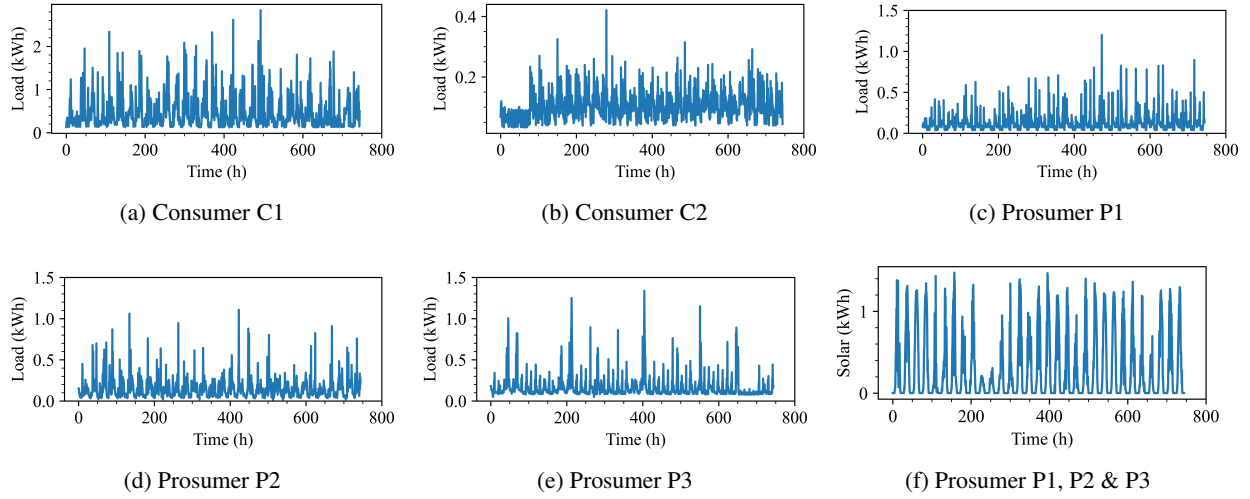


Figure 4: Half-hourly load and solar power profiles for the consumers and prosumers participating in P2P energy trading.

## 4 Case Study

### 4.1 Simulation Parameters

Fig. 3 shows a low voltage 15-bus radial distribution network [32] which is used to demonstrate the effectiveness of the proposed algorithm for reducing energy costs while satisfying network constraints. The distribution network parameters are given in [32]. Two consumers (denoted as C1 and C2) and three prosumers (denoted as P1, P2, and P3) are considered to participate in the P2P energy trading scheme. These have time-varying load and solar power profiles (with 30 minutes resolution) instead of fixed ones as shown in Fig. 4. The load and solar power profiles are for one month (744 hours) and they are obtained from UK's customer led network revolution (CLNR)<sup>2</sup> and UK power networks (UKPN)<sup>3</sup> respectively. The ESP's import and export prices are set to be  $\lambda_b^t = 0.05$  £/kWh and  $\lambda_s^t = 0.03$  £/kWh respectively.

The prosumers P1, P2 and P3 all have 1 kW of installed solar capacity and are exposed to equal amounts of solar irradiance. Each prosumer has battery parameters<sup>4</sup> as given in Table 1.

<sup>2</sup><http://www.networkrevolution.co.uk/project-library/dataset-tc1a-basic-profiling-domestic-smart-meter-customers/>

<sup>3</sup><https://data.london.gov.uk/dataset/photovoltaic-pv-solar-panel-energy-generation-data>

<sup>4</sup><https://www.tesla.com/support/energy/powerwall/documents/documents/>

Table 1: Battery Parameters.

Parameter	Value/Description
Battery type	Tesla Powerwall
Life cycle	5000
Initial SoC	50%
Usable capacity	13.5 kWh
Depth of discharge	100%
Price per kWh (£/kWh)	314.64
Round trip efficiency	92.5%

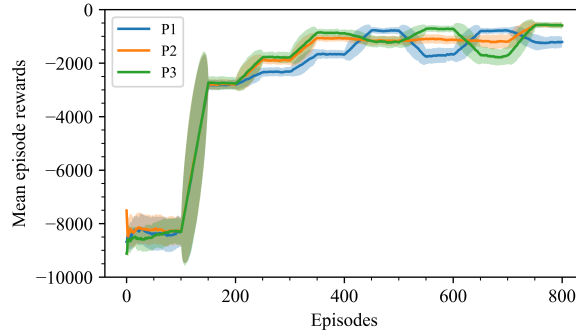


Figure 5: Mean episode rewards of the agents during the training process.

The actor and critic networks for each prosumer agent are designed using hyper-parameters tabulated in Table 2. Algorithm 1 is developed and implemented in Python using PyTorch framework [38]. An OpenAI Gym environment [39] is designed to model the multi-agent energy trading environment.

Table 2: Hyper-parameters for each Actor and Critic Network.

Hyper-parameter	Actor Network	Critic Network
Optimizer	Adam	Adam
Batch size	256	256
Discount factor	0.95	0.95
Learning rate	$1 \times 10^{-4}$	$3 \times 10^{-4}$
No. of hidden layers	2	2
No. of nodes in each layer	500	500

## 4.2 Performance Analysis Without Network Constraints

In this section, convergence and performance analysis of the proposed algorithm without first considering distribution network constraints are presented. The agents are trained with 800 episodes and the evolution of the episode rewards is shown in Fig. 5. As the agents explore their action spaces according to the Ornstein-Uhlenbeck process (16), the episode rewards keep fluctuating until after 300 episodes when the training becomes stable. This shows that the proposed MADDPG achieves stable trainings despite the energy trading environment being non-stationary from the perspective of each agent.

For presentation purposes, we show performance of the proposed algorithm using the first 100 hours of the dataset which is shown in Fig. 4. Fig. 6 shows that the energy selling and buying price are high and low when net energy is positive and negative respectively. Positive net energy means that total consumption is more than total energy generation in the P2P energy sharing community, and the converse is true.

Using prosumer P1 as case study, Fig. 7 shows the optimal battery control actions. In the figure, the (negative) charge and (positive) discharge power of the battery are scaled to -1 and 1 in order to plot on one graph the battery SoC and power output. We can observe that the proposed algorithm can learn the optimal policy to charge/discharge the battery

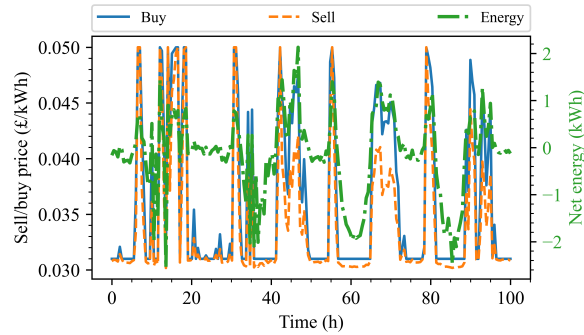


Figure 6: Variation of energy buying/selling price with total net energy of the prosumers without considering network constraints.

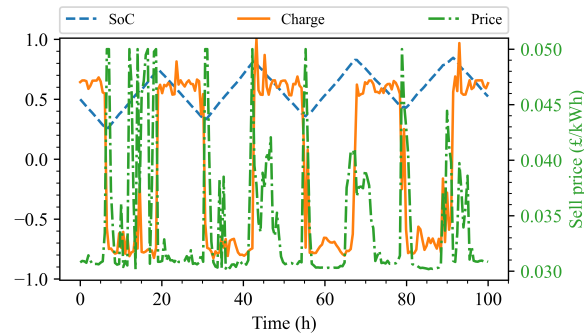


Figure 7: The charging (negative charge) and discharging (positive charge) action of the battery for prosumer 1 as it responds to the price (e.g. the selling price).

optimally. That is, to charge the battery when the price (e.g. selling price) is low and discharge the battery when the price is high, thus, reducing the energy costs.

### 4.3 Performance Analysis With Network Constraints

In this section we evaluate the effectiveness of the proposed algorithm for charging and discharging the batteries optimally while satisfying distribution network constraints. We use U.K's January 2017 wholesale (WS) market electricity price which is obtained from the institution of civil engineers (ICE)<sup>5</sup> as shown in Fig. 9 to determine the DNTs. We also assume that the loads have 0.95 power factor and the solar and battery energy storage systems have unity power factor. We present the results using the first 100 hours of the dataset in Fig. 4. Fig. 8 shows that learning is stable even when network constraints are considered.

Table 3: Average Episode Rewards.

Prosumer	Without DNTs	With DNTs	Difference (%)
P1	-2464.25	-1919.88	22.1
P2	-22661.05	-1919.55	15.1
P3	-2258.64	-2073.27	8.2

Table 3 compares the average episode rewards of Fig. 5 and Fig. 8. We can observe that prosumers benefit more (by having more than 8.2% of accumulated episode rewards) when they support distribution network constraints than when they do not. Rewards are highest for P1 because it has the lowest total load: P1, P2 and P3 have a total load of 194 kWh, 219.17 kWh and 240.76 kWh respectively. Thus, much of the renewable energy generation is sold to other prosumers for profit, hence increasing the rewards.

<sup>5</sup><https://www.ice.org.uk/knowledge-and-resources/briefing-sheet/the-changing-price-of-wholesale-uk-electricity/>

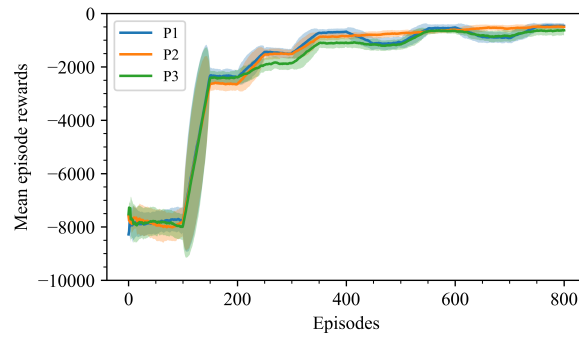


Figure 8: Mean episode rewards of the agents during the training while considering the network constraints.

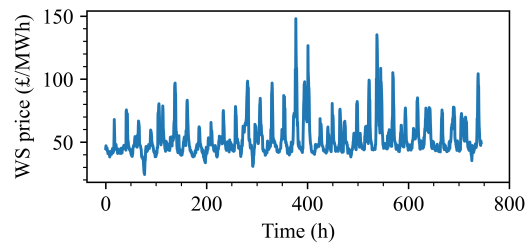


Figure 9: U.K's wholesale (WS) market price for the month of January, 2017.

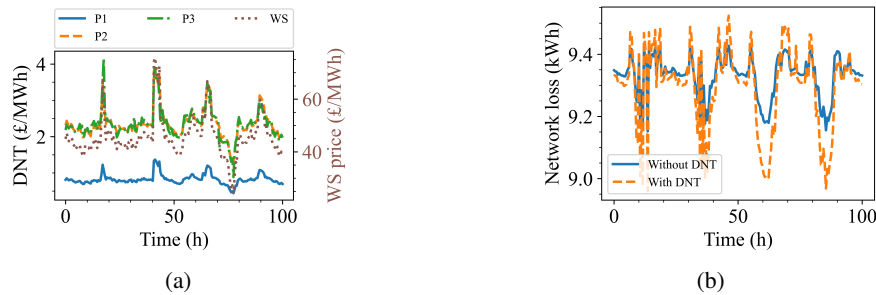


Figure 10: (a) Distribution network tariffs (DNT) for P1, P2 and P3, and (b) network loss comparison with and without DNTs.

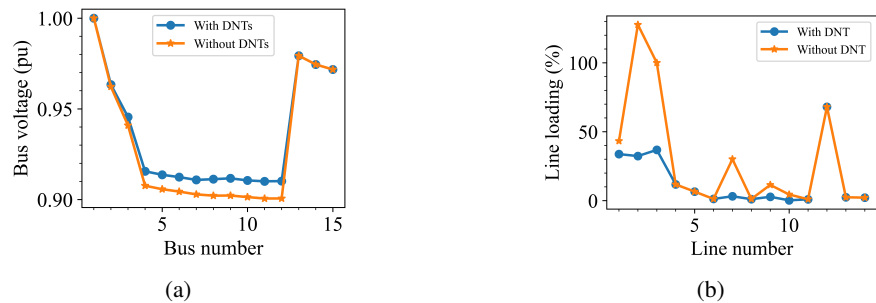


Figure 11: (a) Average bus voltage and (b) line loading with and without DNTs.

Further, Fig. 10a shows that P1 contributes least to network loss, congestion and voltage limit violation as it has the lowest value of the DNT. The DNTs change according to the wholesale price, making them economically suitable for influencing the consumption pattern of prosumers.

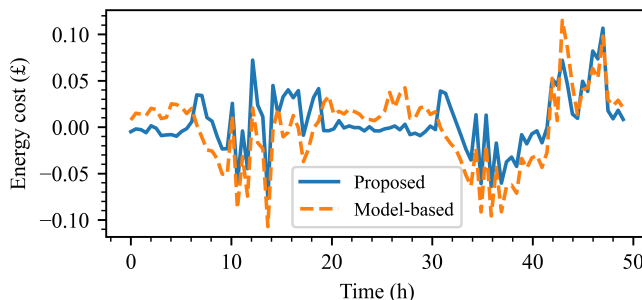


Figure 12: Comparison of total import (positive) and export (negative) energy costs for the prosumers between the proposed algorithm and model-based approach.

Benefits of incorporating DNTs in the P2P energy selling and buying price to the distribution network are also shown in Fig. 10b and Fig. 11. Fig. 10b shows that by incorporating the DNTs in the proposed algorithm, network losses are reduced. We can observe in Fig. 11a that voltage regulation is improved and in Fig. 11b that (peak) congestion is reduced by more than 50% when DNTs are used.

#### 4.4 Performance Comparison

In this section, effectiveness of the proposed algorithm at reducing import energy costs from the main grid is compared to that of a model-based approach derived from the method detailed in [31]. To reduce complexity and the number of variables required by the model-based approach, the results are presented using the first 50 hours of the dataset in Fig. 4. The comparison result is shown in Fig. 12. We can observe that the energy cost result obtained by the proposed algorithm is consistent with that obtained from the model-based approach, verifying that the results produced by the proposed algorithm are accurate.

## 5 Conclusion

In this paper, we have proposed a MADDPG-based algorithm to minimize the energy costs of prosumers participating in peer-to-peer energy trading while considering the distribution network constraints. The energy costs are minimized by scheduling the operation of batteries optimally as flexible assets. First, the battery scheduling process is modelled as a Markov decision process. Then, the MADDPG algorithm proposed (which is model-free) is used to learn the optimal battery scheduling strategies that minimize the energy costs. To satisfy the distribution network constraints, we have proposed the use of DNTs. The DNTs act as incentives enticing the prosumers to either reduce or increase their consumption as a way of satisfying the network constraints. Simulation results based on real-world datasets have shown that the algorithm proposed can optimally minimize the energy costs while also satisfying the distribution network constraints. Minimizing the energy costs by also scheduling the operation of flexible loads is a potential future work.

## References

- [1] Y. Zhou, J. Wu, C. Long, and W. Ming, "State-of-the-art analysis and perspectives for peer-to-peer energy trading," *Engineering*, vol. 6, no. 7, pp. 739–753, 2020.
- [2] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-peer trading in electricity networks: An overview," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3185–3200, 2020.
- [3] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation, and control*. John Wiley & Sons, 2013.
- [4] J. Qiu, J. Zhao, H. Yang, and Z. Y. Dong, "Optimal scheduling for prosumers in coupled transactive power and gas systems," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1970–1980, 2018.
- [5] L. M. Camarinha-Matos, "Collaborative smart grids—a survey on trends," *Renewable and Sustainable Energy Reviews*, vol. 65, pp. 283–294, 2016.
- [6] O. Abrishambaf, F. Lezama, P. Faria, and Z. Vale, "Towards transactive energy systems: An analysis on current trends," *Energy Strategy Reviews*, vol. 26, p. 100418, 2019.

- [7] C. Long, J. Wu, Y. Zhou, and N. Jenkins, "Peer-to-peer energy sharing through a two-stage aggregated battery control in a community microgrid," *Applied Energy*, vol. 226, pp. 261–276, 2018.
- [8] H. Wang and J. Huang, "Incentivizing energy trading for interconnected microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2647–2657, 2016.
- [9] J. Abdella and K. Shuaib, "Peer to peer distributed energy trading in smart grids: A survey," *Energies*, vol. 11, no. 6, p. 1560, 2018.
- [10] D. Wang, B. Liu, H. Jia, Z. Zhang, J. Chen, and D. Huang, "Peer-to-peer electricity transaction decision of user-side smart energy system based on SARSA reinforcement learning method," *CSEE Journal of Power and Energy Systems*, 2020.
- [11] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement learning-based microgrid energy trading with a reduced power plant schedule," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10728–10737, 2019.
- [12] T. Chen and W. Su, "Local energy trading behavior modeling with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 62806–62814, 2018.
- [13] W. Bi, Y. Shu, W. Dong, and Q. Yang, "Real-time energy management of microgrid using reinforcement learning," in *2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pp. 38–41, IEEE, 2020.
- [14] G. Gao, Y. Wen, X. Wu, and R. Wang, "Distributed energy trading and scheduling among microgrids via multi-agent reinforcement learning," *arXiv preprint arXiv:2007.04517*, 2020.
- [15] B.-G. Kim, Y. Zhang, M. Van Der Schaar, and J.-W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2187–2198, 2015.
- [16] J. Guerrero, A. C. Chapman, and G. Verbič, "Decentralized P2P energy trading under network constraints in a low-voltage network," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5163–5173, 2018.
- [17] T. Morstyn, A. Teytelboym, C. Hepburn, and M. D. McCulloch, "Integrating P2P energy trading with probabilistic distribution locational marginal pricing," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3095–3106, 2019.
- [18] J. Kim and Y. Dvorkin, "A p2p-dominant distribution system architecture," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2716–2725, 2019.
- [19] T. Morstyn and M. D. McCulloch, "Multiclass energy management for peer-to-peer energy trading driven by prosumer preferences," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 4005–4014, 2018.
- [20] A. Paudel, M. Khorasany, and H. B. Gooi, "Decentralized local energy trading in microgrids with voltage management," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1111–1121, 2020.
- [21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [23] T. Chen and S. Bu, "Realistic peer-to-peer energy trading model for microgrids using deep reinforcement learning," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pp. 1–5, IEEE, 2019.
- [24] J.-G. Kim and B. Lee, "Automatic P2P energy trading model based on reinforcement learning using long short-term delayed reward," *Energies*, vol. 13, no. 20, p. 5359, 2020.
- [25] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, and K. Li, "Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4513–4521, 2020.
- [26] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, "Deep reinforcement learning for smart building energy management: A survey," *arXiv preprint arXiv:2008.05074*, 2020.
- [27] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.
- [28] P. Olivella-Rosell, E. Bullich-Massagué, M. Aragüés-Peñalba, A. Sumper, S. Ø. Ottesen, J.-A. Vidal-Clos, and R. Villafáfila-Robles, "Optimization problem for meeting distribution system operator requests in local flexibility markets with distributed energy resources," *Applied Energy*, vol. 210, pp. 881–895, 2018.
- [29] T. Esmar and P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on Energy Conversion*, vol. 22, no. 2, pp. 439–449, 2007.

- 
- [30] S. Han, S. Han, and H. Aki, "A practical battery wear model for electric vehicle charging applications," *Applied Energy*, vol. 113, pp. 1100–1108, 2014.
  - [31] N. Liu, X. Yu, C. Wang, C. Li, L. Ma, and J. Lei, "Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3569–3583, 2017.
  - [32] A. Papavasiliou, "Analysis of distribution locational marginal prices," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4872–4882, 2017.
  - [33] L. Bai, J. Wang, C. Wang, C. Chen, and F. Li, "Distribution locational marginal pricing (dlmp) for congestion management and voltage support," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4061–4073, 2017.
  - [34] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*, pp. 157–163, Elsevier, 1994.
  - [35] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, PMLR, 2014.
  - [36] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
  - [37] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical Review*, vol. 36, no. 5, p. 823, 1930.
  - [38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.
  - [39] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.