Designing Efficient Local Flexibility Markets in the Presence of Reinforcement-Learning Agents

Designing Efficient Local Flexibility Markets in the Presence of Reinforcement-Learning Agents

Haoyang Zhang, *Graduate Student Member, IEEE,* Georgios Tsaousoglou, *Member, IEEE,* Sen Zhan, *Graduate Student Member, IEEE,* Koen Kok, *Senior Member, IEEE,* and Nikolaos G. Paterakis, *Senior Member, IEEE*

***Abstract*—Local Flexibility Markets (LFMs) are considered a promising framework towards resolving voltage and congestion issues of power distribution systems in an economically efficient manner. However, the need for location-specific flexibility services renders LFMs naturally imperfectly competitive and market efficiency is severely challenged by strategic participants that exploit their locally monopolistic power. Previous works have been considering either non-strategic participants, or strategic participants with perfect information (e.g. about the network characteristics etc) that can readily compute their payoff-maximizing bidding strategy. In this paper, we take on the problem of designing an efficient LFM in the more realistic case where market participants do not possess this information and, instead, learn to improve their bidding policies through experience. To that end, we develop a multi-agent reinforcement learning algorithm to model the participants' learning-to-bid process. In this framework, we first present two popular LFM pricing schemes (pay-as-bid and distribution locational marginal pricing) and expose that learning agents can discover ways to exploit them, resulting in severe dispatch inefficiency. We then present a game-theoretic pricing scheme that theoretically incentivizes truthful bidding and empirically demonstrate that this property improves the efficiency of the resulting dispatch also in the presence of learning agents. In particular, the proposed scheme is able to outperform the popular distribution locational marginal pricing (DLMP) scheme, in terms of efficiency, by a factor of $15-23\%$.**

***Index Terms*—AC OPF, Local flexibility market, Incentive compatibility, Multi-agent reinforcement learning, VCG**

I. Introduction

Load electrification and high penetration of behind-the-meter renewables, poses significant challenges for distribution system operators (DSOs) in ensuring the safe and reliable operation of power distribution systems. In particular, towards dealing with voltage and congestion issues, DSOs are envisioned to procure flexibility services from distributed energy resources. Such active resources include electric vehicles, building energy management systems, etc., which are increasingly configured with monitoring and control capabilities by aggregator entities that integrate them into electricity markets. It is now a well-recognized opportunity that such aggregators can also act as local flexibility service providers (FSPs), i.e. use their capability to control the profile of their registered resources towards offering flexibility services to the local DSO. Such flexibility services are location-specific and refer to a controlled power deviation from a pre-scheduled setpoint.

Making efficient flexibility dispatch decisions entails solving the Optimal Power Flow (OPF) problem of the distribution network. Towards resolving voltage and congestion issues in an economically optimal way, the OPF problem utilizes information about the cost of the flexibility actions of different FSPs. The need to motivate FSPs to communicate and offer their flexibility motivates the establishment of a market framework specifically for local flexibility services. Such a framework is now quite mature in the literature, while also gaining traction in the industry, and is referred to as a Local Flexibility Market (LFM) [1], [2]. Through such a market, FSPs communicate their flexibility capabilities to the DSO, in the form of location-and-time-specific *bids*, and the DSO decides upon each FSP's dispatch and payment.

Different pricing schemes have been adopted for defining the participants' payments, with the predominant ones being the pay-as-bid (PAB) and the distribution locational marginal pricing (DLMP). The PAB scheme compensates each participant for a deviation from its scheduled profile at the per-unit price that the participant bid for that deviation [3]. This, however, clearly creates an incentive for each participant to inflate its bids, in order to increase its own payments. On the contrary, the DLMP scheme defines a single per-unit clearing price per distribution network node and per timeslot, with which all deviations in that node and timeslot are compensated. These nodal prices, similar to the locational marginal prices used in a transmission system, are defined based on the marginal cost of the most expensive source that was dispatched in each node and time. Mathematically, they are instantiated by the Lagrange multipliers corresponding to the power balance constraints of the OPF problem [4].

Under the DLMP scheme, the optimal solution of the OPF problem is also a *competitive equilibrium*, i.e. a point from which no price-taking participant is willing to deviate unilaterally. In practice this means that, if the LFM was a perfectly competitive market, the DLMP scheme would incentivize each participant to bid no more than its marginal flexibility cost and, if dispatched, make a per-unit profit off the difference between its own cost and the nodal price (i.e. the marginal cost of the most expensive resource that was dispatched). Unfortunately, though, LFMs are notoriously markets of imperfect competition in which the above property no longer holds. In particular, an FSP that controls the resources under a particular node can have oligopolistic, or even monopolistic, power towards resolving voltage or

H. Zhang, S. Zhan, K. Kok, and N. G. Paterakis are with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: h.zhang@tue.nl).

G. Tsaousoglou is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark.

The first two authors contributed equally to this work.

congestion issues in that node's neighborhood.

Thereby, an FSP can exploit its position to manipulate the prices for its own benefit, and to the detriment of system efficiency. This phenomenon is well-documented for transmission systems (e.g. see [5] and related literature on strategic market participation) while, more recently, it was also demonstrated for distribution systems [6]. The interaction of multiple strategic participants can be modeled through an Equilibrium Problem with Equilibrium Constraints (EPEC), and the resulting dispatch can be remarkably inefficient [7].

Towards remedying such problems, Mechanism Design theory prescribes elaborate payment rules, such that a truthful costs' declaration becomes each participant's profit-maximizing strategy. In other words, truthful declarations are in equilibrium, i.e. no participant has an incentive to deviate towards an untruthful bid. Naturally, if each participant reveals its true costs to the DSO, the latter acquires all the necessary information to dispatch the system efficiently. In this case, we say that the incentives of the market participants (maximizing profits) and the one of the DSO (calculating an efficient dispatch) are aligned, and a mechanism that achieves such alignment is called *incentive compatible*. Indeed, recent works [8], [9] have leveraged concepts from Mechanism Design to design LFMs that are efficient and robust to strategic behavior.

Despite their strong theoretical foundations, both EPEC models and incentive-compatible mechanisms are fairly confined to the realm of theory. That is because, in practice, market participants do not possess the necessary information (e.g. the other participants' bids, the network characteristics, etc) to compute an optimal bidding strategy, which makes the notion of equilibrium elusive. Rather, the behavior of actual market participants with imperfect information, can be better modeled by *learning* algorithms through which a participant gradually improves its bidding policy by drawing on the knowledge acquired from past experience. Besides, the concept of having intelligent agents control the actions of flexible resources has been ramping up rapidly in the related literature (e.g. see [10] and references therein). In that direction, the authors in [11] demonstrate how intelligent learning agents, employing deep reinforcement learning, can learn to bid in LFMs.

In light of the above, this paper takes on the problem of designing a LFM (i.e. a dispatch algorithm and a pricing scheme) that retains market efficiency in the presence of self-interested learning agents bidding strategically on behalf of distributed resources. Regarding the dispatch algorithm, we present a linearization technique to approximate the quadratically-constrained AC-OPF problem and assess its accuracy, while configuring it with three possible pricing schemes for determining the participants' payments. The first two are the PAB and DLMP schemes which are, at least in principle, prone to manipulation. They serve as comparison benchmarks for the third (proposed) pricing scheme, based on Grove's pivot payment rule which constitutes the payment rule of the celebrated Vickrey-Clarke-Groves (VCG) mechanism: a game-theoretic mechanism renowned for achieving incentive compatibility and, subsequently, efficiency.

This paper's contribution is to examine whether the above theoretical properties of the three schemes propagate also to the more relevant practical case where the participants cannot calculate optimal, payoff-maximizing bids, but gradually improve their bidding policies by learning from past experience. Especially in the context of an LFM, we are interested in:

1) Whether learning agents, acting on behalf of FSPs, will in fact discover ways to exploit the Achilles hill of the DLMP and PAB schemes: the absence of incentive compatibility;
2) Whether, in the presence of learning nodes with agents, the proposed (theoretically incentive-compatible) scheme can manage to implement more efficient market outcomes than DLMP and PAB in practice.

The remainder of this paper is organized as follows: Section II introduces the system model and Section III formulates the efficient LFM design problem. Section IV formulates the three schemes discussed and elaborates on their properties. In Section V we introduce two multi-agent reinforcement learning algorithms through which the LFM participants learn to optimize their policies. Finally, Section VI presents the empirical evaluation of the three schemes' performance in a case study, and Section VII concludes the paper by answering the two questions posed above.

II. System Model

Consider a power distribution network, with $\mathcal{N}$ denoting the set of nodes and $\mathcal{L}$ the set of lines, operated over a horizon $\mathcal{T}$ of discrete timeslots of equal duration. Each node $n \in \mathcal{N}$ is characterized by a demand $\widehat{\mathrm{p}}_{nt}$ scheduled for each timeslot $t \in \mathcal{T}$ (e.g. as a result of a day-ahead market process). A node also features flexible energy resources, such that its net demand can be modified by regulating upwards (p^{up}_{nt}) or downwards (p^{dn}_{nt}), resulting in an effective demand p_{nt}, as in:

$$p_{nt} = \widehat{\mathrm{p}}_{nt} + p^{\mathrm{up}}_{nt} - p^{\mathrm{dn}}_{nt}, \quad n \in \mathcal{N}, t \in \mathcal{T}. \tag{1}$$

This flexibility is harnessed by the DSO who is responsible for maintaining the distribution system within safe operational margins. The system's safe operational region is modeled based on the linearized distribution flow (LinDistFlow) model [12] which approximates the common branch flow equations (DistFlow) by dropping the branch loss and shunt components [13]. The active/reactive power balance equations read as:

$$p_{nt} + \sum_{l \in \delta^+(n)} p^{\mathrm{L}}_{lt} = \sum_{l \in \delta^-(n)} p^{\mathrm{L}}_{lt}, \quad n \in \mathcal{N}, t \in \mathcal{T} \tag{2}$$

$$q_{nt} + \sum_{l \in \delta^+(n)} q^{\mathrm{L}}_{lt} = \sum_{l \in \delta^-(n)} q^{\mathrm{L}}_{lt}, \quad n \in \mathcal{N}, t \in \mathcal{T}. \tag{3}$$

where p^{L}_{lt} and q^{L}_{lt} represent the active and reactive power flow at branch l, while $\delta^-(n)$ denotes the lines connected to node n from preceding nodes, and $\delta^+(n)$ denotes the lines that start from node n and are connected to successor nodes. The apparent power on a branch is limited by

$$(p^{\mathrm{L}}_{lt})^2 + (q^{\mathrm{L}}_{lt})^2 \leq (\bar{S}^{\mathrm{L}}_l)^2, \quad l \in \mathcal{L}, t \in \mathcal{T}, \tag{4}$$

and the same stands true for a node's power:

$$(p_{nt})^2 + (q_{n,t})^2 \leq (\bar{S}_n)^2, \quad n \in \mathcal{N}, t \in \mathcal{T} \tag{5}$$

The relationship between the squared magnitude voltages v_{nt} and v_{it} of neighboring nodes n and i, is given by

$$v_{nt} = v_{it} - 2\mathrm{R}_l p^{\mathrm{L}}_{lt} - 2\mathrm{X}_l q^{\mathrm{L}}_{lt}, \quad n \in \mathcal{N}, l \in \delta^-(n) \cap \delta^+(i), t \in \mathcal{T}, \tag{6}$$

where R_l (X_l) is the resistance (reactance) of line l that connects n to i. All voltages should be within safe bounds:

$$\underline{\mathrm{v}} \leq v_{nt} \leq \bar{\mathrm{v}}, \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{7}$$

and the same stands true for the power exchanges at the feeder:

$$-\bar{\mathrm{p}}^{\mathrm{ex}} \leq p_{0t} \leq \bar{\mathrm{p}}^{\mathrm{ex}}, \quad t \in \mathcal{T}. \tag{8}$$

$$-\bar{\mathrm{q}}^{\mathrm{ex}} \leq q_{0t} \leq \bar{\mathrm{q}}^{\mathrm{ex}}, \quad t \in \mathcal{T}. \tag{9}$$

Finally, the relationship between a node's active and reactive power is given by

$$-\tan\theta_n p_{nt} \leq q_{nt} \leq \tan\theta_n p_{nt}, \quad n \in \mathcal{N}, t \in \mathcal{T} \tag{10}$$

which ensures that a node's resources are operated at a sufficiently high reactive and active power ratio $\tan\theta_n$ to limit the circulation of reactive power in the network (cf. [14]).

We assume, without loss of generality, that all the resources of a node are managed by a single FSP/aggregator. Each node is generally managed by a different FSP. A node's flexibility actions are bounded by:

$$0 \leq p^{\mathrm{up}}_{nt} \leq \overline{\mathrm{p}}^{\mathrm{up}}_n, \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{11}$$

$$0 \leq p^{\mathrm{dn}}_{nt} \leq \overline{\mathrm{p}}^{\mathrm{dn}}_n, \quad n \in \mathcal{N}, t \in \mathcal{T}. \tag{12}$$

To define the cost of a flexibility action, let us divide the range $[0, \overline{\mathrm{p}}^{\mathrm{up}}_n]$ (and $[0, \overline{\mathrm{p}}^{\mathrm{dn}}_n]$) into a set $\mathcal{B}^{\mathrm{up}}$ (and $\mathcal{B}^{\mathrm{dn}}$) of blocks, such that:

$$p^{\mathrm{up}}_{nt} = \sum_{b \in \mathcal{B}^{\mathrm{up}}} p^{\mathrm{up}}_{nbt}, \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{13}$$

$$p^{\mathrm{dn}}_{nt} = \sum_{b \in \mathcal{B}^{\mathrm{dn}}} p^{\mathrm{dn}}_{nbt}, \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{14}$$

where each block $b \in \mathcal{B}^{\mathrm{up/dn}}$ bears a different marginal flexibility cost $\mathrm{c}^{\mathrm{up}}_{nbt}$ (benefit $\mathrm{c}^{\mathrm{dn}}_{nbt}$). The cost (benefit) of a flexibility action p^{up}_{nt} (p^{dn}_{nt}) is then defined as

$$\lambda^{\mathrm{up}}_{nbt} = \mathrm{c}^{\mathrm{up}}_{nbt} + \zeta^{\mathrm{up}}_t, \quad n \in \mathcal{N}, b \in \mathcal{B}^{\mathrm{up}}, t \in \mathcal{T}, \tag{15}$$

$$\lambda^{\mathrm{dn}}_{nbt} = -\mathrm{c}^{\mathrm{dn}}_{nbt} - \zeta^{\mathrm{dn}}_t, \quad n \in \mathcal{N}, b \in \mathcal{B}^{\mathrm{dn}}, t \in \mathcal{T}, \tag{16}$$

i.e. the sum of the node's marginal cost increase (decrease) and the price ζ^{up}_t (ζ^{dn}_t) that the node will receive from (pay to) the balancing market for an upwards (downwards) change of its schedule.

III. Problem Formulation

Based on the definitions of the previous section, the optimal flexibility dispatch is defined as the optimizer of the following problem:

$$\min_{\mathcal{V}} \left\{ \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \left(\sum_{b \in \mathcal{B}^{\mathrm{up}}} \lambda^{\mathrm{up}}_{nbt} p^{\mathrm{up}}_{nbt} + \sum_{b \in \mathcal{B}^{\mathrm{up}}} \lambda^{\mathrm{dn}}_{nbt} p^{\mathrm{dn}}_{nbt} \right) \right\} \tag{17}$$

s.t. $(1) - (14)$,

where the problem's variables are

$$\mathcal{V} = \Bigg\{ \left(p^{\mathrm{L}}_{lt}, q^{\mathrm{L}}_{lt} \right)_{l \in \mathcal{L}, t \in \mathcal{T}}$$

$$\left(p_{nt} p^{\mathrm{up}}_{nt}, p^{\mathrm{dn}}_{nt}, q_{n,t}, v_{n,t} \left(p^{\mathrm{up}}_{nbt} \right)_{b \in \mathcal{B}^{\mathrm{up}}}, \left(p^{\mathrm{dn}}_{nbt} \right)_{b \in \mathcal{B}^{\mathrm{dn}}}, \right)_{n \in \mathcal{N}, t \in \mathcal{T}} \Bigg\}.$$

However, problem (17) cannot be solved directly since the DSO is not aware of the nodes' (agents') flexibility costs. This motivates the establishment of a Local Flexibility Market (LFM) mechanism, through which the DSO tries to elicit this information from the nodes, so that it can then dispatch the system efficiently. A mechanism $\mathcal{M} = (\mathcal{D}, \mathcal{P})$, in this context, asks each node to declare its flexibility costs $(\lambda^{\mathrm{up}}_{nbt})_{b \in \mathcal{B}^{\mathrm{up}}}, (\lambda^{\mathrm{dn}}_{nbt})_{b \in \mathcal{B}^{\mathrm{dn}}}$ and deploys a dispatch rule $\mathcal{D}$ and a pricing scheme $\mathcal{P}$, each one analysed in the following subsections, before introducing the LFM Design Problem.

A. Dispatch Rule

The dispatch rule defines that the nodes' flexibility actions are determined by solving problem (17), but using the declared costs $\widehat{\lambda}^{\mathrm{up}}_{nbt}, \widehat{\lambda}^{\mathrm{dn}}_{nbt}$ instead of the (unknown to the DSO) true costs $\lambda^{\mathrm{up}}_{nbt}, \lambda^{\mathrm{dn}}_{nbt}$, i.e.

$$\min_{\mathcal{V}} \left\{ \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \Big(\sum_{b \in \mathcal{B}^{\mathrm{up}}} \widehat{\lambda}^{\mathrm{up}}_{nbt} p^{\mathrm{up}}_{nbt} + \sum_{b \in \mathcal{B}^{\mathrm{dn}}} \widehat{\lambda}^{\mathrm{dn}}_{nbt} p^{\mathrm{dn}}_{nbt} \Big) \right\} \tag{18}$$

s.t. $(1) - (14)$.

Still though, even with given (declared) cost parameters, problem (18) is a convex quadratically-constrained problem (due to constraints (4) and (5)) which often leads to numerical stability problems [15]. This numerical stability issue arises from factors such as potential ill-conditioning of the Hessian matrix and finite precision of the commercial solver tied to quadratic terms, leading to sensitivity to small input changes and impacting the stability of the solving process. Following the work in [15] and [16], a piecewise linearization method is applied as shown in Fig. 1. Using the branch power constraint (4) as an example, the x and y axes represent the active and reactive power branch flow, respectively, and the original circular feasible region is linearized into a polygon with N^{arc} linear segments whose polar angle is:

$$\theta^{arc}_{n^{arc},l,t} = \frac{360^\circ}{N^{arc}}, \quad l \in \mathcal{L}, t \in \mathcal{T}. \tag{19}$$

Details of the linearization can be found in Appendix A (omitted in this version due to space limitations)

B. Pricing Scheme

The role of the pricing scheme is to use payments

$$r_n = f\Big(\left(\widehat{\lambda}^{\mathrm{up}}_{nbt}, \widehat{\lambda}^{\mathrm{dn}}_{nbt} \right)_{t \in \mathcal{T}}; \left(\widehat{\lambda}^{\mathrm{up}}_{mbt}, \widehat{\lambda}^{\mathrm{dn}}_{mbt} \right)_{m \neq n, t \in \mathcal{T}} \Big), \quad n \in \mathcal{N}, \tag{20}$$

calculated as a function f of the nodes' cost declarations (and subsequent dispatch), so as to incentivize the nodes to declare their flexibility costs truthfully (i.e. $\widehat{\lambda}^{\mathrm{up}}_{nbt} = \lambda^{\mathrm{up}}_{nbt}$ and $\widehat{\lambda}^{\mathrm{dn}}_{nbt} = \lambda^{\mathrm{dn}}_{nbt}$). Naturally, if the DSO elicits truthful cost

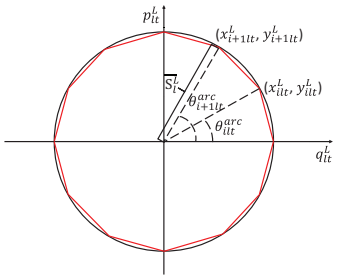


Fig. 1: Piecewise linearization of the quadratic constraint.

declarations, the dispatch rule (18) results in the optimal decisions (since it becomes equivalent to problem (17)). If the pricing scheme achieves this goal, we say that the mechanism is truthful, or *incentive compatible*. On the other hand, if a node can increase its own payoff by making untruthful declarations, problem (18) is solved using false information and the resulting dispatch is suboptimal.

C. The LFM Design Problem

We consider rational nodes where each one tries to choose the declarations $\widehat{\lambda}^{*\text{up}}_{nbt}, \widehat{\lambda}^{*\text{dn}}_{nbt}$ that result in a maximization of its own payoff, where a node's payoff-maximizing choice is defined as:

$$\begin{aligned}&\widehat{\lambda}^{*\text{up}}_{nbt}, \widehat{\lambda}^{*\text{dn}}_{nbt} = \\ &\quad \underset{\widehat{\lambda}^{\text{up}}_{nbt}, \widehat{\lambda}^{\text{dn}}_{nbt}}{\text{argmax}} \{r_n - c_n\} \\ &\qquad \text{s.t. (20)}, \\ &\qquad\quad c_n = \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\Big(\sum_{b\in\mathcal{B}^{\text{up}}}\lambda^{\text{up}}_{nbt}p^{*\text{up}}_{nbt} + \sum_{b\in\mathcal{B}^{\text{up}}}\lambda^{\text{dn}}_{nbt}p^{*\text{dn}}_{nbt}\Big), \\ &\qquad\quad p^{*\text{up}}_{nbt}, p^{*\text{dn}}_{nbt} \in (18),\end{aligned} \tag{21}$$

or, in words, the node's payoff is the payment received from the mechanism (by (20)) minus the actual cost c_n (which is based on the true costs $\lambda^{\text{up}}_{nbt}, \lambda^{\text{dn}}_{nbt}$) it bears from the flexibility actions $p^{*\text{up}}_{nbt}, p^{*\text{dn}}_{nbt}$ that result from the dispatch rule (18). Notice that a node's declarations $\widehat{\lambda}^{\text{up}}_{nbt}, \widehat{\lambda}^{\text{dn}}_{nbt}$ affect its own payoff by parameterizing both the node's payment calculation (cf. (20)) as well as problem (18) that determines the node's dispatch $p^{*\text{up}}_{nbt}, p^{*\text{dn}}_{nbt}$. Also, note that a node n cannot solve problem (21) to optimality since, apart from the problem's non-convex nature, the problem's parameters (e.g. network characteristics etc.) are not known to n. However, what a node *can* (and is expected to) do, is to learn how to gradually adapt its declarations towards improving its payoff over time. Thereupon, the nodes' sequential interactions, through their daily declarations and subsequent payoffs, take the form of a Partially Observable Markov Game (POMG) $\mathcal{G} = (\boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{A}}, \rho, \omega)$, where:

- $\boldsymbol{\mathcal{O}} = \times_{n\in\mathcal{N}}\mathcal{O}_n$ is the joint observation space, where a node's observation $o_{nt} \in \mathcal{O}_n$ is defined by the present timeslot t, its schedule, and upwards/downwards balancing costs, as in

$$o_{nt} = [t, \widehat{\text{p}}_{nt}, \zeta^{\text{up}}_t, \zeta^{\text{dn}}_t], \quad n \in \mathcal{N}, t \in \mathcal{T}; \tag{22}$$

- $\boldsymbol{\mathcal{A}} = \times_{n\in\mathcal{N}}\mathcal{A}_n$ is the joint action space, where a node's action $a_{nt} \in \mathcal{A}_n$ is the cost declarations (or *bids*) for each flexibility block, i.e.:

$$a_{nt} = \Big[(\widehat{\lambda}^{\text{up}}_{nbt})_{b\in\mathcal{B}^{\text{up}}}, (\widehat{\lambda}^{\text{dn}}_{nbt})_{b\in\mathcal{B}^{\text{dn}}}\Big], \quad n \in \mathcal{N}, t \in \mathcal{T}; \tag{23}$$

- A node's reward $R_{nt} = \rho(o_{nt}, a_{nt})$ is defined by the objective function of problem (21):

$$R_{nt} = r_n - c_n, \quad n \in \mathcal{N}, t \in \mathcal{T}; \tag{24}$$

- The observation transition function ω, where

$$o_{nt+1} = \omega(o_{nt}), \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{25}$$

  is a random process, (it will be specifically defined in the simulation setup).

For preliminaries on POMGs, the reader is referred to Appendix B (omitted in this version due to space limitations). A node's policy is a function that maps a given observation to an action, i.e.:

$$a_{nt} = \pi_n(o_{nt}; \xi_n), \quad n \in \mathcal{N}, t \in \mathcal{T}, \tag{26}$$

where ξ_n is a vector of parameters that define the policy (e.g. the weights of a neural network). Once all nodes make their declarations, the DSO decides the dispatch and payments and the game transitions to the next stage. In the POMG, a node gradually learns to make actions that improve its own payoff, over episodes of experienced observation-action-reward triples. Let $\mathcal{E}$ denote a given budget of experience episodes and $\psi_{ne} = (o_{nte}, a_{nte}, R_{nte})_{t\in\mathcal{T}}$ denote the observation-action-reward trajectory of n in experienced episode e. After each episode, a node updates its policy parameters as

$$\xi_{ne+1} = \ell(\xi_{ne}, \psi_{ne}), \quad n \in \mathcal{N}, e \in \mathcal{E}, \tag{27}$$

where $\ell(\cdot)$ is the policy update or *learning* rule.

Based on the above, and in the presence of learning nodes, the LFM designer's objective is to design a payment rule $f(\cdot)$ that prompts the nodes towards learning bidding policies that result in efficient LFM dispatch decisions. The pursuit of such an efficient payment rule design is formalized as

$$\begin{aligned}&\min_{f(\cdot)}\Bigg\{\sum_{e\in\mathcal{E}}\sum_{n\in\mathcal{N}}\mathbb{E}\Big[\sum_{t\in\mathcal{T}}\Big(\sum_{b\in\mathcal{B}^{\text{up}}}\lambda^{\text{up}}_{nbt}p^{*\text{up}}_{nbt} + \sum_{b\in\mathcal{B}^{\text{dn}}}\lambda^{\text{dn}}_{nbt}p^{*\text{dn}}_{nbt}\Big)\Big]\Bigg\} \\ &\text{s.t. (20), (22)} - \text{(27)}, \\ &\quad p^{*\text{up}}_{nbt}, p^{*\text{dn}}_{nbt} \in (18),\end{aligned} \tag{28}$$

where the expectation is over joint trajectories $\boldsymbol{\psi}_e$ conditioned over joint policies $\boldsymbol{\pi}_e$. The intuition behind problem (28), i.e., the sequence of dependencies that link the problem's objective function to the choice of the payment rule f, is as follows: The objective function depends on the dispatch decisions $p^{*\text{up}}_{nbt}, p^{*\text{dn}}_{nbt}$ which, through problem (18), depend on the participants bids (23); the latter comes as a result of the bidding policies (26) which are parameterized by the policy parameters $\xi_{n,e}$; those, in turn, are updated by the learning rule (27) based on each episode's experienced trajectory $\psi_{n,e}$ which includes the experienced rewards (24). These rewards

are partially defined by the payments r_n, and subsequently by the payment rule function f, as defined in (20).

In the following section, we present three pricing rules f, towards approximating an attractive solution to problem (28).

IV. Pricing Scheme Design

In this section, we present three pricing schemes:

- the pay-as-bid (PAB) pricing scheme which is often used in practice,
- the Distribution Locational Marginal Pricing (DLMP) scheme that is a popular choice in the related literature,
- the proposed scheme, based on Groves' pivot rule,

and discuss their properties before empirically assessing their performance.

A. Pay-as-Bid

Under the PAB pricing mechanism [3], a node's payment for an operational horizon $\mathcal{T}$ is defined as

$$r_n^{\mathrm{PAB}} = \sum_{t\in\mathcal{T}} \Big(\sum_{b\in\mathcal{B}^{\mathrm{up}}} \widehat{\lambda}^{\mathrm{up}}_{nbt} p^{*\mathrm{up}}_{nbt} + \sum_{b\in\mathcal{B}^{\mathrm{dn}}} \widehat{\lambda}^{\mathrm{dn}}_{nbt} p^{*\mathrm{dn}}_{nbt} \Big), \quad n\in\mathcal{N}, \tag{29}$$

i.e., n is compensated for its dispatched quantities $p^{*\mathrm{up}}_{nbt}, p^{*\mathrm{dn}}_{nbt}$ at the per-unit prices $\widehat{\lambda}^{\mathrm{up}}_{nbt}, \widehat{\lambda}^{\mathrm{dn}}_{nbt}$ that n declared. This scheme is clearly not *incentive compatible* since it incentivizes each node to declare higher costs than its true ones. In fact, if a participant declares its true costs then, for any dispatch, its payments would equal its costs resulting in a payoff of zero. However, it is not easy for a node to determine by how much it should inflate its bids. If a low-cost node overshoots, it could well happen that another, higher-cost node (that does not inflate its declarations as much) would be dispatched instead. Clearly, dispatching a high-cost node instead of a low-cost one, due to false declarations, is a prominent source of inefficiency. Thus, the PAB is not expected to achieve a good performance in terms of the objective function of problem (28).

B. Distribution Locational Marginal Pricing

Similar to the locational marginal price scheme that is used in the wholesale, transmission-level electricity market, the DLMPs constitute node-differentiated prices for a distribution grid. Given cost declarations $\widehat{\lambda}^{\mathrm{up}}_{nbt}, \widehat{\lambda}^{\mathrm{dn}}_{nbt}$ from all nodes, the DLMPs $(\eta_{nt})_{n\in\mathcal{N},t\in\mathcal{T}}$ are instantiated by the optimal dual variables of the active power balance constraints (2), as in

$$0 \le \eta_{nt} \perp \Big(p_{nt} + \sum_{l\in\delta^-(n)} p^{\mathrm{L}}_{lt} - \sum_{l\in\delta^+(n)} p^{\mathrm{L}}_{lt} \Big), n\in\mathcal{N}, t\in\mathcal{T}, \tag{30}$$

and they represent the additional flexibility cost at t resulting from a marginal increase in a node's active power. This interpretation follows from computing the first-order partial derivative of the Lagrangian function of problem (18), cf. [4], [8], [17]; the reader is referred to Appendix C (omitted in this version due to space limitations) for the details. The DLMP scheme payments take the form:

$$r_n^{\mathrm{DLMP}} = \sum_{t\in\mathcal{T}} \eta_{nt} p^*_{nt}, \quad n\in\mathcal{N}. \tag{31}$$

While marginal pricing schemes are well-known to be efficient for perfectly competitive markets, an LFM is a prominent example of an imperfect competition market; that is, due to the spatial characteristics of the underlying distribution network, a node often possesses oligopolistic (sometimes even monopolistic) power over resolving voltage and congestion problems within the node's neighborhood. Indeed, it has been previously demonstrated that, under the DLMP scheme, a node can increase its own payoff by strategically inflating its bids [8], [9]. Nonetheless, in practice, a node does not have perfect information over the network characteristics or the other nodes' bids, and thus it cannot calculate its optimal, payoff-maximizing bid. It remains an open question whether a node can actually discover/learn payoff-maximizing untruthful bidding policies, through experience, in the realistic case where the node cannot solve the payoff-maximizing problem (21) directly. This question will be addressed in the case study, along with the main question of the resulting dispatch efficiency (cf. (28)) under the learned bidding policies induced by the DLMP scheme.

C. The Grove's pivot rule

Different from the PAB and DLMP mechanisms where a node is compensated based on its dispatch quantities multiplied by respective prices, the Grove's Pivot Rule (GPR) defines a node's payment based on its "externality", i.e., the difference that the node's presence makes in the aggregated cost of the other nodes' dispatch, as in

$$r_{nt}^{\mathrm{GPR}} = C_{-n}(\mathcal{N}_{-n}) - C_{-n}(\mathcal{N}), \quad n\in\mathcal{N}, \tag{32}$$

where

$$C_{-n}(\mathcal{N}_{-n}) = \tag{33}$$
$$\min_{\mathcal{V}} \Bigg\{ \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{N}/\{n\}} \Big(\sum_{b\in\mathcal{B}^{\mathrm{up}}} \widehat{\lambda}^{\mathrm{up}}_{mbt} p^{\mathrm{up}}_{mbt} + \sum_{b\in\mathcal{B}^{\mathrm{dn}}} \widehat{\lambda}^{\mathrm{dn}}_{mbt} p^{\mathrm{dn}}_{mbt} \Big) \Bigg\}$$
$$\text{s.t. } (1)-(16),$$

is the counterfactual system cost that would have been if n was not participating in the market, and

$$C_{-n}(\mathcal{N}) = \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{N}/\{n\}} \Big(\sum_{b\in\mathcal{B}^{\mathrm{up}}} \widehat{\lambda}^{\mathrm{up}}_{mbt} p^{*\mathrm{up}}_{mbt} + \sum_{b\in\mathcal{B}^{\mathrm{dn}}} \widehat{\lambda}^{\mathrm{dn}}_{mbt} p^{*\mathrm{dn}}_{mbt} \Big) \tag{34}$$

is the system cost of problem (18) solved with n present but without accounting for n's part of the cost.

Grove's pivot rule, combined with the dispatch rule (18) form the renowned Vickery-Clarke-Groves (VCG) mechanism, which is known to be incentive compatible, in the sense that truthful bidding is a dominant strategy for each participant. In our context, this means that the true costs $\lambda^{\mathrm{up}}_{nbt}, \lambda^{\mathrm{dn}}_{nbt}$ are the solutions of each node's payoff-maximizing problem (21). Nonetheless, given the intractability of problem (21), it remains an open question whether the attractive theoretical properties of this scheme propagate to a setting with repeatedly interacting learning nodes. Thus, in the case study we will empirically assess whether the nodes can discover/learn that bidding truthfully maximizes their payoffs, or whether the

efficiency-maximizing truthful bidding policies will be elusive in practice. Ultimately, we are interested in the efficiency of the learned policies induced when the GPR is adopted as the LFM's pricing scheme.

V. Bidding Policies based on MARL

This section specifies the policies and learning rules (cf. (26) and (27)) that each node uses to optimize its declarations towards maximizing its own payoff, i.e., approximating the solution to problem (21) under any given pricing scheme. Specifically, we model the nodes' interaction using the powerful framework of Multi-Agent Reinforcement Learning (MARL). The choice to adopt this approach is justified by the dynamic adaptation of strategies employed by other nodes as they learn and adapt from their experiences. Hence, the multi-agent deep deterministic policy gradient (MADDPG) algorithm [18] is deployed to simulate the strategic bidding behavior of the flexible nodes. The algorithm forms a distributed counterpart of the centralized DDPG, where decentralized nodes maintain local critic networks based on the aggregated observations and actions of all nodes and provide feedback to the local actors. For each node n, the critic network $Q_n(\boldsymbol{o}_t, \boldsymbol{a}_t|\theta_n^Q)$ determines the Q-value based on the joint observation $\boldsymbol{o}_t = (o_{1t}, o_{2t}, ..., o_{Nt})$ and joint action $\boldsymbol{a}_t = (a_{1t}, a_{2t}, ..., a_{Nt})$ of all nodes at time step t, and the actor network $\mu_n(o_{nt}|\theta_n^\mu) = \arg\max_{a_{nt}} Q_n(\boldsymbol{o}_t, \boldsymbol{a}_t)$ determines the greedy action to take based on the local observation o_{nt}. To maintain consistent target values during TD backups and enhance the stability of the learning process, supplementary networks known as the target critic network $Q'_n(\boldsymbol{o}_t, \boldsymbol{a}_t|\theta_n^{Q'})$ and target actor network $\mu'_n(o_n|\theta_n^{\mu'})$ are employed, in addition to the regular critic and actor networks. The target networks possess identical network architectures as the standard networks and soft updates by copying the parameters from their respective regular networks at fixed intervals as follows:

$$\theta_n^{Q'} = \tau\theta_n^Q + (1-\tau)\theta_n^{Q'} \tag{35}$$

$$\theta_n^{\mu'} = \tau\theta_n^\mu + (1-\tau)\theta_n^{\mu'} \tag{36}$$

where the value of the learning rate τ is close to 1. To train the networks, a reply buffer $\mathcal{R}_n$ of a size $N^{\mathcal{R}}$ is created for each agent, and the experience of the nodes is stored in the form of $(\boldsymbol{o}_t, \boldsymbol{a}_t, r_{nt}, \boldsymbol{o}_{t+1})$. When training the networks, a mini-batch of size N^m is sampled uniformly from the reply buffer, and the parameters of the critic and actor networks are updated. Given the mini-batch above, node updates the parameters θ_n^Q of the critic network by minimizing the Smooth L1 Loss function $L(\theta_n^Q)$ defined as:

$$L(\theta_n^Q) = \begin{cases} \frac{1}{N^m}\sum_{n=1}^{N^m} 0.5\zeta_n^2 & \text{if } |\zeta_n| \leq 1, \\ \frac{1}{N^m}\sum_{n=1}^{N^m}(|\zeta_n| - 0.5) & \text{otherwise,} \end{cases} \tag{37}$$

$$\zeta_n = r_{nt} + \gamma Q'_n(\boldsymbol{o}_{t+1}, \boldsymbol{a}_{t+1}|\theta_n^{Q'}) - Q_n(\boldsymbol{o}_t, \boldsymbol{a}_t|\theta_n^Q) \tag{38}$$

The objective of the target is to minimize the TD error, denoted as ζ_n, within the mini-batch. which is defined as the gap between the estimated Q-value between target networks and regular networks, and the parameters in the neural network can be updated via the back-propagation method. To evaluate the performance of the actor network, a performance objective $J(\theta_n^\mu)$ is formulated in (39) and updated based on the policy gradient theorem in (40), aiming to maximize the expected profit return, as described below:

$$J(\theta_n^\mu) = \mathbf{E}(Q_n(\boldsymbol{o}_t, \boldsymbol{a}_t(\theta_n^\mu)|\theta_n^Q) \tag{39}$$

$$\nabla_{\theta_n^\mu} J(\theta_n^\mu) = \frac{1}{N^m}\sum_{n=1}^{N^m}(\nabla_{a_{nt}} Q(\boldsymbol{o}_t, \boldsymbol{a}_t(\theta_n^\mu)|\theta_n^Q) \nabla_{\theta_n^\mu} a_{nt}(\theta_n^\mu)) \tag{40}$$

In addition, the Gaussian noise $\mathcal{N}_{nt}^\epsilon$ is added to the action during exploration to encourage exploration and avoid falling into the local optima as follows:

$$a_{nt} = clip(\mu_n(o_{nt}|\theta_n^\mu) + \mathcal{N}_{nt}^\epsilon(0, \sigma_n^\epsilon), a_{nt}^{min}, a_{nt}^{max}) \tag{41}$$

where σ_n^ϵ is the standard deviation of the Gaussian noise. To mitigate excessive exploration of actions over time and facilitate convergence, the parameter σ_n^ϵ is decaying linearly with episodes by multiplying with a decaying factor κ at the end of each episode.

However, MADDPG encounters the challenge of the curse of dimensionality problem. This arises from the exponential expansion of the joint state-action space of the critic network as the number of nodes increases, rendering the training process intractable [19]. As an extension of MADDPG, MF-MADDPG integrates the MF mechanism into its framework that approximates the interactions among a large number of nodes by considering the average effect from the overall population or neighboring nodes as shown in Fig. 2. It assumes $\bar{a}_{nt}$ represents the average action of node n's neighborhood. The action a_j of each neighbor j can be expressed as $\bar{a}_{nt}$ combined with a small fluctuation δa_{nt}^j given by:

$$a_{nt} = \bar{a}_{nt} + \delta a_{nt}^j, \text{ where } \bar{a}_{nt} = \frac{1}{N-1}\sum_{j\in\mathcal{N}/n} a_{jt} \tag{42}$$

Subsequently, the original critic network Q_n^{org} can be decomposed into individual pairwise critic networks, denoted as Q_n and further simplified through the application of a twice-differentiable Taylor expansion, as described below:

$$\begin{aligned} Q_n^{org}(\boldsymbol{o}_t, \boldsymbol{a}_t) &= \frac{1}{N-1}\sum_{j\in\mathcal{N}/n} Q_n(\boldsymbol{o}_t, a_{nt}, a_{jt}) \\ &= \frac{1}{N-1}\sum_{j\in\mathcal{N}/n}\big(Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt}) + \nabla_{\bar{a}_{nt}} Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt})\cdot\delta a_{nt}^j \\ &\quad + \frac{1}{2}\delta a_{nt}^j \cdot \nabla^2_{\bar{a}_{nt}^j} Q_n(\boldsymbol{o}_t, a_{nt}, \tilde{a}_{nt})\cdot\delta a_{nt}^j\big) \\ &= Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt}) + \nabla_{\bar{a}_{nt}} Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt})(\frac{1}{N-1}\sum_{j\in\mathcal{N}/n}\delta a_{nt}^j) \\ &\quad + \frac{1}{2(N-1)}\sum_{j\in\mathcal{N}/n}\big(\delta a_{nt}^j \cdot \nabla^2_{\bar{a}_{nt}^j} Q_n(\boldsymbol{o}_t, a_{nt}, \tilde{a}_{nt})\cdot\delta a_{nt}^j\big) \\ &= Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt}) + \frac{1}{2(N-1)}\sum_{j\in\mathcal{N}/n} R_{n,t}(a_{jt}) \\ &\approx Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt}) \end{aligned} \tag{43}$$

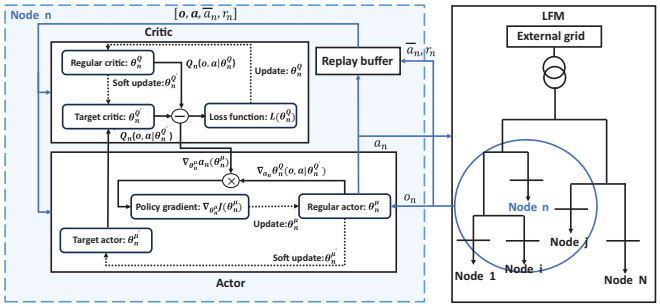


Fig. 2: MF-MADDPG framework

Algorithm 1 MF-MADDPG Algorithm

Set hyperparameters and reply buffer $\mathcal{R}$
Randomly initialize regular (target) critic and actor networks of the nodes θ_n^Q ($\theta_n^{Q'}$), θ_n^μ ($\theta^{\mu'}$)
while $episode \leq N^E$ **do**
  $t = 1$ and nodes take the initial observations $\boldsymbol{o}_{nt}$
  while $t \leq T$ **do**
    Nodes take actions $\boldsymbol{a}_t$ by (41)
    Obtain rewards $\boldsymbol{r}_t$
    Get the observation of the next time step $\boldsymbol{o}_{t+1}$
    Store $(\boldsymbol{o}_t, \boldsymbol{a}_t, \bar{a}_{nt}, r_{nt}, \boldsymbol{o}_{t+1})$ in $\mathcal{R}_n$
    Update critic network by (37) and (38)
    Update actor network by (39) and (40)
    Soft update target networks by (35) and (36)
    $t \leftarrow t + 1$
  $\sigma_n^\epsilon \leftarrow \kappa \sigma_n^\epsilon$ for each node n
  $episode \leftarrow episode + 1$

where the first-order term in the equation can be omitted since $\sum_{j \in \mathcal{N}/n} \delta a_{nt}^j = 0$. In addition, the second-order term can be disregarded as the second-order remainders $R_{nt}(a_{jt}) = \delta a_{nt}^j \cdot \nabla^2_{\bar{a}_{nt}^j} Q_n(\boldsymbol{o}_t, a_{nt}, \tilde{a}_{nt}) \cdot \delta a_{nt}^j$ are determined using the external action distribution of node j based on the perspective node n which is thus essentially a random variable. Furthermore, it has been proven in [20] that the remainders $R_{nt}(a_{jt})$ are bounded within the symmetric interval $[-2M, 2M]$, given the mild condition that the Q-function is M-smooth. As a result, the second order remainder R_{nt} acts as a small fluctuation near zero. And the original critic networks in (35) to (40) can be approximated by the critic network $Q_n(\boldsymbol{o}_t, a_{nt}, \bar{a}_{nt})$. In contrast to the original critic network, the input dimension of the critic network has been reduced from $[N \times (dim^o + dim^a)]$ to $[N \times dim^o + 2 \times dim^a)]$, where dim^o and dim^a are the dimension of the observations and actions, respectively, resulting in an enhancement of the scalability of the MARLs. The whole process of the MF-MADDPG algorithm is shown in Algorithm 1.

VI. Case Study

A. Experimental Setting

This section evaluates the performance of the three LFM pricing mechanisms. In the baseline scenario, all nodes bid their costs truthfully. Conversely, in the strategic bidding scenarios, nodes employ a MARL algorithm to discover the (possibly untruthful) bidding policies that maximize their own payoffs. The test system is a 20 kV 11-bus MV-DN as shown in Fig. 3, where Bus 1 is the slack bus of the system and the base for power is 1 MVA. The network is assumed to be balanced, allowing the use of a single-phase model, and the switches at branch $(4, 11)$ and $(6, 7)$ are disconnected so that the system is operated in a radial configuration. Details related to the grid parameters can be found in [21].

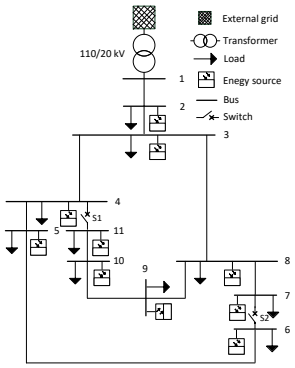


Fig. 3: Test system

The network comprises 10 active nodes (buses 2 to 11). Each node's resources are managed by an aggregator that coordinates the flexibility actions of households located behind the node. Thus, in alignment with the system model, it is regarded that each node is managed by a single agent. The nodes first participate in the day-ahead electricity market and obtain the scheduled power profiles for the next day. After that, they calculate the available upward and downward flexibility capacities and submit the flexibility bids to the LFM. After the LFM is cleared, they receive the payments from the DSO for providing the flexibility services and then pay for the imbalance costs arising from deviations in the day-ahead scheduled profiles due to the provision of flexibility services. The test case spans a duration of 4 weeks, with the initial three weeks allocated for training purposes and the final week designated for testing. The positive and negative imbalance prices are sampled from 10^{th} January to 6^{th} February 2022, in the Netherlands using data from ENTSO-E [22]. The scheduled profile of each node is sampled from Simbench [23]. The hyperparameters of the MARL algorithm are determined using the grid search method. A discount factor of 0.99 is chosen for the critic network. The size of the minibatch and reply buffer are 35 and 5×10^4, respectively. The critic and actor networks are updated by the Adam optimizer with learning rates of 5×10^{-4} and 10^{-4}, respectively. The target networks are then updated with a soft update rate of 0.001 and are updated 5 times in each episode. The mean and standard deviation of the Gaussian noise for exploration are 0 and 0.3, respectively. The MARL algorithms are implemented in Python 3.8 using Pytorch [24] and the optimization model is solved using the GUROBI 9.5.2 solver [25] on a computer equipped with a 6-core 2.60 GHz Intel(R) Core(TM) i7-9750H CPU and 16 GB of RAM.

B. Evaluating the Dispatch Rule

We first test the linearized dispatch rule (18), in terms of how accurately it approximates the original, quadratically

TABLE I: Objective value of different LFM dispatch models

	Objective value (€)						
Day	1	2	3	4	5	6	7
Quadratic LFM	-4805.9	-1429.5	63.6	-459.1	950.3	5064.0	3468.4
Linear LFM	-4792.1	-1423.2	67.9	-453.7	959.4	5088.9	3482.1
Difference	0.3%	0.4%	6.7%	1.2%	1.0%	0.5%	0.4%

constrained LFM problem (17). To this end, the objective values of both the linearized and the quadratic LFM formulations are compared for the test week, as presented in Table I. When all nodes bid truthfully, the objective value represents the system's cost, comprising the sum of operational costs and the imbalance costs incurred by nodes for providing flexibility services. The objective values demonstrate that the adopted linear LFM yields higher system costs than the quadratic LFM over 7 days. This is because the linear LFM reduces the feasible region of the energy sources and branch capacity, leading to a suboptimal outcome. However, the differences between them are relatively small, averaging 1.5%. Although the difference on the third day is notable, it is primarily due to the low objective value on that day and the absolute difference is only €4.3, which is negligible. Consequently, it can be inferred that the linear LFM can closely approximate the quadratic LFM with a negligible difference.

C. Evaluating the Pricing Schemes

This subsection presents the main result of empirically evaluating the efficiency of the three pricing schemes. Figure 4 illustrates the convergence of the MF-MADDPG algorithm across three LFM pricing mechanisms. The x-axis denotes the 2000 training episodes and the y-axis represents the system cost of the LFM (objective function of problem (28)), where a lower system cost corresponds to greater LFM efficiency. The figure validates the paper's hypothesis that the proposed, incentive compatible pricing scheme succeeds in incentivizing the nodes to learn bidding policies that result in a lower system cost (higher efficiency) compared to the two benchmarks.

Figure 5 provides more details to support this conclusion by presenting the system costs of the three schemes for each day, while in Table II, the advantage gained by the proposed scheme is more specifically quantified: The optimal system cost (solution of problem (17) under truthful declarations) is €2850.8, while in the presence of strategic learning nodes, the proposed scheme significantly outperforms the two benchmarks. More specifically, for both MARL algorithms the proposed scheme results in a cost that is 22.6% (and respectively 32.5%) higher than the baseline scenario, whereas the respective optimality gap for the DLMP scheme is 55.5% (and 51.5% respectively). It is worth noting that, compared to the DLMP scheme, the proposed scheme reduces the system cost by $15-23\%$.

The PAB scheme achieves the worst performance, with a system cost that is 227.5% (and 164.8%) higher than the optimal baseline. Moreover, Fig. 4 demonstrates that, in contrast to the proposed and the DLMP schemes, the nodes' learning policies under the PAB scheme are volatile and not convergent.

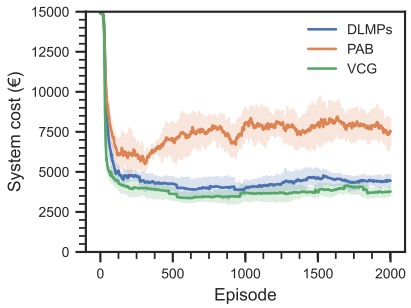


Fig. 4: Convergence of system cost with MF-MADDPG

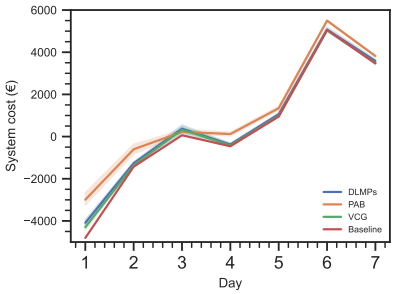


Fig. 5: System cost of different LFM pricing mechanisms

This is explained by the fact that the PAB scheme makes the discovery of a good bidding policy a very difficult and highly non-stationary task for a node, as pointed out in Section IV-A.

Overall, the results provide an empirical demonstration of the proposed scheme's superiority towards enhancing the efficiency of the LFM in the presence of strategic, learning nodes.

D. Assessing the Payments

In this subsection, we provide further insight into the implications of each pricing scheme in terms of the flexibility cost incurred for the DSO, i.e. the aggregated payments $\sum_{n\in\mathcal{N}} r_n$ that the DSO makes to the nodes. Figure 6 shows how the flexibility cost of the DSO evolves across learning episodes. At the beginning of the episodes, the FSPs submit bids with high flexibility prices, resulting in high flexibility costs. Nevertheless, the competition among FSPs leads to a gradual reduction in flexibility bidding prices, resulting in decreased flexibility costs as the training episodes increase. The PAB scheme incurs the lowest cost to the DSO. This was expected since the PAB scheme rewards a node's activated flexibility blocks at different prices, while the DLMP scheme rewards the whole of the node's flexibility at the price of the marginal (i.e. most expensive) accepted block. Moreover, the proposed scheme results in a flexibility cost that is higher than the one of the DLMP. This is exactly what the theory predicts for the equilibrium strategies (i.e. for the case where nodes can compute the payoff-maximizing bids by problem (21)), and Fig. 6 validates that this expectation propagates also to the case of nodes that learn to approximate their payoff-maximizing strategies.

A more quantitative view is provided in Table II, where the flexibility costs across three pricing mechanisms are compared in both baseline and strategic bidding scenarios. In the baseline scenario, the PAB demonstrates the lowest flexibility

TABLE II: Comparison of different LFM pricing mechanisms

Pricing mechanism		DLMPs			PAB			VCG		
Bidding method		Baseline	Strategic	Gap	Baseline	Strategic	Gap	Baseline	Strategic	Gap
System cost (€)	MADDPG	2850.8	4556.4	55.5%	2850.8	9336.8	227.5%	2850.8	3495.6	22.6%
	MF-MADDPG		4437.3	51.5%		7548.6	164.8%		3777.7	32.5%
Flexibility payment (€)	MADDPG	15521.7	15919.3	2.5%	2850.8	13932.9	388.7%	16973.2	17814.6	5.0%
	MF-MADDPG		16226.6	4.5%		11688.9	310.0%		17390.6	2.5%

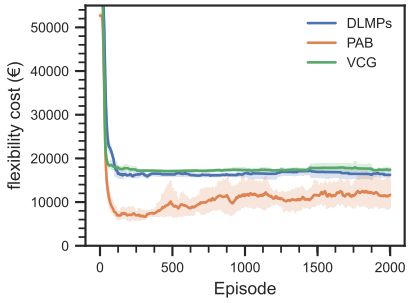


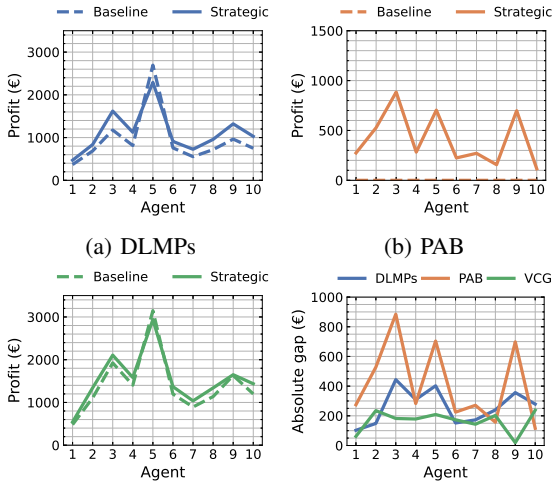
Fig. 6: Convergence of flexibility cost with MF-MADDPG

cost at €2850.8, marking an 81.6% reduction compared to DLMPs (€15521.7) and an 83.2% reduction compared to VCG (€16973.2). However, this result should be taken with a grain of sand, since the PAB scheme results in a highly non-stationary learning task for the nodes and their bidding behavior is more difficult to predict; so, it is not safe to say that this result would pertain to other settings and case studies. Indeed, Fig. 6 validates this remark by indicating that the PAB scheme exhibits the highest differentiation among different seeds (shadowed area around the curve).

Moreover, by choosing the PAB scheme, the DSO implies that the nodes are actually expected to strategize since truthful bidding would result in zero payoffs. As a result, the difference in the flexibility cost between the baseline and strategic bidding scenarios is as high as $310 - 389\%$. In contrast, the flexibility costs for DLMPs and VCG mechanisms exhibit minimal changes between the baseline and strategic scenarios.

E. Implications on the Nodes' Incentives

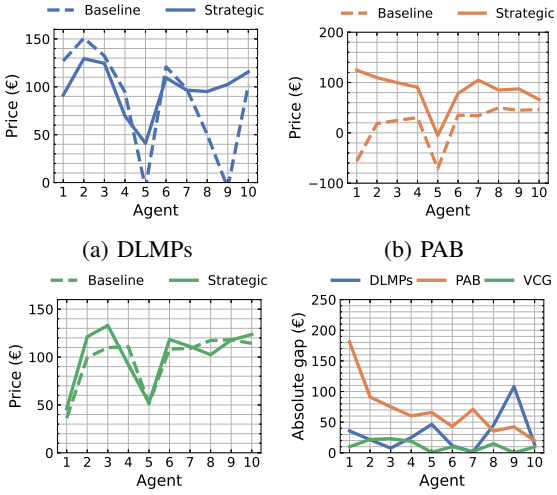
This subsection discusses the incentives of the nodes' under the different pricing schemes. To that end, Fig. 7-(a-c) show the profits of the nodes under the baseline and strategic scenarios for the three schemes, while Fig. 7-(d) summarizes the nodes' profits' gap between the baseline and the strategic scenarios. It is validated again that the PAB scheme creates the highest incentive for nodes to be untruthful since the difference in profits between baseline and strategic bidding is the highest. The proposed scheme is the one with the lowest incentive to bid untruthfully, while the DLMP scheme falls in between. Specifically, the root mean squared error (RMSE) in agent profits under the strategic bidding scenario compared to the truthful bidding scenario amounts to €283.3, €485.3, and €177.8 for the DLMPs, PAB, and VCG, respectively. Moreover, it is worth noting that, under the PAB and DLMP schemes, certain nodes (e.g. 3, 5, 9) have a significantly higher incentive to strategize than others. In contrast, under the



(a) DLMPs (b) PAB

(c) VCG (d) Absolute gap

Fig. 7: Profits of the nodes



(a) DLMPs (b) PAB

(c) VCG (d) Absolute gap

Fig. 8: Prices of the nodes

proposed scheme, the incentive to be untruthful is, not only lower overall, but also more evenly distributed among nodes.

Similarly, the average prices of the three pricing mechanisms and their absolute gaps are depicted in Fig. 8. The proposed scheme results in per-unit prices that closely track the ones that would occur under the baseline scenario, in contrast to the PAB and DLMP schemes, where strategic behavior causes a high distortion of the prices. Specifically, in the baseline scenario, the average prices for DLMPs, PAB, and

VCG are €86.5, €15.6, and €97.5, respectively. Under the strategic scenario, these averages shift to €97.5, €84.1, and €101.8 for DLMPs, PAB, and VCG, respectively. Thereby, VCG shows the smallest increase, at 4.3%, compared to 11.0% for DLMPs and a substantial 68.5% for PAB.

Furthermore, the price distortion is, again, unequally distributed among different nodes: the RMSE in agent prices under the strategic bidding scenario, in comparison to the truthful bidding scenario, are €43.1, €80.6, and €13.7 for DLMPs, PAB, and VCG, respectively. The results clearly demonstrate that VCG maintains the most consistent prices for nodes among the three pricing mechanisms.

VII. Conclusion and Outlook

In this paper, we took on the problem of designing an efficient Local Flexibility Market (LFM) through which a DSO can procure flexibility services to resolve voltage and congestion problems of the network. We relaxed the assumption of compliant nodes and considered intelligent nodes that learn to optimize the nodes' market participation, each one with the objective of selfishly optimizing its own payoff. In order to incentivize the nodes towards learning bidding policies that enhance the efficiency of the LFM dispatch, we considered three pricing schemes: the pay-as-bid, the distribution locational marginal pricing, and the proposed scheme based on Grove's pivot rule. The nodes' behavior was modeled using two deep Reinforcement Learning algorithms, and their learning was simulated for each of the three pricing schemes, along with the respective results on the LFM efficiency.

As an overarching conclusion, the simulation results answer both questions posed in the Introduction on the affirmative:

1) Learning nodes can indeed discover strategies to manipulate the two popular LFM schemes (pay-as-bid and distribution locational marginal pricing) which is alarming and calls for investing more effort and research into market designs that shield the system against such dangerous phenomena;
2) The attractive theoretical properties of the incentive compatible Grove's pricing scheme *do* propagate to the case of learning nodes despite the absence of perfect information and the fact that they cannot perfectly reason out the market outcomes. In fact, the proposed scheme was able to reduce the system's cost by $15 - 23\%$ compared to the state-of-the-art DLMP scheme.

Our analysis further delves into the schemes' effects on flexibility costs, social costs, profits, and prices of the market participants. The simulation results reveal that the increased dispatch efficiency of the proposed scheme may come at the cost of higher payments to the flexible nodes. It is left for future work to study the trade-off between efficiency and flexibility payments and achieve a good balance between the two.

References

[1] X. Jin, Q. Wu, and H. Jia, "Local flexibility markets: Literature review on concepts, models and clearing methods," *Applied Energy*, vol. 261, p. 114387, 2020.

[2] E. Prat, I. Dukovska, R. Nellikkath, M. Thoma, L. Herre, and S. Chatzivasileiadis, "Network-aware flexibility requests for distribution-level flexibility markets," *IEEE Transactions on Power Systems*, 2023.

[3] A. Akbari-Dibavar, B. Mohammadi-Ivatloo, and K. Zare, "Electricity market pricing: Uniform pricing vs. pay-as-bid pricing," *Electricity Markets: New Players and Pricing Uncertainties*, pp. 19–35, 2020.

[4] 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.

[5] C. Ruiz and A. J. Conejo, "Pool strategy of a producer with endogenous formation of locational marginal prices," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1855–1866, 2009.

[6] K. Steriotis, K. Šepetanc, K. Smpoukis, N. Efthymiopoulos, P. Makris, E. Varvarigos, and H. Pandžić, "Stacked revenues maximization of distributed battery storage units via emerging flexibility markets," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 1, pp. 464–478, 2021.

[7] A. Khaksary, K. Steriotis, G. Tsaousoglou, P. Makris, N. Efthymiopoulos, and E. Varvarigos, "Electricity market equilibrium analysis for strategic demand aggregators: The value of demand flexibility portfolios' mix," in *2023 IEEE Belgrade PowerTech*. IEEE, 2023, pp. 01–06.

[8] K. Seklos, G. Tsaousoglou, K. Steriotis, N. Efthymiopoulos, P. Makris, and E. Varvarigos, "Designing a distribution level flexibility market using mechanism design and optimal power flow," in *2020 international conference on smart energy systems and technologies (SEST)*. IEEE, 2020, pp. 1–6.

[9] G. Tsaousoglou, J. S. Giraldo, P. Pinson, and N. G. Paterakis, "Mechanism design for fair and efficient dso flexibility markets," *IEEE transactions on smart grid*, vol. 12, no. 3, pp. 2249–2260, 2021.

[10] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Applied energy*, vol. 235, pp. 1072–1089, 2019.

[11] Y. Ye, D. Papadaskalopoulos, Q. Yuan, Y. Tang, and G. Strbac, "Multi-agent deep reinforcement learning for coordinated energy trading and flexibility services provision in local electricity markets," *IEEE Transactions on Smart Grid*, vol. 14, no. 2, pp. 1541–1554, 2022.

[12] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[13] A. Sanjab, Y. Mou, A. Virag, and K. Kessels, "A linear model for distributed flexibility markets and dlmps: A comparison with the socp formulation," vol. 2021, pp. 3181–3185, 2021.

[14] E. Dall'Anese, S. V. Dhople, and G. B. Giannakis, "Optimal dispatch of photovoltaic inverters in residential distribution systems," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 487–497, 2014.

[15] Z. Yang, H. Zhong, Q. Xia, A. Bose, and C. Kang, "Optimal power flow based on successive linear approximation of power flow equations," *IET Generation, Transmission & Distribution*, vol. 10, no. 14, pp. 3654–3662, 2016.

[16] G. K. Papazoglou, A. A. Forouli, E. A. Bakirtzis, P. N. Biskas, and A. G. Bakirtzis, "Day-ahead local flexibility market for active and reactive power with linearized network constraints," *Electric Power Systems Research*, vol. 212, p. 108317, 2022.

[17] A. Papavasiliou, "Analysis of distribution locational marginal prices," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4872–4882, 2017.

[18] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[19] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

[20] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 5571–5580.

[21] S. Barsali *et al.*, *Benchmark systems for network integration of renewable and distributed energy resources*, 2014.

[22] ENTSO-E, "day-ahead prices," https://transparency.entsoe.eu/, 2023, accessed: 25 October 2023.

[23] S. Meinecke, D. Sarajlić, S. R. Drauz, A. Klettke, L.-P. Lauven, C. Rehtanz, A. Moser, and M. Braun, "Simbench—a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis," *Energies*, vol. 13, no. 12, p. 3290, 2020.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer *et al.*, "Pytorch: An imperative style, high-performance deep learning library advances in neural information processing systems, vol. 32," 2019.

[25] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023, version 9.5.2. [Online]. Available: https://www.gurobi.com/documentation/9.5/refman/