



# Online decision making for trading wind energy

Miguel Angel Muñoz<sup>1</sup> · Pierre Pinson<sup>2,3</sup> · Jalal Kazempour<sup>4</sup>

Received: 7 August 2022 / Accepted: 7 June 2023 / Published online: 28 June 2023  
© Crown 2023

## Abstract

We propose and develop a new algorithm for trading wind energy in electricity markets, within an online learning and optimization framework. In particular, we combine a component-wise adaptive variant of the gradient descent algorithm with recent advances in the feature-driven newsvendor model. This results in an online offering approach capable of leveraging data-rich environments, while adapting to the nonstationary characteristics of energy generation and electricity markets, also with a minimal computational burden. The performance of our approach is analyzed based on several numerical experiments, showing both better adaptability to nonstationary uncertain parameters and significant economic gains.

**Keywords** Decision making under uncertainty · Online learning · Electricity market · Newsvendor model

## 1 Introduction

### 1.1 Problem statement

Traditionally, the way in which trading wind energy has been considered relied on a two-step approach. These start with the predictive modeling of future energy

---

✉ Pierre Pinson  
p.pinson@imperial.ac.uk

Miguel Angel Muñoz  
miguelangeljmd@uma.es

Jalal Kazempour  
jalal@dtu.dk

<sup>1</sup> OASYS Group, University of Malaga, Malaga, Spain

<sup>2</sup> Dyson School of Design Engineering, Imperial College London, London, UK

<sup>3</sup> Department of Technology, Management and Economics, Technical University of Denmark, Kgs. Lyngby, Denmark

<sup>4</sup> Department of Wind and Energy Systems, Technical University of Denmark, Kgs. Lyngby, Denmark

generation (within either deterministic or probabilistic frameworks). Such forecasts are subsequently used as input to expected utility maximization strategies or, alternatively, some more general forms of optimization problems, e.g., within a stochastic framework and accommodating risk aversion. Although fruitful, these methodologies may be computationally expensive. As a representative recent example, for a scenario-based stochastic optimization setup to offer in electricity markets, Kraft et al (2023) mentions that computational costs may reach 3 h for a single trading instance. In addition, the value of the final decisions is highly affected by the quality of the forecasts employed. This fact was looked at for the general case of newsvendor problems (which are the type of stochastic optimization problems at hand here) by Maggioni et al (2019), while a detailed investigation of the impact of forecast quality on optimization in electricity markets (though, not exactly for market participation problems), was detailed in Ordoudis and Pinson (2016). As a consequence, it may be beneficial to integrate the forecasting and decision-making steps, within a so-called *prescriptive analytics* framework (Bertsimas and Kallus 2019). In parallel, electricity markets are amid rapid transformations towards reducing granularity and lead times, facilitating the integration of non-dispatchable energy sources but increasing the computational and adaptability requirements of the offering algorithms.

In a data-rich and nonstationary environment, approaches relying on online learning and online convex optimization are of direct relevance. For a very complete introduction to these topics, the reader is referred to Shalev-Shwartz et al (2012). On the one hand, online learning algorithms free the decision-maker from most assumptions about the wind or market dynamics, since it does not require specific probabilistic forecasts or models about such dynamics. This is more generally the case for a broad range of prescriptive analytics approaches that bypass the forecasting step. On the other hand, online learning algorithms are typically efficient methods capable of adapting to the increasing computational needs (as will be illustrated by the numerical case study in this paper). Furthermore, the online learning analysis is based on regret as opposed to the classical maximization of the expected utility, possibly allowing to derive additional insights into the properties of trading strategies.

## 1.2 Status quo with trading wind energy and underlying newsvendor problems

Most wind energy is traded in wholesale electricity markets (referred to as forward markets in this paper), where an offer is submitted prior to the actual delivery of energy. However, the stochastic nature of wind energy entails incurring deviations from the original offer. There are countless ways of approaching this problem depending on the market structure and how uncertainty is accommodated, and therefore, it is infeasible to fully address such a vast literature. However, let us provide an overview in the following. As a starting point, and since there is no single authoritative review that covers this topic of renewable energy offering in electricity markets, we refer the reader to Morales et al (2014), where the authors study different market variants and strategies assuming a classical stochastic programming framework, as well as Conejo et al (2010), which introduces general concepts of decision-making

under uncertainty within electricity markets. We deal, in particular, with markets with a dual-price settlement for imbalances, under which there is no possibility of benefiting from a deviation and where imbalance penalties are asymmetric.

Early works in this area proposed an optimal quantile strategy based on probabilistic forecasts for wind energy production (Bremnes 2004). Specifically, Pinson et al (2007) showed that, in its simplest version of a risk-neutral wind farm without any other assets (e.g., storage, conventional generation), the offering problem necessarily takes the form of a newsvendor problem. Various generalizations were explored by others. Zugno et al (2013a) proposed constraining the offer in both power and probability spaces in order to accommodate risk aversion and behavioral aspects of trading (e.g., anchoring effects towards traditional single-valued forecasts). In parallel, Mazzi and Pinson (2016) devised and tested a reinforcement learning algorithm to track the optimal quantile in a nonstationary environment. Similarly, Dent et al (2011) revisited the problem by accounting for the possibility of a population-based price-making behavior. And, for more complex versions of the offering problems, one can revert to a stochastic programming setup (Morales et al 2010), for instance, owing to inter-temporal constraints, or risk-aversion. If generally considering market offering problems where renewable energy producers are not price-takers (i.e., their decision can then affect market outcomes), Baringo and Conejo (2013), as well as Zugno et al (2013b), have proposed approaches based on bilevel optimization. Recently, Kakhbod et al (2021) have investigated the population effect of renewable energy producers and how this affects their offering strategies. Even though these varied approaches explore alternative angles to generalizing the underlying newsvendor problems in wind energy offering in electricity markets, they still require a two-step procedure (i.e., “predict, then optimize”). In contrast, a prescriptive approach does not require a forecasting step, since it directly goes from input data to decision. Consequently, there is no need to describe future wind power generation and market quantities. Hence, no assumption is made about their dynamics.

Inspired by new advances in decision making under uncertainty in data-rich environments, this problem regained interest in recent years within a prescriptive analytics framework (hence, by integrating forecasting and optimization steps). As a representative example, Stratigakos et al (2022) used an ensemble of decision trees that considers the objective function to estimate the energy production. From the modeling perspective, the work of Muñoz et al (2020) is one of the closest to ours, also aligned with the new stream of research that utilizes features to produce context-specific decisions in a fully data-driven environment. They built upon recent advances with data-driven newsvendor problems (Ban and Rudin 2019), and proposed an approach that iteratively solves a linear optimization problem to update offering decisions. Although relatively inexpensive, the computation time involved may become an issue in electricity markets like the Australian NEM,<sup>1</sup> where trading and dispatching is based on 5-minute time steps and updates. Moreover, this approach seems redundant in the sense that the complete optimization problem is solved at each and every trading session, even though consecutive training sets may

<sup>1</sup> Australian National Electricity Market (NEM). See <https://aemo.com.au/>.

only differ by one or a few samples. Such pitfalls motivates our proposal to explore alternative approaches to wind energy offering in electricity markets.

### 1.3 From optimization to online learning

Instead of using optimization directly, we introduce an offering approach within an online learning paradigm. Online learning can be seen as a special case of online convex optimization (OCO – considering convex loss functions only) where, instead of tracking optimal decisions, one adaptively and recursively estimate parameters of decision rules (often also referred to as policies). Decision rules are functions that yield decisions based on values of relevant input features. For an introduction to online optimization, we refer the reader to the surveys of Shalev-Shwartz et al (2012) and Hazan et al (2016). In addition, for the case of online learning, a recent extensive textbook-like coverage is given by Orabona (2022).

Within OCO, we place emphasis on algorithms that continuously update variables based on gradients (or subgradients) of a convex objective function. Whenever new values of input features and outcomes become available, these algorithms make a step along the gradient, towards the optimum. They ideally accommodate problems for which a closed-form expression to evaluate the sub-gradient exists (and fast to compute) (Duchi et al 2011; Zheng 2011). The well-known online gradient descent approach can be traced back to Zinkevich (2003) and inspired many further developments. Among those are numerous applications within power system operation and electricity markets (Gan and Low 2016; Hauswirth et al 2017; Colombino et al 2019; Guo et al 2021; Yuan et al 2022). These methods offer long-term regret guarantees (Orabona 2022).

Within the frame of decision-making under uncertainty, the strategy followed by online gradient descent algorithms is in sharp contrast with optimization approaches. These latter approaches solve an independent optimization problem with a different training set (a batch of data) every time a decision has to be updated, e.g., the parameter of the decision rule in Muñoz et al (2020). Under convexity assumptions, an optimal solution can be found to each optimization problem, meaning that no single decision can ever achieve better performance on average in that training set. However, there is no certainty that the out-of-sample performance of such a decision enjoys the same privilege in finite sample sets. Instead, only probability guarantees can be offered even if the samples are i.i.d. (Van Parys et al 2021).

Indeed, when the underlying data generating processes are nonstationary, the out-of-sample performance can be very poor. This issue can be partly compensated by using a rolling window setting (Bashir and Lehtonen 2018) that updates the variables frequently. However, there can also be substantial changes within the training set. In that case, the performance of batch optimization approaches may be affected by old samples that do not reflect current conditions. On the contrary, online gradient descent algorithms update the parameters of decision rules through a point-wise update that involves the most recent information only, which enables capturing changes in the characteristics of the underlying data generating processes. Therefore, online gradient methods do not only offer computational advantages. They may also

outperform established approaches, e.g., using linear programming with contextual information (even if using a sliding window scheme). This is illustrated based on the toy model examples in Sect. 4, as well as the case study in Sect. 5. Their superiority eventually is in terms of both (i) better tracking of the optimal solution within a non-stationary environment, as well as (ii) an increase in market revenues.

#### 1.4 Contributions and structure

The Australian NEM is an example of the existing trend towards shortening lead times and increasing granularity in electricity markets. These developments reduce operational and forecast uncertainty, hence facilitating the integration of stochastic renewable energy sources.<sup>2</sup> At the same time, they increase computational needs and require methodologies that adapt to changes in rapid manner. To face these new challenges, we propose an algorithm that combines a feature-driven newsvendor model inspired by Ban and Rudin (2019) with a variant of the online gradient descent algorithm presented in Zeiler (2012). We conceive a case study in which we analyze an hourly forward market that closes just before the start of the next period. It relies on actual data from the Danish Transmission System Operator (TSO), Energinet,<sup>3</sup> and provide a relevant test bench to illustrate and discuss the salient features of our approach. To the best of our knowledge, this is the first paper that analyzes the problem of trading wind energy in an online learning setting. The contributions of our work are threefold:

- we develop an online offering algorithm within an online learning framework. Results show that this algorithm is computationally inexpensive and achieves substantial economic profits;
- we propose a new nonstationary regret benchmark against which we empirically compare our algorithm;
- we showcase the ability of the proposed algorithm to adapt to nonstationary scenarios through a concise illustrative example. In addition, we analyze the superior economic performance and computational efficiency of our approach based on a case study using real-world data (published by the Danish TSO, Energinet) for a period of more than five.

The remaining of the manuscript is structured as follows: Sect. 2 introduces the problem of a wind farm offering in the forward market, and for which balancing using a two-price imbalance settlement. Section 3 develops a new offering algorithm based on an adaptive gradient descent algorithm and explores several performance metrics. Section 4 is built upon two illustrative examples that investigate the behavior of an alternative online implementation and the dynamic response of this algorithm in comparison with previous rolling window approaches. Section 5 empirically analyzes the performance of our proposed algorithm in a case study

<sup>2</sup> Increasing time granularity in electricity markets, innovation landscape brief, International Renewable Energy Agency (IRENA), Report, 2019.

<sup>3</sup> See <https://energinet.dk/>.

based on real data retrieved from the Danish TSO, Energinet. Finally, conclusions and perspectives for future work are gathered in Sect. 6.

## 2 Preliminaries

### 2.1 Mathematical notations

We introduce here some of the most relevant mathematical notations used throughout the paper. These are placed into context when further describing the optimization and learning problems at hand in the following. In terms of indices and sets, we use  $j$  as an index for features and auxiliary information, while  $t$  is an index for time periods (hours in practice, or programme time units in the electricity market of interest). These time indices are gathered within 2 sets  $\mathcal{T}^{\text{in}}$  and  $\mathcal{T}^{\text{out}}$ , which are for training (in-sample) and testing (out-of-sample), respectively.

When looking at newsvendor problems and offering in electricity markets, key parameters include  $\psi_t^+$ , the marginal opportunity cost for overproduction at hour  $t$  (€/MWh), and  $\psi_t^-$ , the marginal opportunity cost for underproduction at hour  $t$  (€/MWh). These are defined based on  $\lambda^F$ ,  $\lambda^{\text{UP}}$ , and  $\lambda^{\text{DW}} \in \mathbb{R}$ , which are the forward, up-regulation and down-regulation prices, respectively. In terms of the renewable energy producer, the asset or portfolio at hand has a nominal capacity  $\bar{E}$ , also translating to a maximum offer in terms of energy in the market for each and every programme time unit (hence, we express  $\bar{E}$  in MWh eventually). The decision variable is then the energy bid  $E_t^F$  (MWh) for that time, while the amount of energy actually produced is  $E_t$  (MWh). Within our data-driven framework, that decision is based on a vector  $\mathbf{x}_t$  of auxiliary information (i.e., features), associated to a decision rule vector  $\mathbf{q}_t$ .

Finally, for the type of online learning approach described in the following, the method and resulting algorithm rely on the gradient or subgradient of the objective function at hand, which we denote by  $\mathbf{g}_t$ , as well as a dynamic learning vector  $\boldsymbol{\eta}_t$ . We write  $g_{t,j}$  and  $\eta_{t,j}$  the  $j$ th components of these vectors. The algorithm has a number of hyperparameters involved, i.e.,  $\mu$  as a forgetting factor to temporally smooth the marginal opportunity costs  $\psi_t^+$  and  $\psi_t^-$ ,  $\eta$  to control the learning rate,  $\alpha$  to smooth the discontinuity in the derivative of the pinball loss function, and  $\rho$  as a decay constant that controls the adaptation to new gradients. A strictly positive, though small, constant  $\epsilon$  is used in the definition of the dynamic learning vector  $\boldsymbol{\eta}_t$  in order to avoid dividing by 0.

### 2.2 Newsvendor problem on a rolling time-window

We first introduce the problem of a wind farm offering in a forward market, which is cleared some time before their actual production is realized. Therefore, the producer is likely to suffer deviations from her offer. These are settled *ex-post* in a real-time (balancing) market under a two-price imbalance settlement mechanism. Furthermore, the offer is assumed to be always accepted, as the marginal

operational cost of wind farms is close to zero and therefore this technology is usually prioritized for being scheduled. The eventual market revenue  $\rho \in \mathbb{R}$  of a wind farm is given by the summation of the amounts obtained in the forward ( $\rho^F$ ) and in the balancing markets ( $\rho^B$ ), i.e.,

$$\rho = \rho^F + \rho^B = \lambda^F E^F - \lambda^{UP}(E^F - E)^+ + \lambda^{DW}(E - E^F)^+, \tag{1}$$

where  $(a)^+ = \max(a, 0)$ . In addition, the unknown parameters  $\lambda^F, \lambda^{UP}$ , and  $\lambda^{DW} \in \mathbb{R}$  are the forward, up-regulation and down-regulation prices, respectively. The key decision variable for the wind farm is her offer  $E^F \in \mathbb{R}^+$  at the forward stage. Note that  $E \in \mathbb{R}^+$  denotes the actual realization of her stochastic energy production, which is obviously unknown at the forward stage. In accordance to (1), the revenue ( $\lambda^F E^F$ ) from the forward stage is then altered when the producer deviates from her offer  $E^F$ . When the production is greater than expected  $E \geq E^F$ , the producer is to sell excess energy generation  $E - E^F > 0$  at the downward regulation at price  $\lambda^{DW}$ . On the contrary, if she produces less than her forward offer  $E \leq E^F$ , the wind farm has to buy the missing energy  $E^F - E > 0$  at teh upward regulation at price  $\lambda^{UP}$ . Under a two-price imbalance settlement, one has  $\lambda^{UP} \geq \lambda^F$  and  $\lambda^{DW} \leq \lambda^F$ , with at most one of them different from  $\lambda^F$  (Morales et al 2014, Ch. 7). In accordance with the aforementioned description, let  $\psi^+, \psi^- \in \mathbb{R}^+$  denote penalties for over- or under-production as

$$\psi^+ = \lambda^F - \lambda^{DW}, \tag{2}$$

$$\psi^- = \lambda^{UP} - \lambda^F. \tag{3}$$

Using (2) and (3) and the equivalence  $E - E^F = (E - E^F)^+ - (E^F - E)^+$ , we reformulate (1) as

$$\rho = \lambda^F E - \left( \psi^+(E - E^F)^+ + \psi^-(E^F - E)^+ \right). \tag{4}$$

Note that the first term of (4) is out of the control of the price-taker wind farm, as both  $\lambda^F$  and  $E$  are uncertain parameters. Therefore, the profit-maximizing offer  $E^{F*}$  of the wind farm in the forward market can be computed by minimizing the expected deviation cost as

$$E^{F*} = \arg \min_{E^F \in [0, \bar{E}]} \mathbb{E} \left[ \psi^+(E - E^F)^+ + \psi^-(E^F - E)^+ \right], \tag{5}$$

where  $\mathbb{E}[\cdot]$  is the expectation operator. The optimization program (5) solves an instance of the very well-studied newsvendor model (Qin et al 2011). Under a price-taker scenario, i.e., when the market participant’s decision are assumed not to affect market outcomes, an analytical solution to (5) can be computed with (Bremnes 2004; Pinson et al 2007)

$$E^{F^*} = F_E^{-1} \left( \frac{\bar{\psi}^+}{\bar{\psi}^+ + \bar{\psi}^-} \right), \tag{6}$$

where  $F_E^{-1}(\cdot)$  is the cumulative distribution function (cdf) of the renewable energy production and the overline denotes the expected value of the random variable (estimated as the average over available data). The reader is referred to Maggioni et al (2019) for a discussion about the value of right distribution in newsvendor applications.

On top of the fact that the true distribution of the wind production and the optimal quotient are generally unknown, (6) suffers from another major drawback, which is its inability to directly profit of additional information that may be available, e.g., wind energy forecasts for neighboring areas, or additional information about the state of the electricity market. In fact, it is usually the case that the wind farm operator has access to a vector of *auxiliary information*, also known as *features*  $\mathbf{x} \subseteq \mathcal{X} \in \mathbb{R}^p$ , where  $p$  denotes the dimension of the feature vector. This feature vector may help explaining the behavior of the uncertain parameters in (5). As proposed by Ban and Rudin (2019), this information can be exploited in newsvendor instances assuming that the optimal offer follows a linear decision rule of the form  $E^F : \mathcal{X} \rightarrow \mathbb{R}$ ,  $E^F = \mathbf{x}^\top \mathbf{q}$  with  $\mathbf{q} \in \mathbb{R}^p$  being a decision vector that parameterizes the linear model. This decision rule can easily reproduce an intercept setting a component of the feature vector  $\mathbf{x}$  equal to one. Then, considering that a set of historical samples  $\{(E_t, \psi_t^-, \psi_t^+, \mathbf{x}_t), \forall t \in \mathcal{T}^{\text{in}}\}$  is available, we compute the best decision  $\mathbf{q}^{\text{LP}}$  for this set by solving the following linear program:

$$\mathbf{q}^{\text{LP}^*} = \arg \min_{\mathbf{q}} \frac{1}{|\mathcal{T}^{\text{in}}|} \sum_{t \in \mathcal{T}^{\text{in}}} \psi_t^+ (E_t - \mathbf{x}_t^\top \mathbf{q})^+ + \psi_t^- (\mathbf{x}_t^\top \mathbf{q} - E_t)^+ \tag{7a}$$

$$\text{s.t. } 0 \leq \mathbf{x}_t^\top \mathbf{q} \leq \bar{E}, \forall t \in \mathcal{T}^{\text{in}}, \tag{7b}$$

where  $|\cdot|$  denotes the cardinality of a set. Note that this model does not implicitly assume a price-taker scenario. In fact, correlations between penalties and wind features may be captured in systems with high wind power penetration. Although the linear structure of the mapping may seem restrictive, more complex relationships can be obtained by transforming the feature space, e.g., using a Taylor approximation (Ban and Rudin 2019) or a spline basis. Next, by defining the box projection

$$\pi(\mathbf{x}, \mathbf{q}) = \min \left( \max(0, \mathbf{x}^\top \mathbf{q}), \bar{E} \right), \tag{8}$$

the optimal offer derived from new contextual information  $\mathbf{x}_{t'}$  can be computed as  $E_{t'}^F = \pi(\mathbf{x}_{t'}, \mathbf{q}^{\text{LP}})$ . As discussed in Muñoz et al (2020), when new points are incorporated into the dataset  $\mathcal{T}^{\text{in}}$ , the problem (7) can be iteratively solved to update the value of  $\mathbf{q}^{\text{LP}}$ . In the remaining of the manuscript, we refer to this approach as LP (from Linear Programming).



### 3 Online learning in newsvendor problems

In the Online Convex Optimization (OCO) framework, a decision-maker faces an online learning problem where iterative decisions are to be made. The cost of each decision is determined by a convex loss function  $f_t : \mathbb{R}^d \rightarrow \mathbb{R}$  unknown beforehand. After a decision  $\mathbf{z}_t \in Z \subseteq \mathbb{R}^d$  is made, the decision-maker learns  $f_t$  and pays  $f_t(\mathbf{z}_t)$ . Within OCO the Online Gradient Descent (OGD) algorithm, introduced by Zinkevich (2003), has proven to be very effective and versatile (Gan and Low 2016; Narayanaswamy et al 2012; Hauswirth et al 2016; Nonhoff and Müller 2020; Wood et al 2021). Starting from an initial value, the OGD performs iterative updates  $\mathbf{z}_t$  based on (sub-)gradients of  $f_t$ , denoted as  $\mathbf{g}_t$  from hereon. The magnitude of the step is controlled through a variable learning rate  $\eta_t$ . On each round, the updated vector is forced to lie within the feasible region  $Z$  through the Euclidean projection. In the OGD we rely on just the last point learned to obtain a gradient, thus resulting in a computationally inexpensive method, especially if the gradient and projection can be computed through a closed-form expression.

The selection of the learning rate is of paramount importance. The original proposal by Zinkevich (2003) presents two main alternatives, namely, a variable and a fixed learning rate. In a dynamic environment, the classical choice  $\eta_t \in \mathbb{R}^+$ ,  $\eta_t \propto t^{-1/2}$  where  $\propto$  denotes the proportional operator, is not suitable due to the fact that  $\lim_{t \rightarrow \infty} = 0$ , reducing the ability to track changes as  $t$  increases. Alternatively, one could select a fixed value  $\eta_t = \eta$  that keeps this capacity unaltered but may lose the fast convergence that the initial high values of  $\eta_t$  provide. Regardless of the selection, both choices are scale-dependent and treat each component of the gradient vector equally. To tackle this, McMahan and Streeter (2010) and Duchi et al (2011) propose to use a component-wise adaptive rate  $\eta_t \in \mathbb{R}^p$  and  $\eta_{t,j} = \eta (\sum_{k=1}^t g_{k,j}^2)^{-1/2}$  where  $g_{t,j}$  is a component of the gradient vector  $\mathbf{g}_t = [g_{t,1}, \dots, g_{t,j}, \dots, g_{t,p}]^T$ . As in the case of  $\eta_t \propto t^{-1/2}$ , the previous expression is monotonically decreasing (component-wise), again limiting the long-term ability to learn. Aware of this limitation, Zeiler (2012) suggests an exponentially decaying average of the squared gradients to modulate the learning rate based on the most recent information. We employ this gradient descent variant to implement our algorithm in Sect. 3.1.

In the online learning community, the *de facto* metric to evaluate the performance of a series of decision vectors  $\mathbf{z}_1, \dots, \mathbf{z}_T$  is the regret  $\mathcal{R}_T \in \mathbb{R}$ . The regret provides a versatile and, in a sense, normalized metric to compare an algorithm through different problems with the advantage that little assumption is made about the oracle that generates the decisions. Traditionally, the benchmark used to compute regret is the best single action in hindsight that can be obtained as the solution to an offline optimization problem under perfect information. However, in a dynamic environment, this benchmark can be beaten easily. In Sect. 3.3 we propose an alternative benchmark more suitable for the nonstationary context of the wind energy problem.

### 3.1 Online newsvendor

In this section, we particularize the gradient descent introduced in the previous paragraphs to the context of the wind farm offering in a forward market, incorporating elements of the rolling window problem presented in Sect. 2. We name the resulting algorithm OLVN (from OnLine NewsVendor). Contrary to the rolling window approach, the OLVN algorithm updates  $\mathbf{q}$  based on the information provided by the last realization. The objective function (7a) when the set  $T^m$  reduces to one sample yields

$$NV_t(\mathbf{q}) = \psi_t^+(E_t - \mathbf{x}_t^\top \mathbf{q})^+ + \psi_t^-(\mathbf{x}_t^\top \mathbf{q} - E_t)^+. \tag{9}$$

The OLVN method requires computing a gradient of the objective function, for which we analyze two alternative procedures in the following paragraphs.

The first approach is inspired by the work of Zheng (2011) on the pinball loss, a particular case of the objective function found in newsvendor models. Since the pinball loss is not strictly differentiable, the authors propose an alternative smooth approximation to ensure that computing gradients is always possible. Note that in our case the objective function (9) is not differentiable at  $E_t = \mathbf{x}_t^\top \mathbf{q}$ . Therefore, we first propose to circumvent this issue extending the approach in Zheng (2011) to the more general expression (9) that considers arbitrary (positive) penalties as

$$NV_{t,\alpha}(\mathbf{q}) = \psi_t^+(E_t - \mathbf{x}_t^\top \mathbf{q}) + \alpha(\psi_t^+ + \psi_t^-) \log\left(1 + e^{-(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}\right), \tag{10}$$

where  $\alpha > 0$  is a parameter that controls the approximation and where higher values of this parameter result in smoother functions. The function  $NV_{t,\alpha}$  is convex in  $\mathbf{q}$  and upper bounds  $NV_t$  for any value of  $\mathbf{q}$  as proven in Propositions 1 and 2 in Appendix, respectively. Then, we derive a closed-form solution to obtain gradients of (10), yielding

$$\nabla NV_{t,\alpha}(\mathbf{q}) = \left( -\psi_t^+ + (\psi_t^+ + \psi_t^-) \frac{1}{1 + e^{(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}} \right) \mathbf{x}_t. \tag{11}$$

The second approach deals directly with the objective function as formulated in (9). Even though the original objective is not strictly differentiable, a variant of the OLVN algorithm is readily applicable to subdifferentiable functions, provided that a subgradient can be computed instead (Orabona 2022). In this case, the mapping that returns a subdifferential of (9) is given by

$$\partial NV_t(\mathbf{q}) = \begin{cases} -\psi_t^+ \mathbf{x}_t, & E_t - \mathbf{x}_t^\top \mathbf{q} > 0, \\ \psi_t^- \mathbf{x}_t, & E_t - \mathbf{x}_t^\top \mathbf{q} < 0, \\ [-\psi_t^+ \mathbf{x}_t, \psi_t^- \mathbf{x}_t], & E_t - \mathbf{x}_t^\top \mathbf{q} = 0. \end{cases} \tag{12}$$

Note that, when  $E_t - \mathbf{x}_t^\top \mathbf{q} = 0$ , any value in the interval  $[-\psi_t^+ \mathbf{x}_t, \psi_t^- \mathbf{x}_t]$  is a legitimate subgradient belonging to  $\partial NV_t(\mathbf{q})$ . For the sake of simplicity and reproducibility, the implementation of our algorithm returns zero whenever this condition is fulfilled.

Once a gradient as in (11) or a subgradient as in (12) has been computed, the key step of OLVN is to update  $\mathbf{q}_t$  using a multidimensional learning rate  $\boldsymbol{\eta}_t \in \mathbb{R}^p$  through

$$\mathbf{q}_{t+1} = \Pi(\mathbf{q}_t - \boldsymbol{\eta}_t \circ \mathbf{g}_t, \mathbf{x}_t), \tag{13}$$

where  $\circ$  denotes the element-wise product,  $\mathbf{g}_t = \nabla NV_{t,\alpha}(\mathbf{q}_t)$  or  $\mathbf{g}_t = \partial NV_t(\mathbf{q}_t)$  depending on the implementation of OLVN, and  $\Pi$  is a projection operator defined as  $\Pi : \mathbb{R}^p \times \mathcal{X} \rightarrow \mathbb{R}^p$ . Precisely,  $\Pi$  maps its arguments into the solution of the following optimization problem:

$$\Pi(\mathbf{o}, \mathbf{x}) = \arg \min_{\mathbf{q} \in Q(\mathbf{x})} \frac{1}{2} \|\mathbf{q} - \mathbf{o}\|_2, \tag{P}$$

where  $o$  represents a candidate to update the decision vector and is computed  $o = \mathbf{q}_t - \boldsymbol{\eta}_t \circ \mathbf{g}_t$ . The feasible set in (P) is defined by the set-valued mapping  $Q : \mathcal{X} \rightrightarrows \mathbb{R}^p$ ,  $Q(\mathbf{x}) = \{\mathbf{q} : 0 \leq \mathbf{x}^\top \mathbf{q} \leq \bar{E}\}$ . Note that, for any input  $\mathbf{x}$  the output of  $Q$  is a convex region bounded by two parallel hyperplanes. As the Euclidean norm is used, a unique solution is guaranteed to exist for any instance of (P). Generally, the Euclidean projection of a point into a convex set requires solving a convex optimization problem, however the definition of  $Q$  allows us to find a closed-form expression, yielding

$$\Pi(\mathbf{o}, \mathbf{x}) = \begin{cases} \mathbf{o}, & 0 \leq \mathbf{x}^\top \mathbf{o} \leq \bar{E}, \\ \mathbf{o} + \frac{\bar{E} - \mathbf{x}^\top \mathbf{o}}{\|\mathbf{x}\|_2^2} \mathbf{x}, & \mathbf{x}^\top \mathbf{o} > \bar{E}, \\ \mathbf{o} + \frac{-\mathbf{x}^\top \mathbf{o}}{\|\mathbf{x}\|_2^2} \mathbf{x}, & \mathbf{x}^\top \mathbf{o} < 0. \end{cases} \tag{14}$$

This reduces the resolution of the optimization problem (P) to evaluating the above expression. Even though the operator  $\Pi$  guarantees the feasibility of  $\mathbf{q}_t$  under the realization  $\mathbf{x}_{t-1}$ , we need to resort to (8) setting  $E_t^F = \pi(\mathbf{x}_t, \mathbf{q}_t)$  to guarantee  $E_t^F$  remains feasible for any new arbitrary  $\mathbf{x}_t$ .

The last remaining aspect is to compute the vector  $\boldsymbol{\eta}_t$  following the ideas in Zeiler (2012). Let  $\mathbf{g}_t = [g_{t,1}, \dots, g_{t,j}, \dots, g_{t,p}]^\top$  be a gradient or subgradient vector computed through (11) and (12). Then, we can define the squared running average of each component as

$$\bar{g}_{t,j}^2 = \rho \bar{g}_{t-1,j}^2 + (1 - \rho) g_{t,j}^2, \tag{15}$$

where  $\rho \in [0, 1)$  is a decay constant and  $\bar{g}_{0,j}^2 = 0$ . The auxiliary variable  $\bar{g}_{t,j}^2$  is then used to compute the independent learning rate applied to the associated decision vector component following

$$\eta_{t,j} = \frac{\eta}{\sqrt{\bar{g}_{t,j}^2 + \epsilon}}, \tag{16}$$

where  $\epsilon \in \mathbb{R}^+$  helps better conditioning the denominator (by avoiding division by 0) and  $\eta > 0$  is a constant. We use the update given by (15) and (16) in the proposed

OLNV algorithm with the values  $\epsilon = 10^{-6}$  and  $\rho = 0.95$ , as originally suggested in Zeiler (2012). The benefits of this update is twofold. On the one hand, OLNv adapts each learning rate component to the scale of the incumbent feature. On the other hand, OLNv tracks the most recent dynamic between the uncertain vector  $[E_t, \psi_t^+, \psi_t^-]$  and the feature vector  $\mathbf{x}_t$ . The OLNv algorithm for the feature-driven wind energy trading problem is compiled in Algorithm 1.

---

**Algorithm 1** Online Newsvendor (OLNV)

---

**Require:** Initial values  $\mathbf{q}_1 \in \mathbb{R}^p$ ,  $\eta > 0$ ,  $\rho \in [0, 1)$ ,  $\epsilon \in \mathbb{R}^+$

- 1: Initialize  $\bar{g}_{0,j}^2 = 0$ ,  $\forall j$
  - 2: **for**  $t = 1$  to  $T$  **do**
  - 3:   Output  $\mathbf{q}_t$
  - 4:   Receive  $\mathbf{x}_t$
  - 5:   Compute  $E_t^F = \pi(\mathbf{x}_t, \mathbf{q}_t)$
  - 6:   Receive  $NV_t$  and pay  $NV_t(E_t^F)$
  - 7:   Set  $\mathbf{g}_t = \nabla NV_{t,\alpha}(\mathbf{q}_t)$  or  $\mathbf{g}_t = \partial NV_t(\mathbf{q}_t)$
  - 8:   Accumulate  $\bar{g}_{t,j}^2 = \rho \bar{g}_{t-1,j}^2 + (1 - \rho) g_{t,j}^2$ ,  $\forall j$
  - 9:   Compute  $\eta_{t,j} = \eta (\bar{g}_{t,j}^2 + \epsilon)^{-1/2}$ ,  $\forall j$
  - 10:   Update  $\mathbf{q}_{t+1} = \Pi(\mathbf{q}_t - \boldsymbol{\eta}_t \circ \mathbf{g}_t, \mathbf{x}_t)$  solving ( $\mathcal{P}$ )
  - 11: **end for**
- 

Despite the fact we have considered a single wind farm in the derivation, the proposed OLNv algorithm is general enough to be exploited for an aggregation of wind farms, or in general, for a portfolio of diverse renewable energy sources with uncertain production, just by combining the capacity and generation of the assets. Equally, the potential spatial correlation among production of wind farms does not affect the feasible region of the newsvendor model, and therefore does not complicate the OLNv algorithm. On the contrary, adding storage to the generation portfolio forces the model to include inter-temporal constraints that dramatically reshape the feasible region, implying that the current decision will affect future outcomes. In this case, the decision-maker can resort to classical dynamic programming (Hargreaves and Hobbs 2012) or more advanced learning algorithms such as budget-constrained online learning (Liakopoulos et al 2019; Sherman and Koren 2021) or reinforcement learning algorithms (Kuznetsova et al 2013; Sutton and Barto 2018).

Finally, even if a population effect may be present for renewables in electricity markets (i.e., even if price-taker individually, the sum of individual actions of these producers may impact market outcomes), several wind power producers can effectively use the OLNv algorithm to improve the profitability of their offer within the same region. The fact that each competing producer has different contextual information available and may process it in alternative ways mitigates possible increases in the volatility of their outcomes that could arise from correlated generation.

### 3.2 Regularization through average penalty anchoring

In an electricity market with a two-price imbalance settlement scheme, it is common that  $\psi_t^+ = \psi_t^- = 0$  over a significant number of hours, meaning that load and generation are close to being balanced. In this situation, from (9), the producer experiences no cost no matter the deviation from the actual production. Moreover, the gradients computed through (9) are zero and therefore the variable vector  $\mathbf{q}_t$  is not updated, wasting information about the relationship between  $E_t^F$  and  $\mathbf{x}_t$ . And, when penalties are different from zero, they typically exhibit random behavior with sharp spikes representing highly imbalanced scenarios which, in turn, yields destabilizing updates of the vector  $\mathbf{q}_t$ . To tackle both issues, we propose performing the following convex transformation of the original penalties:

$$\psi_t^{+'} = \mu\psi_t^+ + (1 - \mu)\bar{\psi}^+, \tag{17}$$

$$\psi_t^{-'} = \mu\psi_t^- + (1 - \mu)\bar{\psi}^-, \tag{18}$$

where  $0 \leq \mu \leq 1$  and  $\bar{\psi}^+, \bar{\psi}^- \in \mathbb{R}^+$  are the historical average penalties. This convex transformation is inspired by the concept of constraining the optimal offer around the point forecast proposed by Zugno et al (2013a). In contrast though, we do not impose hard constraints on the decision vector  $\mathbf{q}_t$ . Instead, we smooth the objective function using as anchor the sample average optimal market quantile determined by the average market penalties  $\bar{\psi}^+$  and  $\bar{\psi}^-$ . To do so, we consider a convex combination of the original objective function (7a) with an additional term that minimizes such a quantile,

$$\begin{aligned} NV_t^R &= \mu\psi_t^+(E_t - \mathbf{x}_t^\top \mathbf{q})^+ + \mu\psi_t^-(\mathbf{x}_t^\top \mathbf{q} - E_t)^+ \\ &\quad + (1 - \mu)\bar{\psi}^+(E_t - \mathbf{x}_t^\top \mathbf{q})^+ + (1 - \mu)\bar{\psi}^-(\mathbf{x}_t^\top \mathbf{q} - E_t)^+. \end{aligned} \tag{19}$$

Then, using (17) and (18), the original objective structure is recovered, i.e.,

$$NV_t^R = \psi_t^{+'}(E_t - \mathbf{x}_t^\top \mathbf{q})^+ + \psi_t^{-'}(\mathbf{x}_t^\top \mathbf{q} - E_t)^+. \tag{20}$$

Therefore, by replacing  $\psi_t^+, \psi_t^-$  with  $\psi_t^{+'}, \psi_t^{-'}$  in the original objective function, we regularize the learning procedure at no extra computational cost. When the available samples are not sufficient to provide reliable estimates of the true  $\bar{\psi}^+$  and  $\bar{\psi}^-$ , the producer can resort to assume a balanced market with penalties  $\bar{\psi}^+ = \bar{\psi}^- = 1$ . Thus, with  $\mu < 1$ , provided that  $\bar{\psi}^+, \bar{\psi}^- > 0$ , the algorithm utilizes the information contained in samples with both penalties equal to zero, potentially accelerating the convergence and obtaining smoother updates through the gradient. The same reasoning applies to the smooth objective function.

### 3.3 Performance evaluation

In order to assess the economic performance of our algorithm over a set of testing samples  $\{(E_t, \psi_t^-, \psi_t^+, \mathbf{x}_t), \forall t \in \mathcal{T}^{\text{OOS}}\}$ , we use the average deviation cost. To lighten the notation, we write  $T = |\mathcal{T}^{\text{OOS}}|$ . Consider that we have obtained successive offers  $E_1^F, \dots, E_T^F$  over the test set, by using (7) and (8) or from Algorithm 1, after iteratively going through all the samples belonging to the test set  $\mathcal{T}^{\text{OOS}}$ . We then calculate the average deviation cost as

$$NV^{\text{OOS}} = \frac{1}{T} \sum_{t \in \mathcal{T}^{\text{OOS}}} \psi_t^-(E_t - E_t^F)^+ + \psi_t^+(E_t^F - E_t)^+. \tag{21}$$

The value of this metric gives limited information about how a particular method is performing. A natural benchmark is the score obtained when a forecast of the wind energy production (in the sense of minimizing the root mean square error) is directly used as an offer in the market. We refer to this method as FO (from FOrecast). Let  $NV_{\text{FO}}^{\text{OOS}}$  be the deviation cost incurred by FO. We then redefine the original metric in relative terms, i.e.,

$$NV^{\text{OOS}}(\%) = 100 \frac{NV^{\text{OOS}} - NV_{\text{FO}}^{\text{OOS}}}{NV_{\text{FO}}^{\text{OOS}}}. \tag{22}$$

Consequently, the metric expresses an improvement (as a percentage), where a value of 100% means perfect performance with zero deviation cost.

For online learning problems the customary performance measure is the regret. Traditionally, the regret compares a sequence of decision  $\mathbf{q}_1, \dots, \mathbf{q}_T$  against the best single vector in hindsight  $\mathbf{q}^{\mathcal{H}}$ . The latter is computed *ex-post* solving a problem analogous to (7) once the whole collection of samples belonging to  $\mathcal{T}^{\text{OOS}}$  is known. Let  $Q^{\mathcal{H}}$  be the intersection of all feasible sets  $Q(\mathbf{x}_t)$ , more precisely  $Q^{\mathcal{H}} : \mathcal{X} \rightrightarrows \mathbb{R}^p$ ,  $Q^{\mathcal{H}} = \{\mathbf{q} : 0 \leq \mathbf{x}_t^T \mathbf{q} \leq \bar{E}, \forall t \in \mathcal{T}^{\text{OOS}}\}$ . The *static* regret is

$$\mathcal{R}_T^s = \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{q}_t) - \min_{\mathbf{q} \in Q^{\mathcal{H}}} \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{q}). \tag{23}$$

Given the assumption of a nonstationary environment, outperforming a constant  $\mathbf{q}^{\mathcal{H}}$  can be a relatively easy task even though it is determined under perfect information. Alternatively, one may consider the *worst-case* regret (Besbes et al 2015) interchanging the sum and minimum, i.e.,

$$\mathcal{R}_T^w = \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{q}_t) - \sum_{t \in \mathcal{T}^{\text{OOS}}} \min_{\mathbf{q} \in Q(\mathbf{x}_t)} NV_t(\mathbf{q}), \tag{24}$$

where the second term of (24) gives the best individual decision  $\mathbf{q}_t^{\mathcal{H}} \in \arg \min_{\mathbf{q} \in Q(\mathbf{x}_t)} NV_t(\mathbf{q})$ . The regret computed in this way can be very pessimistic and unrealistic. Note that in the context of the wind farm, it is always possible to find a value for  $\mathbf{q}$  such that  $E_t - \mathbf{x}_t^T \mathbf{q} = 0$ , and therefore (24) readily reduces to the summation of the original objective function  $\mathcal{R}_T^w = \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{q}_t)$ . Alternatively,

Zinkevich (2003) proposed to compare the performance of online algorithms against a sequence of arbitrary decisions  $\mathbf{u}_1, \dots, \mathbf{u}_T, \mathbf{u}_t \in Q(\mathbf{x}_t)$ ,

$$\mathcal{R}_T^d = \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{q}_t) - \sum_{t \in \mathcal{T}^{\text{OOS}}} NV_t(\mathbf{u}_t). \tag{25}$$

We refer to this approach as *dynamic regret*. This formulation allows to define a metric with an adjustable difficulty between the previous benchmarks. Note that (23) and (24) are special cases of (25) with  $\mathbf{u}_t = \mathbf{q}^{\mathcal{H}t} \forall t$  and  $\mathbf{u}_t = \mathbf{q}_t^{\mathcal{H}t} \forall t$ , respectively. Then, the question is how to choose a reasonable series of reference benchmarks  $\mathbf{u}_t$  to use against OLVN. To this end, we propose dividing  $\mathcal{T}^{\text{OOS}}$  in  $k$  adjacent partitions of equal length  $l$ , except possibly the last one. Without loss of generality, by assuming  $T - kl = 0$ , we have  $\mathcal{T}_i^{\text{OOS}} = \{t : (i - 1)l + 1 \leq t \leq il\}, i = 1, \dots, k$ . Let us define the feasible sets  $Q_i^{\mathcal{H}t} = \{\mathbf{q} : 0 \leq \mathbf{x}_t^\top \mathbf{q} \leq \bar{E}, \forall t \in \mathcal{T}_i^{\text{OOS}}\}$ . Accordingly, we can compute  $\mathbf{q}_i^{\mathcal{H}t} = \arg \min_{\mathbf{q} \in Q_i^{\mathcal{H}t}} \sum_{t \in \mathcal{T}_i^{\text{OOS}}} NV_t(\mathbf{q})$ . Finally, the sequence of reference benchmarks that we propose to use in this paper is  $\mathbf{u}_t = \mathbf{q}_i^{\mathcal{H}t}, \forall t \in \mathcal{T}_i^{\text{OOS}}$ . We will empirically investigate the regret performance of OLVN in the case study presented in Sect. 5.

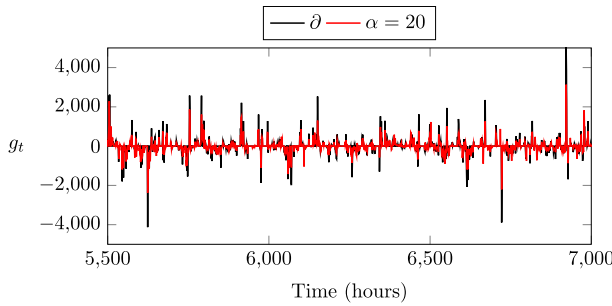
## 4 Illustrative examples

This section analyzes several illustrative examples to gain insight into the behavior of OLVN. The first case compares the two alternative implementations introduced in Sect. 3.1 and discusses their main properties. As a result of this analysis, we select the subgradient objective function as the default procedure to perform the update of  $\mathbf{q}_t$  in OLVN. One of the key features of online learning algorithms is their tracking ability, given the chronological order in which the updates are performed. In the second illustrative example, we deal with alternating penalty scenarios, showing the salient properties of OLVN to adapt to a changing environment.

### 4.1 Comparing the smooth and subgradient implementations

This illustrative example aims to elucidate whether the smooth approximation presented in (10) provides any advantage over the direct subgradient implementation of OLVN. This will allow us to determine which implementation to be used for further numerical experiments.

We consider a simplified setting with a single feature, a forecast of the wind power generation that we also use as the baseline for the FO method, and a single regressor  $q_t \in \mathbb{R}$ . No intercept is considered to ease the representation and analysis of  $q_t$ . We sample the feature from a uniform distribution  $x_t \sim U[10, 90]$  (MW) and the true wind generation series is built adding a Gaussian noise,  $E_t = x_t + \epsilon_t$  with  $\epsilon_t \sim \mathcal{N}(0, 6)$  (MW). We generate a dataset of a 1-year duration (8760 samples, as if of hourly temporal resolution). Given that the penalties  $\psi_t^+$  and  $\psi_t^-$  are difficult to simulate, we compute them based on real day-ahead and regulation prices of the



**Fig. 1** Sample of  $\partial NV_t$  and  $\nabla NV_{t,20}$  computed in the dataset of the illustrative example

**Table 1** Average absolute value  $|\bar{g}|$  and standard deviation  $\sigma$  of the (sub-) gradients and the metric  $NV^{\text{os}}(\%)$  computed for three smooth ( $\alpha$ ) and one subgradient ( $\partial$ ) implementations of the OLN

	$\partial$	$\alpha = 0.05$	$\alpha = 5$	$\alpha = 20$
$ \bar{g} $	121.7	122.0	125.6	133.5
$\sigma$	380.8	379.7	310.4	293.3
$NV^{\text{os}}(\%)$	5.3	5.2	0.8	-14.5

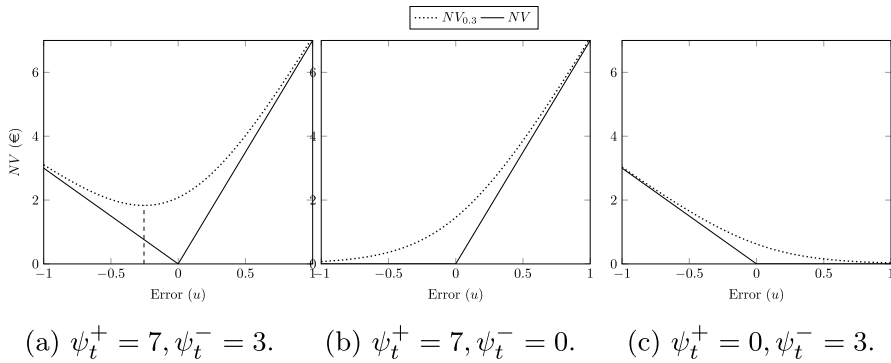
Danish DK1 bidding zone. We retrieve data corresponding to the year 2017 from the data portal of the Danish TSO, Energinet.<sup>4</sup> Four implementations of Algorithm 1 are executed, three of them computing gradients of the smooth objective function through (11) with  $\alpha = 0.05, 5$  and  $20$  and the last one using subgradients of the original cost mapping as in (12), to which we refer to as  $\partial$ . All instances are initialized with  $q_1 = 1$ , which means that the first offer produced by FO and OLN are the same. In this section we do not use any convex transformation of the prices, i.e.,  $\mu = 1$ , and we set  $\eta = 0.005$ . We run the OLN algorithm throughout the dataset, performing updates of  $q_t$  every hour.

In this section, we accompany the numerical results with some theoretical analysis. The function  $NV_{t,\alpha}(\mathbf{q})$  approximates well the original function  $NV_t(\mathbf{q})$  when  $|E_t - \mathbf{x}_t^\top \mathbf{q}| \rightarrow \infty$  as shown in Proposition 3 in Appendix. Then, an interesting point of analysis related to the behavior of both functions in the neighborhood of  $E_t - \mathbf{x}_t^\top \mathbf{q} = 0$ , defined by  $\varphi = \{\mathbf{q} : -\delta \leq E_t - \mathbf{x}_t^\top \mathbf{q} \leq \delta\}$  with  $\delta > 0$ . Let  $\mathbf{q}_1$  and  $\mathbf{q}_2$  be two vectors with  $E_t - \mathbf{x}_t^\top \mathbf{q}_1 \leq 0, E_t - \mathbf{x}_t^\top \mathbf{q}_2 \geq 0$  and  $\mathbf{q}_1, \mathbf{q}_2 \in \varphi$ . The subgradient that OLN computes for each vector changes substantially with  $\partial NV_t(\mathbf{q}_1) = \psi_t^- \mathbf{x}_t$  and  $NV_t(\mathbf{q}_2) = -\psi_t^+ \mathbf{x}_t$ , which may result in very different updates of the vector  $\mathbf{q}$  for similar values of  $\mathbf{x}_t$  or  $\mathbf{q}_t$ . Conversely,  $NV_{t,\alpha}$  is everywhere differentiable, which ensures a smooth change of  $\nabla NV_{t,\alpha}(\mathbf{q})$  for similar values of  $\mathbf{q} - T$  and  $\mathbf{x}_t$ .

Figure 1 shows a sample of  $\partial NV_t$  and  $\nabla NV_{t,20}$  that corresponds to the subgradient and gradient of the smooth objective function with  $\alpha = 20$ . Only  $NV_t$  and  $NV_{t,20}$  are

<sup>4</sup> See <https://www.energidataservice.dk/>.





**Fig. 2** Different instances of the original  $NV$  and smooth  $NV_{0.3}$  objective function with  $\alpha = 0.3$  and  $u = E_t - x_t q$

represented, for the sake of clarity. Most of the spikes in the case of  $NV_{t,20}$  are comparatively lower due to the aforementioned smoothing effect in the neighborhood of  $E_t - \mathbf{x}_t^\top \mathbf{q} = 0$ . This is aligned with the decreasing value of the standard deviation of the (sub-)gradients  $\sigma$  collated in Table 1 as  $\alpha$  increases.

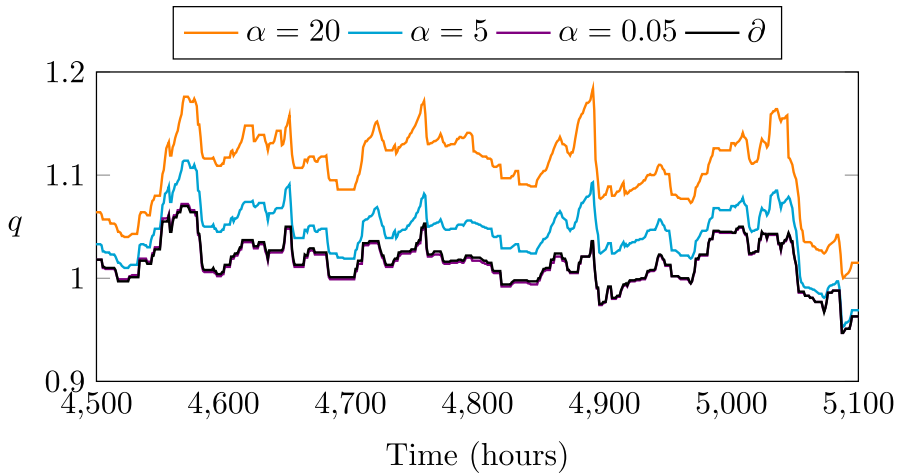
On the contrary, the mean absolute value of the (sub-)gradients, denoted as  $|\bar{g}|$ , follows the opposite evolution. To understand the rationale behind this evolution, we provide Fig. 2 showing three instances of the original and smooth losses. In all cases, we see that  $NV_{t,\alpha}$  is an upper bound for  $NV_t$  by a finite amount as expressed in Proposition 2 (with a proof available in Appendix). However, Fig. 2(a) shows that the minimum of  $NV_{t,\alpha}$  is not aligned with the minimum of the original pinball loss function. This is true whenever  $\psi_t^+ \neq \psi_t^-$  (i.e., asymmetric penalties in the market), a common situation in markets with a two-price imbalance settlement. Furthermore, when one penalty is equal to zero, the minimum is never attained.

Consequently, the gradient computed through (11) almost always introduces a deviation that is positive, compared to the true value returned by (12). The value of this error is given by the following expression:

$$\nabla NV_{t,\alpha} - \partial NV_t(\mathbf{q}) = \begin{cases} (\psi_t^+ + \psi_t^-)(1 + e^{(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha})^{-1} \mathbf{x}_t, & E_t - \mathbf{x}_t^\top \mathbf{q} > 0, \\ -(\psi_t^+ + \psi_t^-)(1 + e^{-(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha})^{-1} \mathbf{x}_t, & E_t - \mathbf{x}_t^\top \mathbf{q} < 0, \\ [-\frac{\psi_t^+ + \psi_t^-}{2} \mathbf{x}_t, \frac{\psi_t^+ + \psi_t^-}{2} \mathbf{x}_t], & E_t - \mathbf{x}_t^\top \mathbf{q} = 0. \end{cases} \quad (26)$$

The imperfect approximation of  $NV_{t,\alpha}$  distorts the magnitude and even the sign of the gradients, causing a long-term drift of  $q_t$  that increases with the smoothing parameter  $\alpha$  as shown in Fig. 3.

Finally, the last row of Table 1 presents the  $NV^{\text{OOS}}(\%)$  obtained by each implementation with respect to FO. One sees that  $NV^{\text{OOS}}(\%)$  deteriorates when  $\alpha$  increases. The smooth approach increasingly dampens the evolution of the decision vector for higher values of  $\alpha$ , but at the expense of a biased  $q_t$  and with non-negligible economic losses. Therefore, the smooth approximation does not provide any substantial



**Fig. 3** Example of the evolution of the coefficient  $q$  for different implementations of OLVN

advantage over the subgradient implementation in this application, given that the producer is neutral to risk and volatility (only being concerned with expected profits), while there is no technical constraint that encourages a smooth evolution of  $q$ . As a consequence, we will only use subgradients to implement the OLVN method throughout the remainder of the manuscript.

## 4.2 Dynamic behavior

In this illustrative example, we compare the tracking ability of OLVN and LP approaches in a nonstationary environment. Similar to the previous case, we assume that the producer has access to a unique feature and considers a model with a single regressor. Again, we sample the forecast from a uniform distribution  $x_t \sim U(10, 90)$  (MW) and the true wind power generation series is obtained by adding a normal noise  $E_t = x_t + \epsilon_t$  with  $\epsilon_t \sim \mathcal{N}(0, 6)$  (MW). Instead of the real DK1 data, we consider two possible scenarios with penalties  $\psi_t^+ = 1, \psi_t^- = 3$  and  $\psi_t^+ = 3, \psi_t^- = 1$ , alternating every two months. This process yields 8 months of data (5760 h) using the last 4 months (2880 h) as the test set. The start of the test set is aligned with the beginning of a two-month scenario with  $\psi_t^+ = 1$  and  $\psi_t^- = 3$ . The rolling window approach is implemented solving the optimization problem (7) with a set of historical samples  $\mathcal{I}^{\text{in}}$ . Then, we use (8) to cast an offer based on the context  $E_t^{\text{F}} = \pi(x_t, q_t^{\text{LP}})$ . The coefficient  $q_t^{\text{LP}}$  is refreshed every 24 h by solving problem (7), and based on a rolling window. The reason for a 24-hour update is twofold: it is aligned with the original proposal in Muñoz et al (2020) and we empirically checked that there was little economic gain to be obtained with more frequent updates. The computing time in the case of an hourly update, for example, took 24 times longer. As will be shown in the following, LP based on a rolling window approach only produces small changes over the training set, resulting in similar  $q_t^{\text{LP}}$ . We train four versions of the LP model

**Table 2** Out-of-sample  $NV^{oos}$  (%) obtained in the test set of the illustrative example

	OLNV	LP-1M	LP-2M	LP-3M	LP-4M
$NV^{oos}$ (%)	13	5	-5	-6	0

with  $|T^m| = 720, 1440, 2160$  and  $2880$  (1, 2, 3, or 4 months), denoted as LP-1 M to LP-4 M, respectively. We use the first four months of the dataset to construct the initial training sets. Although the concept of training is not strictly the same for OLNv (since it always learns on the fly, as new samples become available), only the last month of the training set is used to update the value of  $q_t$ , originally initialized with  $q_1 = 1$ , to resemble a model that has been operating for some time.

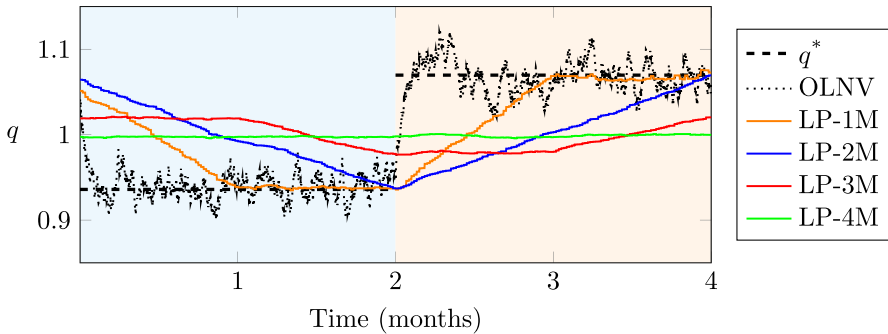
Figure 4 depicts the evolution of the single regressor  $q_t$  over the test set, together with the optimal  $q^*$  for each penalty scenario. Over the first two months, the higher value of  $\psi_t^-$  penalizes offers above the true production  $E_t^F > E_t$  and, consequently, the optimal strategy is to underestimate  $E_t^F$  with  $q^* < 1$ . Over the final months, we observe the opposite.

As one may expect, the evolution of the decision vector of LP models is smoother than in the case of OLNv, given that the former approach considers many historical samples at once to perform the update. However, Fig. 4 also shows that the trajectory of  $q_t$  produced by the rolling window models LP-1 M to LP-4 M is substantially lagged with respect to the change in the penalty scenario (emphasized by different background colors). This delay increases with the length of the training set, to the point that LP-4 M completely overlooks it. Note that the length of the training set in LP-4 M and the period of the penalty scenarios are identical. Therefore, the number of samples that penalizes under- or overproduction is equal and remains constant. As a result, LP-4 M offers no incentive to overestimate or underestimate the forecast, yielding the same value as FO (neglecting slight deviations due to the finite sample and noise).

Figure 4 additionally shows that OLNv is substantially faster at tracking the optimal  $q^*$ . In contrast, the LP problem (7) determines the decision  $q_t$  with the best performance on average in the training set, assuming that all the samples in the set are equally probable representations of future outcomes. Conversely, OLNv only uses the most recent information to perform a point-wise update that swiftly captures changes in the environment.

The tracking capability of both approaches has an impact on their economic performance. Table 2 summarizes the out-of-sample  $NV^{oos}(\%)$  obtained by each approach in the test set. In line with the previous analysis, LP-4 M obtains the same performance as FO. The other three LP methods experience decreasing  $NV^{oos}(\%)$  as the length of the training set and the lag of  $q_t$  increase. Finally, the adaptability of OLNv allows outperforming the LP approaches.

In this simplified example, we could have analyzed LP models with a shorter training set, probably resulting in reduced lag and better performances. However, in a realistic situation with a huge feature space and random penalties, months of data are typically required to capture the underlying relationships and generalize well in the out-of-sample set (Muñoz et al 2020). Therefore, the length of the training set



**Fig. 4** Evolution of  $q$  produced by five models over the test set. The blue and orange shaded periods correspond to the penalty scenarios  $\psi_t^+ = 1, \psi_t^- = 3$  and  $\psi_t^+ = 3, \psi_t^- = 1$ , respectively. The entry  $q^*$  corresponds to the best single vector for each penalty scenario

of the LP models has to be selected as a trade-off; enough data is required to learn a policy that generalizes well, but shorter sets capture dynamics better. On the contrary, the OLNv approach completely avoids this dichotomy, providing a fast and effective method that adapts to uncertain parameters generated by nonstationary environment.

## 5 Case study

Electricity markets are in the midst of a rapid development towards reducing the time between market transactions and the actual exchange of electricity. Examples of this transformation are given, i.e., by the reduction of the electricity lead time (Australian NEM or the Californian CAISO<sup>5</sup>) or by the development of new intraday markets (OMIE intraday markets or NordPool ELBAS<sup>6</sup>). Inspired by this trend, we analyze a case study that considers an online forward market that takes place every hour followed by a balancing market with a two-price imbalance settlement. The gate closure of the forward market happens just before the start of the next period. We assume that the wind farm continuously participates in the market and her offer is always accepted.

In the following we first describe the data used in this case study. Then, several benchmark methods are proposed to compare against OLNv. Finally, in a last part, we analyze the numerical results obtained, based on regret, economic performance and computational costs.

<sup>5</sup> See <https://aemo.com.au> and <http://www.caiso.com/>.

<sup>6</sup> See <https://www.omie.es/> and <https://www.nordpoolgroup.com/>.

**Table 3** Installed capacity in MW by bidding zone and technology

Years	DK1		DK2	
	Onshore	Offshore	Onshore	Offshore
2015	2966	843	608	428
2016	2966	843	608	428
2017	2966	843	608	428
2018	3664	1277	759	423
2019	3669	1277	757	423
2020	3645	1277	757	423
2021	3725	1277	756	423

**Table 4** Average RMSE (MWh) of the original forecast, the persistent (naive 1 h lag) and improved 1 h-ahead forecast computed on the out-of-sample period 07/01/2015 to 06/04/2021 with a normalized generation capacity of 100 MW

Model	DK1		DK2	
	Onshore	Offshore	Onshore	Offshore
Original	6.19	9.55	6.77	10.68
Persistent	3.36	6.39	3.90	7.49
Improved	2.72	5.70	3.34	6.66

### 5.1 Data and experimental setup

This case study is based on historical data compiled by the Danish TSO, Energinet.dk, since it includes market prices and several wind power forecasts that can be employed as input features. We collect the true and day-ahead forecast issued by Energinet for the on- and offshore wind power production of both DK1 and DK2 Danish bidding zones together with the day-ahead and regulation prices of DK1 for the period 01/07/2015 to 06/04/2021 (mm/dd/yyyy). The day-ahead spot and regulation prices are mapped into hourly penalties through equations (2) and (3) and some small negative values, obtained due to rounding errors, are filtered out.

The raw wind power forecast series are also processed to suit our needs. Given that the installed capacity of the four wind categories shown in Table 3 varies differently over the dataset, we independently normalize each series to lie between 0 and 100 MW, a figure that can easily represent the capacity of a large wind farm. According to the Danish TSO, the raw wind power forecasts are issued between 12 to 36 h ahead, although the exact time is difficult to know because no timestamp is provided. To overcome this issue, we use a standard ordinary least square regression model to produce enhanced forecasts with an accuracy comparable to an hour-ahead forecast and, therefore, suitable for our case study. We feed each raw wind power forecast into an independent linear regression model together with the last three lags of the true historical wind realization of the pertaining series. Finally, we use

the first 6 months of our dataset to independently train each of the four predictive models, one per column of Table 3.

Table 4 compares the root mean square error (RMSE) of the original and improved out-of-sample forecast against the naive benchmark provided by the first lag of each series (the wind power production of the previous hour), also known in the literature as persistence. Results show that the improved hour-ahead series significantly outperforms both original forecasts and persistence. As a byproduct, note that the wind power forecasts issued by the Danish TSO have quality metrics (e.g., RMSE) that are consistent with expectation, i.e., with offshore conditions being harder to predict than onshore conditions, while DK2 also having lower predictability since having small capacity and coverage area.

Once we have processed the wind power production series, we explain how we use them in our case study. The power generation of the wind farm offering in the market is simulated using the normalized onshore time-series of the Danish DK1 bidding zone, which is consistent with the bidding zone of the imbalance penalties utilized. The four hour-ahead forecasts of the wind power production of DK1 and DK2 are available to the producer as contextual information. Although additional wind power forecasts of neighboring bidding zones could have been used as features, we restrict ourselves to the ones produced by the Danish TSO to avoid potential inconsistencies regarding the issuing time that could cast doubt on the results obtained (Muñoz et al 2020).

Given that our goal is to reduce the imbalance cost incurred by the wind farm, we also consider several price-related features to be used as contextual information. To this end, we include the first lag of the imbalance penalties  $\psi_{t-1}^+$  and  $\psi_{t-1}^-$  in the vector of contextual information. As commented in Sect. 2, it is well known that the ratio between the penalties provides valuable information about the optimal decision of the newsvendor model and, therefore, we add the series  $r_{t-1} = \psi_{t-1}^+ / (\psi_{t-1}^+ + \psi_{t-1}^- + v)$  where  $v = 10^{-5}$  is a constant that helps better condition the denominator. Finally, we add a column of ones that enable one of the regressors to become an intercept, completing our feature set.

As a summary, let  $E_t^{on1}, E_t^{of1}, E_t^{on2}, E_t^{of2}$  denote the hour-ahead wind power forecast of DK1 onshore, DK1 offshore, DK2 onshore and DK2 offshore, respectively. Then, at the moment of delivering the offer, the producer has available a feature vector  $\mathbf{x}_t = [1, E_t^{on1}, E_t^{of1}, E_t^{on2}, E_t^{of2}, \psi_{t-1}^+, \psi_{t-1}^-, r_{t-1}]^T$  to infer the optimal offer  $E_t^F$ .

## 5.2 Benchmark methods and implementation details

In this section, we describe several benchmark methods against which we compare the performance of OLVN. The first benchmark approach is the enhanced hourly forecast of DK1 itself, produced through the ordinary least square regression model described before. Although a prediction that minimizes the RMSE may seem naive, one can expect that the deviation cost incurred by the producer vanishes as the RMSE of the forecast approaches zero. Therefore, an hour-ahead forecast is expected to perform relatively well. We also use this hour-ahead forecast as the

**Table 5** Out-of-sample  $NV^{oos}$  (%) for different combinations of parameters  $\mu$  and  $\eta_0$  over the span 07/01/2015 to 12/31/2015

$\eta$	$\mu$										
	0	0.1	0.2	0.3	0.4	0.5	0.6	<b>0.7</b>	0.8	0.9	1
$10^{-2}$	-13.8	19.2	33.7	19.2	27.7	8.4	39.7	29.2	32.3	32.3	42.0
$10^{-3}$	12.5	27.1	33.7	36.9	39.2	39.9	42.1	<b>42.2</b>	42.0	41.6	41.5
$10^{-4}$	-5.2	1.3	4.4	6.0	7.0	7.7	8.2	8.6	8.9	9.4	9.4

Highlighted in black are shown the best result and parameters selected

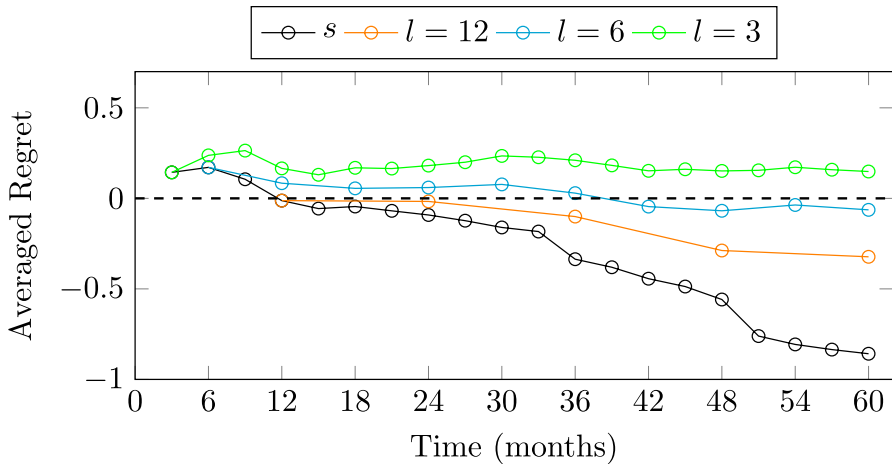
baseline to compute the metric  $NV^{oos}(\%)$  for the rest of the approaches in the way described in Sect. 3.3.

The second benchmark is that of Muñoz et al (2020), based on two-step approach using two variants of (7). In the first step, the first model only considers wind-related features plus the intercept and set  $\psi_t^+ = \psi_t^- = 1, \forall t$ . The series resulting from such model can be interpreted as an enhanced forecast of the wind energy production with a reduced mean absolute error. In a second step, this enhanced forecast is fed into (7), considering this time the true historical penalties  $\psi_t^+$  and  $\psi_t^-$  to correct for market patterns but neglecting the capacity constraint (7b). The training set is updated following a rolling window, adding new samples and eliminating the same amount of the oldest. We replicate this method, called LP2 (Linear Programming 2-steps), considering the four-hour-ahead enhanced wind power forecasts of DK1 and DK2 as the input of the first step, this is,  $\mathbf{x}_t = [1, E_t^{on1}, E_t^{of1}, E_t^{on2}, E_t^{of2}]^T$ . In line with their findings, we choose a training set of  $|T^m| = 4320$  (6 months) and a rolling window step of 24 h.

In addition, we analyze a rolling window model, called LP, that solves exactly (7) and (8) using the full vector of available contextual information. This method is the one from the illustrative example in Sect. 4.2, but with different inputs. Given the similarities with the other rolling window approach LP2, we also choose a training set length of 6 months and a rolling window step of 24 h.

Finally, we discuss a benchmark that cannot be implemented in practice, inspired by the static regret metric defined in (23). We assume perfect information about the whole out-of-sample dataset and consider (7) to compute the best linear model in hindsight, determined by the vector  $\mathbf{q}^{H}$ . Once this optimal single vector is computed, the whole sequence of offers is determined through  $E_t^F = \pi(\mathbf{x}_t, \mathbf{q}^H)$ . We name this benchmark FX (for FiXed).

Next, we discuss the implementation of OLVN in this case study. The OLVN algorithm does not need to solve an optimization problem but requires initializing two parameters. To choose  $\mu$  and  $\eta$ , we perform an offline grid search on the chunk of data spanning 07/01/2015 to 12/31/2015. As candidate values for  $\mu$  we consider  $[0, 0.1, \dots, 1]$  and for  $\eta$  we analyze  $[10^{-2}, 10^{-3}, 10^{-4}]$ . The grid search is carried out executing  $3 \times 11 = 33$  independent instances of the OLVN algorithm, initializing each time the OLVN regressor associated with the onshore DK1 forecast to 1 and the rest of the values to 0.01. The average  $NV^{oos}(\%)$  obtained by each instance is



**Fig. 5** Average dynamic regret  $R_T^d/T$  for  $l = 3, 6, 12$  months and static regret  $R_T^s/T$  updated every 3 months (denoted as  $s$ ) of the OLN method

collated in Table 5. After analyzing the results, we select the combination of values  $\mu = 0.7$  and  $\eta = 0.001$  which achieve the highest  $NV^{\text{OOS}}(\%)$ . Even though in this case study a grid search was used for the sake of clarity, other more complex cross-validation techniques (Refaeilzadeh 2009) can be used instead to select the values of  $\mu$  and  $\eta$ , including repeating this process periodically to update the values of  $\mu$  and  $\eta$  after a change in the environment.

In this case study, we assume a balanced penalty anchor  $\bar{\psi}^+ = \bar{\psi}^- = 1$ . Again, we initialize the OLN regressor associated with the onshore DK1 forecast to 1 and the rest of the values to 0.01. In other words, we start the online offering with a strategy very close to FO, mainly relying on the forecast of the wind energy production. We use the next 6 months (01/01/2016 to 06/30/2016) to update (initialize)  $\mathbf{q}_{\text{OL}}$  with the aim of having a fair comparison against LP and LP2.

The performance of all the methods presented in this section is evaluated using the dataset spanning from 07/01/2016 to 06/04/2021 (5 years with 43 200 samples). The optimization models LP, LP2, and FX are implemented with the Python package Pyomo (Bynum et al 2021) and solved through the optimization solver CPLEX,<sup>7</sup> whereas the implementation OLN is developed by the authors based on standard Python packages and uploaded to an open repository.<sup>8</sup>

### 5.3 Numerical results

Next, we discuss the results obtained in this case study. We start examining the regret suffered by OLN over the aforementioned out-of-sample dataset with a

<sup>7</sup> IBM ILOG CPLEX Optimization Studio. See <https://www.ibm.com/analytics/cplex-optimizer>.

<sup>8</sup> Experiment's code and data available at: <https://github.com/Miguel897/online-trading-wind-energy>.



**Table 6** Out-of-sample  $NV^{oos}$  (%) and execution time (s) over the span 07/01/2016 to 06/04/2021

	LP2	FX	LP	OLNV
$NV^{oos}(\%)$	3.8	24.6	31.0	38.6
Time (s)	23366	53	16077	179

length of  $D = 43,200$  hours (60 months). Let  $\mathcal{T}_j^{oos} = \cup_{i=1}^j \mathcal{T}_i^{oos}$  and recall  $\mathbf{u}_t = \mathbf{q}_t^H, \forall t \in \mathcal{T}_i^{oos}$ . We assess the average dynamic regret  $R_T^d/T$  for each sequence  $\mathcal{T}_j^{oos}, j = 1, \dots, D/l$  with partition length  $l = 2160, 4320, 8640$  hours (3, 6, 12 months). As an additional case, we compute the evolution of the static regret for a sequence  $\mathcal{T}_j^{oos}, j = 1, \dots, 20$  with a step of  $l = 2160$  hours (3 months). In each step, we refresh the best single action in hindsight as  $\mathbf{q}_j^{\mathcal{H}} = \arg \min_{\mathbf{q} \in \mathcal{Q}_j^{\mathcal{H}}} \sum_{t \in \mathcal{T}_j^{oos}} NV_t(\mathbf{q})$  and  $\mathbf{u}_t = \mathbf{q}_j^{\mathcal{H}} \forall t$ .

The four aforementioned regret series are depicted in Fig. 5. As expected, the average dynamic regret incurred by OLNv deteriorates quickly as  $l$  decreases since lower values of  $l$  translate in a more challenging benchmark closer to the worst-case regret defined in (24). Nevertheless, Fig. 5 clearly shows that OLNv achieves a sublinear static regret, i.e.,  $\lim_{T \rightarrow \infty} \sup \mathcal{R}_T^s/T \leq 0$ . This is also the case for the dynamic regret with partitions of length  $l \geq 6$  months, proving the ability of OLNv to track dynamic environments.

The economic gains obtained by each method are assessed through the  $NV^{oos}(\%)$ . The average values achieved over the evaluation dataset are collated in Table 6. First, note that all methods outperform the naive FO strategy of offering the DK1 forecast, obtaining positive values and demonstrating that this set of features contributes to reducing the deviation cost.

The LP2 method is developed in a context where recent lags in the penalties are not available. Indeed, the lack of penalty-related features translates into a modest score, showing the evident benefits of disclosing recent information in electricity markets, i.e., reducing the lead time. Even though FX determines the optimal  $\mathbf{q}^{\mathcal{H}}$  in hindsight (i.e., under perfect information), its choice is limited to a single vector for the whole horizon. The fact that several approaches perform better than FX proves the dynamic behavior of the uncertain parameters and the need for updating the decision vector. Therefore, it does not come as a surprise that LP improves the first two approaches as it relies on the full vector of features and periodically updates  $\mathbf{q}_t^{LP}$ . However, the superior adaptability of OLNv allows it to obtain the best score, achieving an additional 7.6% compared to LP and a total 38.6% deviation cost reduction compared to FO. The latter figure translates into an extra 25,930.22 €/year on average for a wind farm with a capacity of 100 MW.

Finally, the last row of Table 6 summarizes the computational time corresponding to four approaches. The FX method requires little time as it only solves a single optimization problem for the whole horizon. This contrasts with the significant amount of time required by the constant re-optimization of LP and LP2. It is noteworthy that even though OLNv produces 24 times more updates of the vector  $\mathbf{q}_t$ , the time invested is several orders of magnitude lower. In conclusion, OLNv is

up to the challenge of the electricity markets transformation achieving significant cost reduction together with exceptional computational performance.

## 6 Conclusions

This paper develops a new algorithm, named OLVN, combining a variant of the online gradient descent with recent advances that extends the newsvendor model to consider contextual information directly. The component-wise update of the learning rate enables the use of features with different scales seamlessly. In nonstationary environments, conventional stochastic approaches may consider misleading old samples in their training sets. On the contrary, our algorithm tracks the most recent information of the gradients, adapting the learning rate to follow the dynamics of the uncertain parameters and potentially obtaining higher profits. The closed-form expressions derived to compute the projection into the feasible region and a gradient of the objective function yield an efficient algorithm that can be used in computationally expensive problems. We envision the use of OLVN in future electricity markets that evolves toward continuous offering with reduced lead time. In particular, we apply this algorithm to the wind farm problem offering in an hourly forward market with a dual-price settlement for imbalances.

Several numerical experiments are carried out to assess the properties of the proposed OLVN algorithm. In the first illustrative example, we compare the behavior of two alternative implementations, namely, a subgradient approach and a smooth approximation of the original newsvendor function. The numerical and theoretical analysis provided in this example indicates that computing subgradient on the original objective function proves more profitable since it avoids update errors that may be introduced by the smooth approximation. Consequently, we determined that the subgradient implementation was the most suitable to this application and used it throughout the rest of the numerical experiments. Nevertheless, the smooth approximation could be utilized in other applications where other technical concerns advise a smooth update.

The second example shows the adaptability of the OLVN algorithm to nonstationary environments, clearly outperforming other stochastic approaches that optimize (using mathematical programming techniques) over a training set of past information. This superior performance is justified by the point-wise update that only uses the most recent information. Our case study, built upon real data of the Danish TSO Energinet, shows that OLVN achieves a 38.6% cost reduction against using a point forecast as offer and 7.6% compared to a state-of-the-art method. These significant improvements contribute to accelerating the integration of renewable energy technologies. Furthermore, we empirically analyze several dynamic definitions of regret, showing the desired sublinear convergence against most benchmarks.

Although this research focused on wind energy producers, OLVN is readily applicable to managing a portfolio of variable renewable energies with zero marginal cost, including wind, solar and other technologies. Similar algorithms can be developed when the producer's portfolio includes other assets such as loads, thermal power plants, or energy storage facilities, replacing the aggregated source

of uncertainty, i.e., the variable net energy production, by a linear decision rule. In this case, the projection step on the feasible region would likely involve solving a quadratic optimization program that can still be efficiently solved with modern solvers, when the feasible region is convex. Another attractive front is extending the OLVN algorithm to address inter-temporal constraints, observing a similar note with regard to the feasible region as in the previous case. This may require first generalizing the newsvendor framework to offering in electricity markets though.

Future work also includes delving into the theoretical guarantees that this algorithm offers in terms of regret. On a different front, a wealth of other algorithms within the field of online learning can be applied to this problem, potentially bringing additional benefits such as faster convergence rates or improved performance. Similarly, variable selection techniques could help determine the subset of the available feature streams that provide the most economic value, whereas nonlinear mapping, i.e., kernels or generalized additive models (GAMs), can extend the regression capabilities of the method. Another exciting line of research concerns the risk analysis of the producer, where other metrics can be used instead of the expected value to create risk-averse strategies.

### Appendix: Smooth function properties

This appendix provides a lemma and several propositions related to the smooth approximation  $NV_{t,\alpha}$  defined in (10). Some of the proofs in this appendix are based on the proofs provided in Zheng (2011). In this appendix we assume that  $\psi_t^+, \psi_t^- \geq 0$  and  $\psi_t^+ + \psi_t^- > 0 \forall t$ . Next, we define an auxiliary function  $S_{t,\alpha}(u), S_{t,\alpha} : \mathbb{R} \rightarrow \mathbb{R}$  with  $\alpha > 0$  as follows

$$S_{t,\alpha}(u) = \psi_t^+ u + \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{-u/\alpha}), \tag{27}$$

where  $u \in \mathbb{R}$ . We use this function in the proofs covered within this appendix. First, we prove the convexity of  $S_{t,\alpha}$ .

**Lemma 1** *For any given  $\alpha > 0$ , the function  $S_{t,\alpha}$ , defined in (27), is a convex function.*

**Proof** From the definition of  $S_{t,\alpha}$  in (27), we calculate that

$$\frac{d^2 S_{t,\alpha}(u)}{du^2} = \frac{\psi_t^+ + \psi_t^-}{\alpha} \frac{e^{-\frac{u}{\alpha}}}{(1 + e^{-\frac{u}{\alpha}})^2} > 0, \tag{28}$$

for any  $u \in \mathbb{R}$  since  $\psi_t^+ + \psi_t^- > 0$  and  $\alpha > 0$ .

We use this intermediate result to prove the convexity of  $NV_{t,\alpha}$  in the following Proposition. □

**Proposition 1** *For any given  $\alpha > 0$ , the function  $NV_{t,\alpha}$ , defined in (10), is a convex function of  $\mathbf{q}$ .*

**Proof** Let  $u = E_t - \mathbf{x}_t^\top \mathbf{q}$  in (27). Thus,

$$NV_{t,\alpha}(\mathbf{q}) = S_{t,\alpha}(E_t - \mathbf{x}_t^\top \mathbf{q}). \tag{29}$$

For  $0 \leq \omega \leq 1$  and any  $\mathbf{q}_1$  and  $\mathbf{q}_2$ , we have

$$\begin{aligned} NV_{t,\alpha}(\omega \mathbf{q}_1 + (1 - \omega)\mathbf{q}_2) &= S_{t,\alpha}(E_t - \mathbf{x}_t^\top (\omega \mathbf{q}_1 + (1 - \omega)\mathbf{q}_2)) \\ &= S_{t,\alpha}(E_t - \omega \mathbf{x}_t^\top \mathbf{q}_1 - (1 - \omega)\mathbf{x}_t^\top \mathbf{q}_2) \end{aligned} \tag{30}$$

$$\begin{aligned} &= S_{t,\alpha}(\omega(E_t - \mathbf{x}_t^\top \mathbf{q}_1) - (1 - \omega)(E_t - \mathbf{x}_t^\top \mathbf{q}_2)) \\ &\leq \omega S_{t,\alpha}(E_t - \mathbf{x}_t^\top \mathbf{q}_1) + (1 - \omega)S_{t,\alpha}(E_t - \mathbf{x}_t^\top \mathbf{q}_2), \end{aligned} \tag{31}$$

where the inequality in (31) follows from the convexity of  $S_{t,\alpha}$ , proved in Lemma 1. Then, using (A3), the above inequality renders

$$NV_{t,\alpha}(\omega \mathbf{q}_1 + (1 - \omega)\mathbf{q}_2) \leq \omega NV_{t,\alpha}(\mathbf{q}_1) + (1 - \omega)NV_{t,\alpha}(\mathbf{q}_2), \tag{32}$$

showing that  $NV_{t,\alpha}$  is a convex function on  $\mathbf{q}$ .

Next, we show that  $NV_{t,\alpha}$  asymptotically approaches  $NV_t$  for  $\alpha \rightarrow 0$ . We also show that the function  $NV_{t,\alpha}$  upper bounds  $NV_t$  for all  $q \in \mathbb{R}^p$ .  $\square$

**Proposition 2** Let  $NV_t$  and  $NV_{t,\alpha}$  be the functions defined in (9) and (10), in that order, with  $\alpha > 0$ . Then, we have

$$0 < NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) \leq \alpha(\psi_t^+ + \psi_t^-) \log 2, \tag{33}$$

for any  $\mathbf{q} \in \mathbb{R}^p$ . Thus,

$$\lim_{\alpha \rightarrow 0^+} NV_{t,\alpha}(\mathbf{q}) = NV_t(\mathbf{q}). \tag{34}$$

**Proof** When  $E_t - \mathbf{x}_t^\top \mathbf{q} \geq 0$ , we have that

$$NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) = \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{-(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}), \tag{35}$$

hence,

$$0 < NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) \leq \alpha(\psi_t^+ + \psi_t^-) \log 2, \tag{36}$$

for  $E_t - \mathbf{x}_t^\top \mathbf{q} \geq 0$ . When  $E_t - \mathbf{x}_t^\top \mathbf{q} < 0$ ,

$$\begin{aligned} NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) &= (\psi_t^+ + \psi_t^-)(E_t - \mathbf{x}_t^\top \mathbf{q}) \\ &\quad + \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{-(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}) \end{aligned} \tag{37}$$

$$= \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}). \tag{38}$$

While

$$\begin{aligned}
 0 &< \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{(E_t - \mathbf{x}_t^\top \mathbf{q})/\alpha}) \\
 &< \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{0/\alpha}) = \alpha(\psi_t^+ + \psi_t^-) \log 2,
 \end{aligned}
 \tag{38}$$

since  $E_t - \mathbf{x}_t^\top \mathbf{q} < 0$ . This shows that  $NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q})$  also falls in the range  $(0, \alpha(\psi_t^+ + \psi_t^-) \log 2)$  for  $E_t - \mathbf{x}_t^\top \mathbf{q} < 0$ . Thus, (33) is proved. Equation (34) follows directly by letting  $\alpha \rightarrow 0^+$  in (33).

Finally, we show that for high values of  $|E_t - \mathbf{x}_t^\top \mathbf{q}|$  the function  $NV_{t,\alpha}$  asymptotically approximate  $NV_t$ . □

**Proposition 3** *Let  $NV_t$  and  $NV_{t,\alpha}$  be the functions defined in (9) and (10), in that order, with  $\alpha > 0$ . Then, when  $|E_t - \mathbf{x}_t^\top \mathbf{q}| \rightarrow \infty$  we have that  $NV_{t,\alpha} - NV_t \rightarrow 0$ .*

**Proof** For the sake of a clearer exposition we define  $\mu(\mathbf{q}) = E_t - \mathbf{x}_t^\top \mathbf{q}$ , where  $\mu : \mathbb{R}^p \rightarrow \mathbb{R}$ . When  $\mu(\mathbf{q}) \rightarrow +\infty$ , we have that

$$\lim_{\mu(\mathbf{q}) \rightarrow +\infty} NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) = \lim_{\mu(\mathbf{q}) \rightarrow +\infty} \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{-\mu(\mathbf{q})/\alpha}) = 0 \tag{40}$$

When  $\mu(\mathbf{q}) \rightarrow -\infty$ , and using (38), we have that

$$\lim_{\mu(\mathbf{q}) \rightarrow -\infty} NV_{t,\alpha}(\mathbf{q}) - NV_t(\mathbf{q}) = \lim_{\mu(\mathbf{q}) \rightarrow -\infty} \alpha(\psi_t^+ + \psi_t^-) \log(1 + e^{\mu(\mathbf{q})/\alpha}) = 0 \tag{41}$$

Combining both cases, this proposition is proved. □

**Acknowledgements** M. Á. Muñoz is funded by the Spanish Ministry of Science, Innovation and Universities through the State Training Subprogram 2018 of the State Program for the Promotion of Talent and its Employability in R & D & I, within the framework of the State Plan for Scientific and Technical Research and Innovation 2017-2020 and by the European Social Fund. P. Pinson and J. Kazempour are partly supported through the Smart4RES project (European Union’s Horizon 2020, No. 864337). The sole responsibility of this publication lies with the authors. The European Union is not responsible for any use that may be made of the information contained therein.

**Author contribution** MAM Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing—Original Draft, Writing - Review & Editing, Visualization PP Conceptualization, Methodology, Validation, Investigation, Writing—Original Draft, Writing - Review & Editing, Supervision, Funding acquisition JK Conceptualization, Methodology, Validation, Investigation, Writing - Original Draft, Writing - Review & Editing, Supervision, Funding acquisition

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Ban GY, Rudin C (2019) The big data newsvendor: practical insights from machine learning. *Operat Res* 67(1):90–108
- Baringo L, Conejo AJ (2013) Strategic offering for a wind power producer. *IEEE Trans Power Sys* 28(4):4645–4654
- Bashir AA, Lehtonen M (2018) Day-ahead rolling window optimization of islanded microgrid with uncertainty. In: 2018 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe), IEEE, pp 1–6
- Bertsimas D, Kallus N (2019) From predictive to prescriptive analytics. *Manag Sci* 66(3):1025–1044
- Besbes O, Gur Y, Zeevi A (2015) Non-stationary stochastic optimization. *Operat Res* 63(5):1227–1244
- Bremnes JB (2004) Probabilistic wind power forecasts using local quantile regression. *Wind Energy* 7(1):47–54
- Bynum ML, Hackebeitl GA, Hart WE et al (2021) *Pyomo-optimization modeling in Python*, vol 67, 3rd edn. Springer, Berlin
- Colombino M, Dall’Anese E, Bernstein A (2019) Online optimization as a feedback controller: stability and tracking. *IEEE Trans Contr Netw Sys* 7(1):422–432
- Conejo A, Carrión M, Morales J (2010) *Decision-making under uncertainty in electricity markets*. Springer, New York
- Dent C, Bialek J, Hobbs B (2011) Opportunity cost bidding by wind generators in forward markets: analytical results. *IEEE Trans Power Sys* 26(3):1600–1608
- Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res* 12(61):2121–2159
- Gan L, Low SH (2016) An online gradient algorithm for optimal power flow on radial networks. *IEEE J Select Areas Commun* 34(3):625–638
- Guo Z, Pinson P, Chen S et al (2021) Online optimization for real-time peer-to-peer electricity market mechanisms. *IEEE Trans Smart Grid* 12(5):4151–4163
- Hargreaves JJ, Hobbs BF (2012) Commitment and dispatch with uncertain wind generation by dynamic programming. *IEEE Trans Sustain Energy* 3(4):724–734
- Hauswirth A, Bolognani S, Hug G, et al (2016) Projected gradient descent on riemannian manifolds with applications to online power system optimization. In: 2016 54th annual allerton conference on communication, control, and computing (Allerton), IEEE, pp 225–232
- Hauswirth A, Zanardi A, Bolognani S, et al (2017) Online optimization in closed loop on the power flow manifold. In: 2017 IEEE Manchester PowerTech, IEEE, pp 1–6
- Hazan E et al (2016) Introduction to online convex optimization. *Found Trends Optimiz* 2(3–4):157–325
- Kakhbod A, Ozdaglar A, Schneider I (2021) Selling wind. *Energy J* 42(1):1–38
- Kraft E, Russo M, Keles D et al (2023) Stochastic optimization of trading strategies in sequential electricity markets. *Eur J Operat Res* 308(1):400–421
- Kuznetsova E, Li YF, Ruiz C et al (2013) Reinforcement learning for microgrid energy management. *Energy* 59:133–146
- Liakopoulos N, Destounis A, Paschos G, et al (2019) Cautious regret minimization: Online optimization with long-term budget constraints. In: International conference on machine learning, PMLR, pp 3944–3952
- Maggioni F, Cagnolari M, Bertazzi L (2019) The value of the right distribution in stochastic programming with application to a newsvendor problem. *Comput Manag Sci* 16(4):739–758
- Mazzi N, Pinson P (2016) Purely data-driven approaches to trading of renewable energy generation. In: 13th International conference on the European Energy Market (EEM), pp 1–5
- McMahan HB, Streeter M (2010) Adaptive bound optimization for online convex optimization. arXiv preprint [arXiv:1002.4908](https://arxiv.org/abs/1002.4908)
- Morales J, Conejo A, Pérez-Ruiz J (2010) Short-term trading for a wind power producer. *IEEE Trans Power Sys* 25(1):554–564
- Morales J, Conejo A, Madsen H et al (2014) *Integrating renewables in electricity markets - operational problems*. Springer, New York

- Muñoz M, Morales JM, Pineda S (2020) Feature-driven improvement of renewable energy forecasting and trading. *IEEE Trans Power Sys* 35(5):3753–3763
- Narayanaswamy B, Garg VK, Jayram T (2012) Online optimization for the smart (micro) grid. In: *Proceedings of the 3rd international conference on future energy systems*, pp 1–10
- Nonhoff M, Müller MA (2020) Online gradient descent for linear dynamical systems. *IFAC-Papers OnLine* 53(2):945–952
- Orabona F (2022) A modern introduction to online learning. arXiv preprint [arXiv:1912.13213](https://arxiv.org/abs/1912.13213)
- Ordoudis C, Pinson P (2016) Impact of renewable energy forecast imperfections on market-clearing outcomes. In: *Proceedings of the 2016 IEEE international energy conference (ENERGYCON)*, pp 1–6
- Pinson P, Chevallier C, Kariniotakis GN (2007) Trading wind generation from short-term probabilistic forecasts of wind power. *IEEE Trans Power Sys* 22(3):1148–1156
- Qin Y, Wang R, Vakharia AJ et al (2011) The newsvendor problem: review and directions for future research. *Eur J Operat Res* 213(2):361–374
- Refaeilzadeh P, Tang L, Liu H (2009) *Cross-validation*. *Encycl Database Syst*. Springer 5:532–538
- Shalev-Shwartz S et al (2012) Online learning and online convex optimization. *Found Trends Mach Learn* 4(2):107–194
- Sherman U, Koren T (2021) Lazy OCO: Online convex optimization on a switching budget. In: *Conference on learning theory*, PMLR, pp 3972–3988
- Stratigakos A, Camal S, Michiorri A et al (2022) Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy. *IEEE Trans Power Sys* 37(6):4696–4708
- Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction*. MIT Press, Cambridge
- Van Parys BP, Esfahani PM, Kuhn D (2021) From data to decisions: distributionally robust optimization is optimal. *Manag Sci* 67(6):3387–3402
- Wood K, Bianchin G, Dall’Anese E (2021) Online projected gradient descent for stochastic optimization with decision-dependent distributions. *IEEE Contr Sys Lett* 6:1646–1651
- Yuan D, Bhardwaj A, Petersen I et al (2022) Towards online optimization for power grids. *ACM SIGENERGY Energy Inform Rev* 1(1):51–58
- Zeiler MD (2012) ADADELTA: an adaptive learning rate method. arXiv preprint [arXiv:1212.5701](https://arxiv.org/abs/1212.5701)
- Zheng S (2011) Gradient descent algorithms for quantile regression with smooth approximation. *Int J Mach Learn Cybern* 2(3):191–207
- Zinkevich M (2003) Online convex programming and generalized infinitesimal gradient ascent. In: *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp 928–936
- Zugno M, Jónsson T, Pinson P (2013) Trading wind energy on the basis of probabilistic forecasts both of wind generation and of market quantities. *Wind Energy* 16(6):909–926
- Zugno M, Morales González J, Pinson P et al (2013) Pool strategy of a price-maker wind power producer. *IEEE Trans Power Syst* 28(3):3440–3450

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.