



HAL
open science

Multi-Market Bidding of Virtual Power Plants with Machine Learning-Assisted Asynchronous Distributed Optimization

Claire Bizon Monroc, Lukas Stippel, Simon Camal, Georges Kariniotakis, Luca Santosuosso, Sonja Wogrin

► **To cite this version:**

Claire Bizon Monroc, Lukas Stippel, Simon Camal, Georges Kariniotakis, Luca Santosuosso, et al.. Multi-Market Bidding of Virtual Power Plants with Machine Learning-Assisted Asynchronous Distributed Optimization. 2025. <hal-05320139>

HAL Id: hal-05320139

<https://hal.science/hal-05320139v1>

Preprint submitted on 17 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Multi-Market Bidding of Virtual Power Plants with Machine Learning-Assisted Asynchronous Distributed Optimization

Claire Bizon Monroc, Lukas Stippel, Simon Camal and
Georges Kariniotakis
Center for Processes, Renewable Energy
and Energy Systems (PERSEE),
Mines Paris PSL
Sophia Antipolis, France

Luca Santosuosso and Sonja Wogrin
Institute of Electricity Economics and Energy Innovation
Graz University of Technology
Graz, Austria

Abstract—Virtual power plants aggregate distributed energy resources, such as renewables, flexible loads, and storage units, for coordinated participation in multiple electricity markets. This requires the repeated optimization of complex bidding strategies within short time windows. Distributed algorithms address this by decomposing the bidding problem; their convergence is however impeded by synchronous schemes where every update requires outputs from all agents. Alternatively, asynchronous distributed algorithms can use auto-regressive methods to predict outputs, but they remain vulnerable to errors and degrade under prolonged delays. We propose to consider the impact of predictions on the objective costs by directly minimizing a decision regret. We design 2 decision-focused learning methods, and integrate them to a distributed bidding strategy based on the alternating direction method of multipliers. Simulations under failures and delays show that our method converges faster than existing asynchronous distributed alternatives, while approaching the optimal cost of centralized optimization.

Index Terms—Asynchronous distributed computing, communication delays, virtual power plant, decision-focused learning.

I. INTRODUCTION

Virtual Power Plants (VPPs) coordinate the optimal operation of distributed energy resources, such as renewable generation, flexible loads, and storage units. They enhance grid stability [1] and enable active participation in multiple electricity markets across different time scales [2]. As updated forecasts and flexibility requirements are received from transmission system operators (TSOs), VPPs must repeatedly re-optimize their market bids within short time windows [3], while accounting for the dynamic interplay among heterogeneous assets [4].

Distributed optimization algorithms, such as the alternating direction method of multipliers (ADMM) [5], are particularly well suited to this challenge. By decomposing the large-scale bidding problem into smaller subproblems solved in

parallel by communicating agents [6], these methods provide more modularity than conventional centralized optimization approaches: they facilitate the integration of new assets without necessitating modifications to existing subproblems. They also improve scalability as the search space of centralized formulations expands exponentially with the number of agents, whereas distributed subproblems retain tractable complexity. Furthermore, their distributed structure exhibit increased resilience to cyber-attacks. Consequently, distributed optimization is widely regarded as ideally suited to meeting the needs of modern VPPs [7].

Existing distributed algorithms for optimal bidding in VPPs typically employ a *synchronous* communication scheme [8], requiring all subproblems to be solved before proceeding to the next iteration. Yet, as the number of subproblems grows, the likelihood of communication delays or failures increases, potentially resulting in inefficient use of computational resources [9]. *Asynchronous* algorithms can rather proceed to the next iteration before all agents have terminated their computation. This can be done by using the last available decisions communicated by all agents, an approach we will refer to as “latest-value” following [9]. The latest-value approach has been employed in distributed frameworks across diverse applications, including large-scale power system unit commitment [10], security-constrained optimization under stochastic wind generation [11], and peer-to-peer energy trading [12]. While this asynchronous variant of ADMM is proven to converge to the same solution as its synchronous counterpart, it is significantly slower and still requires partial synchronization to limit disparities in the agent update frequencies [13], a limitation noted in [11] for security-constrained unit commitment.

Recent applications of asynchronous ADMM to power systems have proposed to use auto-regressive models to forecast the missing outputs instead: these surrogate predictions are then plugged as inputs to the next subproblems. They have been used to solve optimal power flow [14], [15] or VPP operation problems [9]. These methods are however sensitive to the propagation of prediction errors, and lose their advan-

⁰Submitted to the 24th Power Systems Computation Conference (PSCC 2026). This work is partly carried out in the framework of the AI-NRGY project funded by France 2030 (Grant No: ANR-22-PETA-0004), and supported by the SDW (German Business Foundation)



tages when larger delays or consecutive communication issues require longer horizon prediction [15]. A VPP could then lose profit or be unable to deliver ancillary services to the grid.

Thus far, distributed algorithms for optimizing VPP bidding strategies neither account for the impact of surrogate predictions on the objective nor effectively handle long delays. They moreover require agents to predict each other's outputs, which can infringe on privacy rules protecting the VPP's households. This paper aims to bridge these research gaps with the following contributions:

- 1) We develop an ADMM-based asynchronous distributed algorithm for optimizing VPP bidding across multiple electricity markets. To enable this, we introduce a novel mechanism for handling communication delays and failures, wherein agents substitute missing subproblem outputs with decision-focused surrogate predictions. Compared to existing methods, our approach is more robust to longer delays and component failures.
- 2) We propose to follow a “smart predict-then-optimize” (SPO) [16] approach that learns predictions to minimize the objective regret, i.e. the additional cost incurred by making decisions based on delayed subproblem outputs compared to the optimal delay-free solution.
- 3) We develop two learning strategies within this approach. In the first, a neural network (NN) learns to predict an agent's subproblem outputs by minimizing its individual regret; this however necessitates explicit access to a history of other agents' decisions. The second strategy adopts a privacy-preserving federated learning scheme, wherein the VPP orchestrates agents to collectively minimize the aggregate regret. While this enhances privacy, the inter-agent communication required for ADMM convergence precludes formal privacy guarantees.

Our approach is evaluated on the case study of a VPP aggregating a wind plant, a solar plant, a battery energy storage system (BESS) and households with controllable loads.

The remainder of the paper is structured as follows: Section II formulates the multimarket multistage bidding problem and details the ADMM iterations. Section III presents our proposed learning strategy to enhance robustness against delays and failures. In Section IV, we validate the approach through realistic simulations. Finally, Section V concludes the study.

II. MULTIMARKET MULTISTAGE BIDDING PROBLEM

We consider a VPP that manages renewable producers, a storage system and households with controllable residential loads. The VPP seeks to maximize its earnings on multiple energy markets and minimize its storage cost, while meeting the energy demands of its households. Households have flexible demands represented by time-dependent utility functions, so the VPP can reallocate part of their consumption throughout the day. We start by introducing the centralized formulation of our VPP multistage optimization problem.

A. Centralized Problem Formulation

The VPP operates every day over T time periods of length τ , and participates in 5 markets: the day-ahead (DA) market, 2 Ancillary Service (AS) markets, the continuous intraday market, and a real-time balancing market with dual pricing. To focus on the impact of uncertain renewable production, we neglect price uncertainties: the VPP acts as a price-taker under known prices, where intraday values are the mean prices of energy traded on the continuous intraday market.

A day before delivery, the VPP must plan for all time periods to charge P^{ch} , discharge P^{dis} and send L_i to household i . On the DA market, it bids T production quantities $P^{\text{DA}} \in \mathbb{R}^T$ for the next day. On the frequency markets, it bids for both the automatic Frequency Restoration Reserve (aFRR) and the Frequency Containment Reserve (FCR). This involves a symmetric reserve capacity bid (R^{F}) for FCR, and two separate, asymmetric upward (R^{U}) and downward (R^{D}) capacity reserve bids for aFRR, where $R^{\text{X}} \in \mathbb{R}^T$.

Before each time period $t \in \{1, \dots, T\}$, the VPP receives updated production forecasts. It can update its battery charging strategy as well as the households' loads, and exchange energy on the continuous intraday market by bidding P^{ID} . Finally, at the beginning of time period t , the actual renewable production, the frequency restoration requirements from the TSO and the balancing market-clearing prices are revealed. The VPP then takes the decision to sell $P^+ > 0$ or buy $P^- > 0$ at balancing market clearing-prices. Formally we have, with power and energy quantities in MW and MWh respectively:

- $\lambda, \lambda^{\text{ID}}, \lambda^+$ (resp. λ^-) $\in \mathbb{R}^T$ the price for DA, intraday, and up (resp. down) regulation balancing markets
- $\lambda^{\text{RU}}, \lambda^{\text{R+}}$ (resp. $\lambda^{\text{RD}}, \lambda^{\text{R-}}$) $\in \mathbb{R}^T$ the aFRR price for reserve and activated up (resp. down) regulation.
- $\lambda^{\text{F}} \in \mathbb{R}^T$ FCR price
- M households with utility functions $\{u_{i,t}\}_i \in 1 \dots M$: $u_{i,t}(E)$ returns the value of energy E for i at period t
- $L_{i,t} \in \mathbb{R}^T$ the average loads of household i over time period t
- $L_{i,\text{min}}$ and $L_{i,\text{max}}$ min. and max. load for household i at any time period
- $E_{i,\text{min}}^{\text{h}}$ the min. total energy consumed by i
- Δ_i the max. load change between two periods
- δ^{ID} the max. load change between two stages
- $E_{\text{max}}^{\text{sto}}$ the BESS capacity and E_t^{sto} its state at t
- $E_{\text{min}}^{\text{sto}}$ the min. level above which to keep the BESS
- C^{sto} the storage cost function
- $\eta^{\text{ch}} < 1$ and $\eta^{\text{dis}} < 1$ the efficiency factors of the BESS
- $P_{\text{max}}^{\text{ch}}$ (resp. $P_{\text{max}}^{\text{dis}}$) and $P_{\text{min}}^{\text{ch}}$ (resp. $P_{\text{max}}^{\text{dis}}$) the max. and min. charging (resp. discharging) powers
- P^{W} (resp. P^{PV}) the wind power (resp. PV) production with $\widehat{P^{\text{W}}_t}$ (resp. $\widehat{P^{\text{PV}}_t}$) its forecast available at t
- $r^{\text{U}} \in [0, 1]^T$ (resp. r^{D}) the percentage of aFRR production requested upward (resp. downward)
- $\rho^{\text{U}} \in [0, 1]^T$ (resp. ρ^{D}) the percentage of FCR production requested upward (resp. downward)

Our VPP optimization problem is therefore a 3-stage problem. Stage 0 needs only be solved once at the beginning of a day for the day-ahead and reserve capacity bids. Stage 1 and 2, corresponding to the intraday and balancing markets, are repeatedly solved at every time period based on updated forecasts and observations. As the storage is completely controlled by the VPP, charging and discharging decisions can be freely updated at each optimization step. Changes in household loads however should be notified sufficiently early, so we assume they can only be updated at the intraday stage. We give centralized formulations of our 3 problems below:

1) *Stage 0 problem:* The first intermittent production forecasts $\widehat{P^{W_0}}$ and $\widehat{P^{PV_0}}$ are available at stage 0 for all time periods $t \in \{1, \dots, T\}$. Our stage 0 optimization problem is then:

$$\max_{P^{DA}, P^{ch}, P^{dis}, L, R^U, R^D} \mathcal{B}_D + \sum_{i=1}^M u_{i,t}(E_i^h) - \tau C^{sto}(P^{ch}, P^{dis}) \quad (1)$$

$$\mathcal{B}_D = \tau(\lambda P^{DA} + \lambda^{RU} R^U + \lambda^{RD} R^D + \lambda^F R^F)$$

to be solved under the following constraints:

$$\forall i \in 1, \dots, M \quad E_i^h = \tau L_i \quad (2a)$$

$$\sum_{t=1}^T \tau L_{i,t} \geq E_{i,min}^h \quad (2b)$$

$$L_{i,min} \leq L_i \leq L_{i,max} \quad (2c)$$

$$\forall t \in 1, \dots, T \quad \forall i \in 1, \dots, M \quad |L_{i,t} - L_{i,t-1}| \leq \Delta_i \tau \quad (2d)$$

$$L_{i,0} = L_{i,T} = L_{i,min} \quad (2e)$$

where (2a) defines the energy E_i^h consumed by household i during each time period, and we use the average loads L_i as the decision variables. Loads are assumed deferrable over the day, and (2b) imposes the total energy consumed over the day by household i to be above a given quantity $E_{i,min}^h$. Physical possibility constraints (2c)-(2d) bound the value of the loads at any given period and restrict the rate of change between two time-steps. Finally (2e) fixes the initial and final load values for each day. Additional constraints are then imposed by the storage system:

$$\forall t \in 1, \dots, T \quad E_t^{sto} = E_{t-1}^{sto} + \eta^{ch} \tau P_t^{ch} - \frac{1}{\eta^{dis}} \tau P_t^{dis} \quad (3a)$$

$$P_{min}^{ch} \leq P_t^{ch} \leq P_{max}^{ch} \quad P_{min}^{dis} \leq P_t^{dis} \leq P_{max}^{dis} \quad (3b)$$

$$E_{min}^{sto} \leq E_t^{sto} \leq E_{max}^{sto} \quad (3c)$$

$$E_0^{sto} = E_T^{sto} = E_{min}^{sto} \quad (3d)$$

where (3a) defines the storage state transition as a function of charging and discharging decisions P^{ch}, P^{dis} and efficiency coefficients η^{ch}, η^{dis} , (3b) represent the limits on charging and discharging power and (3c) the bounds on storage capacity. (3d) forces equality between the starting and final battery states.

The VPP must maintain the balance of the system at any

time, with the total power produced $P^{RES} = P^W + P^{PV}$ or discharged (P^{dis}) being equal to the sum of quantities consumed (L_i), charged (P^{ch}) or sold (P^{DA}). We have:

$$\widehat{P^{RES_0}} + P^{dis} + R^D = P^{DA} + \sum_{i=1}^M L_i + P^{ch} + R^U \quad (4)$$

where the forecasts available at stage 0 ($\widehat{P^{W_0}}, \widehat{P^{PV_0}}$) are estimates of future production. Predicting the reserve activated by the TSO is hard [17], so the VPP schedules for the provision of the full reserve bids:

$$0 \leq R^F \leq \widehat{out_{max}^0} - P^{DA} \quad (5a)$$

$$0 \leq R^U \leq \widehat{out_{max}^0} - P^{DA} \quad (5b)$$

$$0 \leq R^F \leq P^{DA} \quad 0 \leq R^D \leq P^{DA}$$

where

$$\widehat{out_{max}^0} = \widehat{P^{W_0}} + \widehat{P^{PV_0}} + P_{max}^{dis}$$

so that (5a) ensures that the VPP cannot commit to any day-ahead bid or upward regulation above its forecast output capacity, and (5b) that any downward regulation is not more important than its scheduled sold production. The bounds on the day-ahead bid are similarly:

$$0 \leq P^{DA} \leq \widehat{out_{max}^0} \quad (6)$$

Before time period t starts, the VPP receives the updated production forecasts $\widehat{P^{W_t}}, \widehat{P^{PV_t}}$ for the next steps $\{t, \dots, T\}$, and updates its decision variables. We formalize the stage 1 problem below:

2) *Stage 1 problem:* At stage 1, the VPP modifies charging, discharging, and load decisions to account for new production estimates, and schedules exchanges on the continuous intraday market.

$$\max_{P^{ID}, P^{IDch}, P^{IDdis}, L^I} \left[\tau \lambda^{ID} P^{ID} + \sum_{i=1}^M u_{i,t}(\tau L_{i,t}^* + \tau L_{i,t}^I) - \tau C^{sto}(P^{*ch} + P^{IDch}, P^{*dis} + P^{IDdis}) \right] \quad (7)$$

where X^* denotes the solution of variable X found for the stage 0 problem (1). The total charging and discharging decisions $P^{*ch} + P^{IDch}$ and $P^{*dis} + P^{IDdis}$, as well as the total loads $L + L^I$ are subject to the same constraints as their day-ahead equivalents eqs. (2) and (3). To not ask a too abrupt change in consumption to the households, the modification of the load for every household is bounded in (8):

$$\forall t \quad |L_{i,t}^I| \leq \delta^{ID} \quad (8)$$

The balancing constraint is then written:

$$\delta^{ID} := \widehat{P^{W_t}} + \widehat{P^{PV_t}} + P^{*dis} + P^{IDdis} + R^{*D} - (P^{*D} + P^{ID} + \sum_{i=1}^M (L_i^* + L_i^I) + P^{*ch} + P^{IDch} + R^{*U}) = 0 \quad (9)$$

and the exchanged energy P^{ID} is again limited by the expected total power output: $P^{ID} \leq \widehat{out_{max}^t}$.

Then, at time period t , the VPP observes the current productions P_t^W and P_t^{PV} and receives the restoration requirements from the TSO. We model those as percentages of the total reserve bid decided at stage 0. We therefore observe r^{U_t} , r^{D_t} , ρ^{U_t} , ρ^{D_t} , so that the final requested upward and downward regulations are $r_t^U R_t^{*U} + \rho_t^U R_t^{*F}$ and $r_t^D R_t^{*D} + \rho_t^D R_t^{*F}$. We introduce our stage 2 problem below.

3) *Stage 2 problem*: At stage 2, the VPP updates balancing exchange variables (P^+ , P^-) as well as charging and discharging decisions, while the load decision variables are now fixed. The objective becomes

$$\mathcal{B}_B = \max_{P^+, P^-, P^{\text{dis}+}, P^{\text{ch}+}} \mathcal{B}_B \left[\tau (\lambda^+ P^+ - \lambda^- P^-) - \tau C^{\text{sto}}(P^{\text{ch}, \text{tot}}, P^{\text{dis}, \text{tot}}) \right]$$

where $P^{x, \text{tot}}$ is the sum of charging/discharging decisions $P^{x, \text{tot}} = P^{*x} + P^{Ix} + P^{x+}$. As current production and restoration requirements are only observed for the current time period, balancing constraints remain the same as in stage 1 for all $k > t$. For time period t , we now have:

$$\begin{aligned} & P_t^W + P_t^{PV} + P_t^{\text{dis}, \text{tot}} + r_t^D R_t^{*D} + P_t^- + (\rho_t^D) R_t^F \\ &= P_t^{*D} + P_t^{*I} + P_t^+ + \sum_{i=1}^M (L_{i,t}^* + L_{i,t}^{*I}) \\ &+ P_t^{\text{ch}, \text{tot}} + r_t^U R_t^{*U} + \rho_t^U R_t^F \end{aligned} \quad (10)$$

Finally, the balancing market bids are positive and again constrained by the possible power output:

$$\begin{aligned} & 0 \leq P_t^+ \leq \text{out}_{\text{max}, t} \quad 0 \leq P_t^- \\ & \text{with } \text{out}_{\text{max}, t} = P_t^W + P_t^{PV} + P_{\text{max}}^{\text{dis}}. \end{aligned} \quad (11)$$

Note that at any time period t , we only need to optimize the values for next future periods $t, \dots, T-1$, while keeping the previous decisions fixed. We now lay out the decomposition of our optimization problems between the different components of the VPP.

B. Relaxation and Decomposition

The VPP must coordinate the decisions of all member agents at every time period to quickly optimize its bidding. We decompose our optimization problem using an augmented Lagrangian relaxation, and derive ADMM updates. We focus on the stage 1 problem, as stage 0 must only be solved once the day before delivery, and stage 2 only optimizes for 2 components' decisions. The derivation is however easily extended to all stages.

Our goal is to define subproblems that can be solved by the components in parallel. The only coupled constraints between the different components are the balancing constraints. As the objectives are moreover linear in the bidding decisions (P^{DA} , P^{ID} , P^+ , P^-), we use an augmented Lagrangian relaxation to derive the ADMM iterates for every time-period k and every household $i \in \{1, \dots, M\}$. With δ^{ID} the term measuring the balancing offset defined in (9), $l \geq 0$ the algorithm's time-

step, $P^{\text{ID}}(l)$, $L^{\text{I}}(l)$, $P^{\text{IDch}}(l)$, $P^{\text{IDdis}}(l)$ vectors of dimension $T-t+1$, and $u_i(L_i)$ the vector of all $u_{i,t}(L_{i,t})$:

$$L_i^{\text{I}}(l+1) = \min_{L^{\text{I}}} -u_i(\tau L_i^* + \tau L_i^{\text{I}}(l)) \quad (12a)$$

$$- \bar{\gamma} L_i^{\text{I}}(l) + \sum_{t=k}^T \frac{\mu_t}{2} \left(\delta_t^{\text{ID}}(l) + L_{i,t}^{\text{I}}(l) \right)^2$$

s.t. load constraints (8) and (2)

$$[P^{\text{Ich}}]_{P^{\text{Idis}}} (l+1) = \min_{P^{\text{Ich}}, P^{\text{Idis}}} C^{\text{sto}}([P^{\text{ch}*} + P^{\text{Ich}}(l)]) \quad (12b)$$

$$+ \bar{\gamma} (P^{\text{IDdis}}(l) - P^{\text{IDch}}(l))$$

$$+ \sum_{t=k}^T \frac{\mu_t}{2} \left(\delta_t^{\text{ID}}(l) + P_t^{\text{IDdis}}(l) - P_t^{\text{IDch}}(l) \right)^2$$

s.t. storage constraints (3)

$$\bar{\gamma}(l+1) = \bar{\gamma}(l) + \delta^{\text{ID}}(l) \quad (12c)$$

$$\forall t \in \{k, T\} \quad (12d)$$

$$P_t^{\text{ID}}(l+1) = \mu_t^{-1} [\tau \lambda_t^{\text{ID}} + \bar{\gamma}_t(l)] + \delta_t^{\text{I}}(l) + P_t^{\text{ID}}(l)$$

$$P_t^{\text{ID}}(l+1) = \min \left(\widehat{\text{out}}_{\text{max}, t}^{\text{ID}}, P_t^{\text{ID}}(l+1) \right)$$

Iterates (12) are synchronous, meaning that all agents wait for the results of all subproblems to be known at time-step l in order to proceed to time-step $l+1$. They probably converge towards to the solution of the centralized problem (7). Yet convergence can be significantly slowed down due to the delays or failures of different agents. We now introduce our new methodology to make ADMM iterates more robust to delays and failures.

III. LEARNING ASSISTANCE FOR ROBUSTNESS TO DELAYS AND FAILURES

If different agents cannot compute their decisions before the next iteration, or the connection between the VPP and the households is interrupted, subproblem outputs need to be imputed. We call these imputations surrogate predictions, and the algorithm that produces them the surrogate models. Models can be selected purely to minimize forecasting accuracy, and this has indeed been the only approach in the literature (see Section D). Yet surrogate predictions only exist to minimize a subsequent subproblem cost, a goal that does not necessarily align with minimizing accuracy. In the following, we thus propose to explicitly optimize the forecasting model based on the subsequent optimization task.

A. A smart predict and optimization approach

We follow a decision-focused or value-oriented forecasting approach, following the SPO paradigm introduced in [16]. It directly minimizes a decision loss derived from the objective cost. If this loss is differentiable, its gradients can directly be used by any gradient-based forecasting model. In this work we focus on two such machine learning (ML) models: a recurrent neural network model and a privacy-preserving tree-based regressor [18]. Privacy-related aspects are discussed later in III-C2. We first define the decision loss function, and establish its differentiability and convexity.

B. Learning problem formulation

The decision loss we minimize is the regret: the cost difference between the forecast-based decision and the optimal decision if the true subproblem outputs were available. It is therefore always non-negative. We enumerate the $M + 2$ member agents in the VPP $\{1, \dots, M + 2\}$, and write c_j the cost function of the subproblem being minimized by agent j . All cost functions are defined in (12), and we write x_l the concatenation of all outputs for time-step l . The regret for time-step $l + 1$ is then:

$$\begin{aligned} \mathcal{R}_l(\hat{x}_l, x'_l) &= c_j(\hat{a}, x'_l) - c_j(a', x'_l) \\ \hat{a} &= \arg \min_a c_j(a, \hat{x}_l) \quad a' = \arg \min_a c_j(a, x'_l) \end{aligned} \quad (13)$$

where the \hat{x}_l are the surrogate predictions produced at time-step l , and x' are subproblem outputs used as reference. A natural option is to set $x' = x$, so that we use as reference the delayed subproblem outputs computed by the agents when these become available. Yet as l and time delays increase, $x(l)$ becomes an increasingly complex function of the past decisions taken by all agents, that is of all past delays and predictions. This raises three issues. First, it makes differentiation harder. Second, it means that for any time-step l , the prediction that minimizes the regret depends on the cumulative effect of past prediction errors. Finally, it is a function of a sequence of past stochastic delays, which is sampled from a probability space growing exponentially with l . Instead, we propose a next-step prediction regret, and define x' as the sequence of subproblems outputs collected when the ADMM iterates are run without delay. We collect N such trajectories, where for each trajectory n we have a sequence of subproblem outputs of length l_n : $[x'_{n,1}, x'_{n,2}, \dots, x'_{n,l_n}]$. We seek to learn a surrogate prediction model F_θ parameterized by θ , that minimizes the average of next-step regrets:

$$\begin{aligned} \theta &= \min_{\theta} \frac{1}{N} \sum_n \sum_l \mathcal{R}_l(\hat{x}_{n,l}, x'_{n,l}) \\ \hat{x}_{n,l} &= F_\theta(x'_{n,1}, \dots, x'_{n,l-1}) \end{aligned} \quad (14)$$

Since F_θ and c^j are chosen to be differentiable, it remains to compute the gradients of \hat{a} with regard to the predicted outputs \hat{x} . This is straightforward in the case of the bidding subproblem, for which the analytical solution has already been derived in (12), and is linear in all relevant variables. All subproblems only depend on the predicted variables in the objective, not in the constraints: indeed, decomposition precisely ensured that subproblems only have local constraints (see Section II-B). Finally, all objectives are quadratic in the predictions. The gradients of the optimal solution to such quadratic problems with respect to the problem's parameters can be derived by differentiating through their optimality conditions [19]. This step can be added as an optimization layer to our ML models, allowing for end-to-end gradient-based learning.

We use a fast gradient solver developed in [20], which can differentiate through a batch of parameterized convex optimization problems in parallel. In order to enhance compu-

tation efficiency, we reformulate our optimization subproblems following the *disciplined parametrized program* introduced by [20], which does not allow the product of two parameters to appear in the objective. For the loads (12a), we write:

$$\min_{L_i^I, z} -u_{i,t} (\tau L_i^* + \tau L_i^I) - \bar{\gamma} L_i^I + \frac{1}{2} \|z\|_2^2$$

s. t. $z = d - \mu_s (L_i^* + L_i^I)$ and $d = \mu_s (\delta^{\text{ID}}(l) + L_i^I(l))$ where μ_s is the diagonal matrix with values $[\sqrt{\mu_1}, \dots, \sqrt{\mu_T}]$. In this formulation, μ_s, γ, L_i^* and d are parameters defining the subproblem which vary for every sample, while z, L^I are the problem variables. The parametrization used for F_θ then depends on the learning method. We introduce our two methods below.

C. Learning methods

We present two learning methods. *Method 1* is a neural-based approach: for every component, a NN learns to predict the next output value by minimizing the regret with gradient descent. This however forces households to keep track of other household's loads, which could be a violation of privacy. *Method 2* offers a privacy-preserving alternative, where each component only accesses its own data and the inference of all outputs is done collaboratively. This method however relies on computationally heavy encrypted aggregation. The two methods therefore represent a tradeoff between privacy and efficiency.

1) *Method 1: Decision-focused Learning with LSTM (DFL-LSTM)*: We select the LSTM architecture, a recurrent neural network now standard for time-series prediction [21]. A different model is trained for each output. Prediction is always done based on the last $K > 0$ observations, where one observation includes both the last bidding decisions and the average loads of all households. The dimension of observations is therefore independent on the number of household agents.

At inference, observations used to predict the next time-step can include previous predictions. When the true result of a subproblem becomes available after a delay inferior to K time-steps, it replaces the corresponding surrogate output in the past observations. Outputs still missing after K time-steps will not be used for subsequent predictions: K therefore serves as a threshold separating delays from failures.

2) *Method 2: Decision-focused Learning with privacy-preserving, federated gradient-boosted trees (DFL-Tree)*: We extend the multivariate federated regression trees for decision-focused learning presented in [18] to support missing data. This model requires a convex loss function provided by our regret (13). To reduce the computation cost and following [22], we approximate the Hessians with identity matrices. We minimize the sum of all subproblem regrets: the loss function is thus additively separable, allowing the calculation of the gradients independently in a privacy-preserving way. A detailed explanation of the model is given in [18]. As our tree-based approach allows for training with missing values, it can account for failures and communication issues in the inference phase. To reduce the computation cost on large

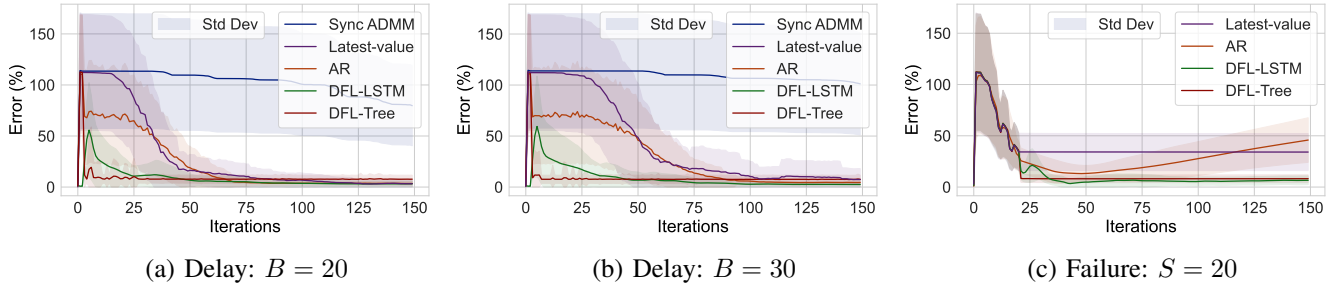


Fig. 1. ADMM with surrogate models - Uniform delays with $B = 20$ (a) and $B = 30$ (b), and failure with $S = 20$ (c) - Average and standard deviation over T time periods for Stage 1

feature dimensions, we learn a separate model for each time period, and only use as inputs the last observations.

We now use historical data to validate both DFL-LSTM and DFL-TREE in our VPP multimarket bidding use-case.

IV. EXPERIMENTS

We consider the following case-study: a VPP is aggregating a wind plant and a solar plant with a total installed capacity of 20 MW, a 5-MWh battery energy storage system and 10 households with controllable residential loads. The day is divided in $T = 96$ time periods, so that an ADMM algorithm is run every 15 minutes to compute the next intraday decisions. We download French energy prices, renewable production and balancing volumes from ENTSOE, for the period between January, 1st and July, 7th 2025. All production data until April, 30th is used to generate production forecasts for May with the PROPHET model [23]. As we do not have access to average intraday prices, we generate pseudo prices following a simple method inspired by [24]: for each time period, we simulate 20 buying and selling offers on the market, and match them to return an average price.

We now lay out our modeling choices, and then present our model training and results.

A. Modeling

We start by introducing the storage cost and household utility functions, and then present our delay profiles. The cost of storage is simply defined as a penalty on lost energy:

$$C^{sto}(P_t^{\text{ch}}, P_t^{\text{dis}}) = P_t^{\text{ch}}(1 - \eta^{\text{ch}}) + P_t^{\text{dis}} \left(\frac{1}{\eta^{\text{dis}} - 1} \right) \quad (15)$$

For each household i , we generate time periods t_i^0, t_i^1 and energy consumption levels p_i^0, p_i^1 . We assume that all inhabitants in the household leave between time periods t_i^0, t_i^1 : the household's load preferences are p_i^0 during that period and p_i^1 otherwise. The utility function $u_{i,t}$ then penalizes divergence from the preference:

$$u_{i,t}(E_{i,t}^{\text{h}}) = (L_{i,t} - \bar{P}_{i,t})^2 \quad \forall i \in \{1, \dots, M\} \quad (16)$$

$$\bar{P}_{i,t} = \begin{cases} p_i^0 & \text{if } t \in [t_i^0, t_i^1] \\ p_i^1 & \text{otherwise} \end{cases}$$

We consider 2 types of perturbations: *stochastic uniform* and *component failure*. Under uniform delays, every component solving a subproblem first samples a time delay d following

a discrete uniform law $d \sim \mathcal{U}(\{0, B\})$ where $B > 0$ is a parameter of the simulation. The results of the optimization are only shared with other components after d time-steps. Under the failure profile, a component stops returning optimization outputs after $S > 0$ time-steps.

B. Results

We use data from the 10 first days of May 2025 to collect $N = 10$ trajectories of the ADMM algorithm run without delay. This dataset is used to train both DFL-TREE and DFL-LSTM, with $K = 5$. The models are then evaluated on May, 11 under 4 scenarios: three delay scenarios, with parameter B taking values in $\{10, 20, 30\}$, and a failure scenario where the bidding component stops computing the intraday bid after $S = 20$ time-steps. The full day simulation involves 1 Stage 0, 94 Stage 1 and 96 Stage 2 optimization problems. For each case, we let ADMM run for a maximum of 1000 iterations. The surrogates are only activated for Stage 1, and we look at the evolution of the centralized cost error. We use the solution to the centralized model (7) to compute the error percentage. Our solution is compared to 3 baselines: the synchronized ADMM algorithm SYNC ADMM, the LATEST-VALUE approach, and AR the auto-regressive method introduced by [15], which is a momentum prediction model with anomaly detection. We report on Fig. 1 the cost average and standard deviation over the 96 time-periods for the first 150 iterations.

Our models learn faster and exhibit lower variance across time periods under the two delay profiles. They are also more robust to the increase in delay length from 20 to 30: in both cases DFL-LSTM reaches an error below 10% within 50 iterations, and DFL-LSTM within 35 iterations. In comparison, AR reaches that threshold in 71 iterations under $B = 20$, and then in 93 iterations for $B = 30$. After component failure, no further improvement in cost is observed with LATEST-VALUE, while AR first allows the cost to keep decreasing. Yet as no new subproblem output becomes available, its prediction errors accumulate over time-steps and eventually cause the cost to increase again. Both DFL-LSTM and DFL-TREE successfully mitigate this issue, letting iterates converge near the optimal cost. The error first decreases faster under DFL-TREE, but converges to higher values on average. We hypothesize that the collaborative learning of DFL-TREE

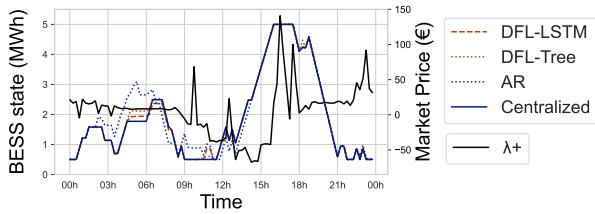


Fig. 2. Evolution of upregulation balancing market price and BESS state under 2 different surrogate models for May, 11 2025, $B = 20$

initially finds better solutions to the optimization problem, but that the recurrent network approach of DFL-LSTM better captures dependency on previous iterates.

The final centralized costs incurred by the VPP at the end of the day (96th time period) are reported in Table I. Our surrogates converge within 0.5% of the original centralized cost of -949 in all delay cases, and within 1% in the case of failure. Fig. 2 represents the evolution of the BESS state throughout the day for delay profile $B = 20$. It is compared to the BESS state under centralized optimization, and we plot the up-regulation balancing price λ^+ on the same graph. Our approach produces decisions that closely follow the centralized optimization: the VPP charges the battery between 12h and 16h when prices are negative but renewable production is maximal, and then discharges between 18h and 21h to benefit from higher prices as its available renewable production is decreasing.

	B=10	B=20	B=30	Failure
Sync	-950	-965	-972	-
Latest-value	-949	-947	-947	-922
AR	-934	-901	-867	-814
DFL-LSTM	-950	-946	-945	-939
DFL-Tree	-950	-950	-953	-948

TABLE I
VPP COSTS AFTER 96 TIME PERIODS

V. CONCLUSION

In this study, we have introduced an ADMM-based asynchronous distributed algorithm for optimizing VPP bidding across multiple electricity markets in the presence of delays and failures. This algorithm relies on a new decision-focused method, in which we learn surrogate predictions that minimize a next-step regret. We have shown that our proposed method converges within 1% of the solution to the original centralized problem, while being more robust to delays and failures than existing asynchronous distributed optimization methods. Further works could scale our approach to more households and extend the regret objective to longer horizons.

REFERENCES

[1] M. Kaiss, Y. Wan, D. Gebbran, C. U. Vila, and T. Dragičević, “Review on virtual power plants/virtual aggregators: concepts, applications, prospects and operation strategies,” *Renewable and Sustainable Energy Reviews*, vol. 211, p. 115242, 2025.

[2] L. Santosuosso, S. Camal, F. Liberati, A. Di Giorgio, A. Michiorri, and G. Kariniotakis, “Stochastic economic model predictive control for renewable energy and ancillary services trading with storage,” *Sustainable Energy, Grids and Networks*, vol. 38, p. 101373, 2024.

[3] N. Naval and J. M. Yusta, “Virtual power plant models and electricity markets—a review,” *Renewable and Sustainable Energy Reviews*, vol. 149, p. 111393, 2021.

[4] L. Santosuosso, S. Camal, A. Di Giorgio, F. Liberati, A. Michiorri, G. Bontron, and G. Kariniotakis, “Economic model predictive control for the energy management problem of a virtual power plant including resources at different voltage levels,” in *IET Conference Proceedings CP823*, vol. 2023, pp. 2044–2048, IET, 2023.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[6] L. Santosuosso, *Distributed Stochastic Optimization for Operating Complex Virtual Power Plants: Leveraging Cascaded Run-of-the-River Hydropower Flexibility for Renewable Energy Integration*. PhD thesis, Université Paris sciences et lettres, 2025.

[7] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, “A survey of distributed optimization and control algorithms for electric power systems,” *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[8] L. Santosuosso, S. Camal, A. Lett, G. Bontron, and G. Kariniotakis, “Distributed economic model predictive control for the joint energy dispatch of wind farms and run-of-the-river hydropower plants,” *Electric Power Systems Research*, vol. 235, p. 110805, 2024.

[9] Q. Wang, W. Wu, B. Wang, G. Wang, Y. Xi, H. Liu, S. Wang, and J. Zhang, “Asynchronous Decomposition Method for the Coordinated Operation of Virtual Power Plants,” *IEEE Transactions on Power Systems*, vol. 38, pp. 767–782, Jan. 2023.

[10] P. Ramanan, M. Yildirim, E. Chow, and N. Gebræel, “An Asynchronous, Decentralized Solution Framework for the Large Scale Unit Commitment Problem,” *IEEE Transactions on Power Systems*, vol. 34, pp. 3677–3686, Sept. 2019.

[11] K. Kim, M. Anitescu, and V. M. Zavala, “An Asynchronous Decomposition Algorithm for Security Constrained Unit Commitment Under Contingency Events,” in *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8, June 2018.

[12] Z. Guo, P. Pinson, Q. Wu, S. Chen, Q. Yang, and Z. Yang, “An Asynchronous Online Negotiation Mechanism for Real-Time Peer-to-Peer Electricity Markets,” *IEEE Transactions on Power Systems*, vol. 37, pp. 1868–1880, May 2022.

[13] R. Zhang and J. Kwok, “Asynchronous Distributed ADMM for Consensus Optimization,” in *Proceedings of the 31st International Conference on Machine Learning*, pp. 1701–1709, PMLR, June 2014.

[14] J. Xu, H. Sun, and C. J. Dent, “ADMM-Based Distributed OPF Problem Meets Stochastic Communication Delay,” *IEEE Transactions on Smart Grid*, vol. 10, pp. 5046–5056, Sept. 2019.

[15] A. Mohammadi and A. Kargarian, “Learning-Aided Asynchronous ADMM for Optimal Power Flow,” *IEEE Transactions on Power Systems*, vol. 37, pp. 1671–1681, May 2022.

[16] A. N. Elmachtoub and P. Grigas, “Smart “predict, then optimize,”” *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[17] S. Camal, *Forecasting and optimization of ancillary services provision by renewable energy sources*. PhD thesis, Université Paris sciences et lettres, 2020.

[18] L. Stippel, S. Camal, and G. Kariniotakis, “Multivariate federated tree-based forecasting combining resilience and privacy: Application to distributed energy resources,” *Electric Power Systems Research*, vol. 234, p. 110730, 2024.

[19] B. Amos and J. Z. Kolter, “Optnet: Differentiable optimization as a layer in neural networks,” in *International conference on machine learning*, pp. 136–145, PMLR, 2017.

[20] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” *Advances in neural information processing systems*, vol. 32, 2019.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

[22] L. Iosipoi and A. Vakhrushev, “Sketchboost: Fast gradient boosted decision tree for multioutput problems,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 25422–25435, 2022.

[23] S. J. Taylor and B. Letham, “Forecasting at scale,” *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[24] T. Carriere, *Towards seamless value-oriented forecasting and data-driven market valorisation of photovoltaic production*. PhD thesis, Université Paris sciences et lettres, 2020.