Online Optimization for the Smart (Micro) Grid

Balakrishnan Narayanaswamy, Vikas K. Garg and T.S. Jayram
IBM Research
Bangalore, India
{murali.balakrishnan,vikaskga,t.s.jayram}@in.ibm.com

ABSTRACT
Growing environmental awareness and new government directives have set the stage for an increase in the fraction of energy supplied using renewable resources. The fast variation in renewable power, coupled with uncertainty in availability, emphasizes the need for algorithms for intelligent online generation scheduling. These algorithms should allow us to compensate for the renewable resource when it is not available and should also account for physical generator constraints. We apply and extend recent work in the field of online optimization to the scheduling of generators in smart (micro) grids and derive bounds on the performance of asymptotically good algorithms in terms of the generator parameters. We also design online algorithms that intelligently leverage available information about the future, such as predictions of wind intensity, and show that they can be used to guarantee near optimal performance under mild assumptions. This allows us to quantify the benefits of resources spent on prediction technologies and different generation sources in the smart grid. Finally, we empirically show how both classes of online algorithms, (with or without the predictions of future availability) significantly outperform certain ‘natural’ algorithms.

Categories and Subject Descriptors
C.1.6 [Optimization]: Convex programming, Gradient methods

General Terms
Intelligent generator scheduling, Online gradient decent, Regret, Online convex optimization (OCO), Economic dispatch

1. INTRODUCTION
Growing environmental awareness and government directives have set the stage for an increase in the fraction of electricity supplied using renewable sources [30]. Distributed generation [2], especially solar and wind power collected across different small generation locations, is gaining considerable importance and their deployment is perceived as vital in achieving carbon reduction goals [24]. Extracting the maximum value from a time varying and intermittent renewable energy resource requires intelligent scheduling of both generation [4] and loads [17].

Intelligent generation scheduling (involving unit commitment [14] and economic dispatch [10]), is the process of scheduling different generation sources to minimize cost while meeting physical constraints of the electricity system. It is a highly non-linear problem, and usually solved using genetic programming or other non-convex optimization techniques [10]. Conventional economic dispatch, a very well researched methodology, is typically conducted 24 hours in advance (offline, day ahead) and uses the fact that the system load can be reasonably well predicted a day in advance. However, in a (micro) grid with high levels of wind penetration, this no longer holds due to the intermittent and unpredictable nature of wind power (which can only be reliably predicted a few minutes in advance [22]). This introduces practical challenges such as the ramping constraints, which limit how fast a generation source can increase or decrease its output over successive steps. Thus, the need for designing techniques, with a firm theoretical basis and worst case guarantees, that enable online generation scheduling subject to such physical constraints becomes very important, and our work is a major step in bridging this gap in literature.

Recent advances in wind prediction [22] offer hope that a reduction in the uncertainty of wind availability will lead to an increase in its value. Online versions of generation scheduling have been studied recently [4]. These algorithms, almost invariably, make assumptions regarding the stochastic nature of wind resources. For example, Xie and Ilic [31] use Model Predictive Control (MPC) for economic dispatch, where a model is constructed that predicts future renewable availability (assuming that the availability arises from some stochastic process) and then this prediction is used for generator optimization. These methods are computationally complex but seem to be effective in practice. This raises some interesting questions that motivate this paper (i) Are there computationally simple algorithms that are still provably effective under non-stationary or arbitrary renewable availability? (ii) Can we build a theoretical basis for the success of these online and MPC based algorithms? (iii) Can we use the theory to design algorithms that optimally incorporate available information about the future?

We address these issues in the context of intelligent online generator scheduling in microgrids with large, unpredictable
ongoing project to establish a research microgrid at the Kuala Belalong Field Studies Centre (KBFSC) \(^1\). The remote location and limited resource availability at this location make it an ideal platform to test new algorithms and technologies for the next generation of microgrids.

The concept of microgrids, which are (semi-) autonomous entities that co-ordinate DERs and loads in a decentralized manner, has been put forth to tackle the problem of large scale control for renewable integration. A microgrid usually comprises a Low Voltage (LV), \(\approx 1 \text{kV}\) locally-controlled cluster of DERs and loads that behaves, from the grid’s perspective, as a single producer or load both electrically and in the energy markets \([11]\). A salient feature of the microgrid lies in its ability to island \([18]\): it can continue to locally generate and consume electricity, possibly at a reduced level, even when disconnected from the grid. To meet carbon reduction goals and minimize electricity generation costs, it is imperative that the microgrids incorporate as large a fraction of the renewable energy generated as possible.

Intelligent scheduling of generation sources and loads is essential to the operation of a microgrid, to allow the integration of volatile DERs such as wind \([17]\), while ensuring stability and reliability. A major limitation of intelligent scheduling concerns the infeasible requirement of constant human intervention during the scheduling of loads and generators, motivating the need for algorithms that automatically modulate the generation or consumption levels with uncertain renewable power availability. In the sequel, while we motivate and describe our techniques in the context of generator scheduling in microgrids we again remark that our results can be readily adapted to handle general grids as well.

2. RESEARCH METHODOLOGY AND CONTRIBUTIONS

We model the intelligent generation scheduling problem as an online optimization problem where the objective function is defined to be the sum of time-dependent cost functions of the various time steps. The cost at each time step is determined by several components. The first is the cost of electricity generation due to the current generation level chosen by the algorithm. This is subject to the ramping constraints imposed by the generation source(s). In addition, there may also be uncertainty in the available wind so that the net effect is that the generated electricity may either be insufficient to meet the current demand or create a surplus. This is modelled as an additional cost function (which could be negative) as determined by the external market prices.

We propose online algorithms for the optimization problems that arise in the smart grid and analyse them in the strong adversarial model \([5]\). This is a powerful paradigm that makes no assumptions regarding the distributions, as in stochastic optimization, or ranges, as in robust optimization, characterizing the uncertainty of the unknown future. Therefore the results tend to have wider applicability.

The standard way to measure the performance of an online algorithm is with respect to an offline optimization strategy that knows the entire problem parameters with certainty \textit{a priori}. The key performance measure is that of \textit{regret}, which measures the difference between the online and the offline costs. It may seem unreasonable to expect any interesting guarantees because the adversary can simply make

\[^1\]http://ubdestate.blogspot.com/2009/06/kuala-belalong-field-studies-centre.html

renewable energy penetrations. Specifically,

- We demonstrate how, even in the harsh scenario where no prediction of the future is available and wind availability is chosen in an arbitrary manner, recent advances in online convex optimization \([7]\) can be fruitfully applied to generator scheduling in the next generation of smart (micro) grids. We also show how to exploit the special structure of the cost function for generator scheduling to obtain performance guarantees in terms of parameters governing the generation sources.

- We describe online algorithms \([13]\) that leverage information about the near future, such as prediction of wind availability and intensity more effectively. Interestingly, these algorithms use a strategy that discounts the future costs appropriately in order to prove guarantees on un-discounted future performance.

- We extend the work in online optimization to prescribe computationally simple online algorithms that model practical constraints of the generation sources, such as ramping constraints and multiple generation sources.

- We empirically show how both the classes of online algorithms, i.e. with or without lookahead, significantly outperform the existing ‘natural’ algorithms in the literature. For example, we show that discounting the future costs (perhaps counter-intuitively) can outperform algorithms with lookahead that do not discount the future. Thus, the theoretical techniques can be used to inform algorithm design.

Our work conclusively establishes the value of the proposed online algorithms and their theoretical analysis for the smart grid. The results equip us with a strong theoretical framework to quantify the benefits of resources spent on prediction technologies and multiple generation sources in the smart grid. While our algorithms can be used for both generator and load scheduling in a general grid, we describe our results in the context of economic dispatch for microgrids.

1.1 Need for intelligent online algorithms in smart (micro) grids

Recent policy amendments and new technology have necessitated the design of new algorithms for intelligent scheduling in smart grids. For example, in addition to the Kyoto Protocol, in 2007, Europe made a unilateral commitment to cutting its emissions by at least 20% of the 1990 levels by 2020. Since fossil fuel-based electricity is projected to account for more than 40% of global greenhouse gas emissions by 2020, renewable integration into the power grid will play a crucial role in meeting these goals.

Incorporating a large penetration of the intermittent Distributed Energy Resources (DERs), while ensuring grid stability, is a hard problem that requires a rethinking of how the grid is operated \([29]\). Traditional power grids are used to supply power from a few central generators to a large customer base. In contrast, the next generation smart grid, that incorporates distributed generation, must allow two-way flow of electricity and information in order to create an automated and distributed energy delivery network \([9]\).

The primary motivation for the problem we study is the ongoing project to establish a research microgrid at the

1

The re-

Kuala Belalong Field Studies Centre (KBFSC) \(^1\). The remote location and limited resource availability at this location make it an ideal platform to test new algorithms and technologies for the next generation of microgrids.

The concept of microgrids, which are (semi-) autonomous entities that co-ordinate DERs and loads in a decentralized manner, has been put forth to tackle the problem of large scale control for renewable integration. A microgrid usually comprises a Low Voltage (LV), \(\approx 1 \text{kV}\) locally-controlled cluster of DERs and loads that behaves, from the grid’s perspective, as a single producer or load both electrically and in the energy markets \([11]\). A salient feature of the microgrid lies in its ability to island \([18]\): it can continue to locally generate and consume electricity, possibly at a reduced level, even when disconnected from the grid. To meet carbon reduction goals and minimize electricity generation costs, it is imperative that the microgrids incorporate as large a fraction of the renewable energy generated as possible.

Intelligent scheduling of generation sources and loads is essential to the operation of a microgrid, to allow the integration of volatile DERs such as wind \([17]\), while ensuring stability and reliability. A major limitation of intelligent scheduling concerns the infeasible requirement of constant human intervention during the scheduling of loads and generators, motivating the need for algorithms that automatically modulate the generation or consumption levels with uncertain renewable power availability. In the sequel, while we motivate and describe our techniques in the context of generator scheduling in microgrids we again remark that our results can be readily adapted to handle general grids as well.

2. RESEARCH METHODOLOGY AND CONTRIBUTIONS

We model the intelligent generation scheduling problem as an online optimization problem where the objective function is defined to be the sum of time-dependent cost functions of the various time steps. The cost at each time step is determined by several components. The first is the cost of electricity generation due to the current generation level chosen by the algorithm. This is subject to the ramping constraints imposed by the generation source(s). In addition, there may also be uncertainty in the available wind so that the net effect is that the generated electricity may either be insufficient to meet the current demand or create a surplus. This is modelled as an additional cost function (which could be negative) as determined by the external market prices.

We propose online algorithms for the optimization problems that arise in the smart grid and analyse them in the strong adversarial model \([5]\). This is a powerful paradigm that makes no assumptions regarding the distributions, as in stochastic optimization, or ranges, as in robust optimization, characterizing the uncertainty of the unknown future. Therefore the results tend to have wider applicability.

The standard way to measure the performance of an online algorithm is with respect to an offline optimization strategy that knows the entire problem parameters with certainty \textit{a priori}. The key performance measure is that of \textit{regret}, which measures the difference between the online and the offline costs. It may seem unreasonable to expect any interesting guarantees because the adversary can simply make
the online algorithm “pay” heavily for the lack of knowledge regarding the future resulting in poor performance. One way to circumvent this is to place restrictions on the offline algorithm’s choices. The remarkable achievement of this theory is that under reasonable restrictions, we can design online algorithms that are intuitive, simple to implement, and yield good performance guarantees. The analysis techniques are quite involved, drawing on tools from convex optimization, Markov decision processes, and stability theory.

In this paper, we consider two possible approaches to tackle the uncertainty in the future. In the first setting, also known as online convex optimization (OCO) [7], the cost function for each time step is fully known only after the algorithm chooses the generation level for that step. In this case, we restrict the offline algorithm to make one fixed choice of the generation level for entire duration, but we stress that this choice is made in hindsight. The central result of OCO theory is that it is possible to design online algorithms that achieve regret sub-linear in the number of time steps $T$. Specifically, taking into account the structure of the cost function, we derive a regret bound of $O(\log T)$ for the generation scheduling problem. We note that average regret per time step is $O(\frac{\log T}{T})$ which vanishes as $T$ tends to $\infty$. In addition to the theoretical bounds, we show in simulation that this simple algorithm has a substantially better performance than a forecaster commonly used in the economic dispatch literature. An extension of this setup is one where the adversary’s choice of the point is not fixed but allowed to vary slightly. Our methods do apply to this problem setting and still yield the same $O(\log T)$ regret bound mentioned above albeit with slightly worse constants.

In the second setting, we consider a more practical problem where the wind information is available for a limited horizon in the future, drawing on recent work in short term wind forecasting [22, 1]. This allows the cost functions to be known with certainty for the next $L$ steps for some significantly large parameter $L$ called the lookahead. In this case, we design a variant of the greedy algorithm based on the available information to decide the generation level for the next time step. The novelty comes from the fact that the strategy discounts the future available costs in a geometrically decreasing fashion. We stress however that the goal is to optimize the sum of costs and not the sum of discounted costs which is just an artifact of the strategy.

We analyze the regret of this algorithm with respect to the strongest possible offline algorithm—one that is allowed to change the generation levels in hindsight. The main result is again a sub-linear regret algorithm for an appropriately large but reasonable lookahead. Such approaches have been considered before and are reminiscent of Model Predictive Control (MPC) algorithms used in control theory [8, 20, 23]. Our analysis provides some theoretical justification for such algorithms. Recall that at the basic level, an MPC algorithm is a sequence of open-loop policies where in each step, the algorithm uses a model to compute an optimal trajectory, and takes just the first step of that path. Then the model is recomputed with respect to the feedback provided at that step. Our algorithm closely mirrors the MPC approach except that we use the discounted paths to argue in support of our algorithm, with respect to the un-discounted total reward for a finite horizon $T$, against an all powerful adversary that can choose an arbitrary path at the end of the finite horizon. Our algorithm generalizes the work in [13] to account for the ramping conditions which, stated in the language of [13], applies to a more general setting where not all state-state transitions are legal. The theoretical bounds also quantify the improvement in performance with lookahead and the interaction between the amount of lookahead, the ramping constraints of the generator and the discounting factor that should be used. Through simulations we show that discounting the future reward, which was a proof strategy used to obtain our bounds, actually performs better that an algorithm that does not discount the future. Thus, the analysis methods we use inform algorithm design.

The rest of the paper is organized as follows. We describe an abstract model of a (micro) grid with scheduleable generators and loads and introduce some notation in Section 3. In Section 4, we show how OCO can be used in a simple economic dispatch model with one generation source and how the particular structure of the cost function in electric power systems allows us to derive strong bounds on performance. We also describe how OCO techniques can be used to handle practical considerations like the ramping constraints of a practical generator or the scheduling of multiple generators. In Section 5, we describe algorithms that efficiently incorporate predictions of the future availability of intermittent resources. Finally, we summarize, and describe some theoretical, algorithmic and practical directions for future research in Section 7.

3. MODEL DESCRIPTION

We first describe the microgrid scenario in more detail and introduce some necessary notation. We first consider a microgrid with a single generation source (say a turbine-boiler generator), though we will describe extensions to multiple generators in Section 4.4. Generators can be modeled as having a quadratic cost curve [26], so that the cost of generating $\theta_t$ using a generator can be expressed as,

$$C_t(\theta_t) = a\theta_t^2 + b\theta_t + c$$

(1)

For a typical 76MW coal generator $^2 a = 0.002, b = 2.680$ and $c = 35.385, \text{ when step size is 10min and } \theta_t \text{ is in MW}$. We model a discrete time version with slot size comparable to the rate of variation in wind power (say 5 to 15 mins).

The microgrid has a time varying source of (free) renewable power $r_t$ and has to satisfy a time varying load $l_t$. The load can be predicted quite accurately, however, the wind power usually cannot be or can only be predicted accurately only a few slots in advance. We are interested in algorithms for both these situations. In our model, we treat the wind power generated as a negative load resulting in a net demand at every slot $d_t = \delta_t + x_t$, where $\delta_t$ is the predicted value of $l_t - r_t$ one day ahead and $x_t$ is the (unknown) error due to unpredictable nature of the wind and deviation in demand from the predicted value. This modeling is quite natural considering the day-ahead nature of electricity markets [26].

Note: Since the base load $\delta_t$ is settled in the day ahead market or can be satisfied cheaply using slower generation, for notational simplicity, we do not consider it below. It can however be included if required in the analysis. The only difference would be that the allowable generation level $\theta_t$ might also take negative values corresponding to a decrease in comparison to the pre-committed level. In the sequel, we will be only concerned with the $x_t$ process.

$^2$http://pscal.ece.gatech.edu/testsys/generators.html
At every time slot, the microgrid schedules some generation \( \theta_t \) at cost \( C_G(\theta_t) \). Once demand (load + wind power) is revealed, the microgrid has to buy the shortfall at \( \lambda^{buy}_t \)/unit of electricity or sell the surplus on the spot market at \( \lambda^{sell}_t \)/unit of electricity. Thus, the net cost to satisfy the load is

\[
C^*_n(\theta_t) = a \theta^2_t + b \theta_t + c + \lambda^{buy}_t(x_t - \theta_t)^+ - \lambda^{sell}_t(\theta_t - x_t)^+ \quad (2)
\]

where, \( \theta_t \geq 0 \), \((x_t - \theta_t)^+\) is \((x_t - \theta_t)\) if \(x_t \geq \theta_t\) and 0 otherwise.

The offline version for the generation scheduling problem, where all the \( x_t \)'s are known a priori, is a convex problem with linear constraints and can be solved using any of the standard convex optimization methods [6]. However, the structure of the constraints allows us to solve the optimization problem more efficiently as outlined in Appendix C.

4. INTELLIGENT GENERATOR SCHEDULING: LOW REGRET DISPATCH

We will see how OCO provides simple algorithms for generator scheduling with no knowledge of the future of the process \( x_t \). We will first show that the cost function (2) has a special structure that improves the bounds on the performance of OCO algorithms.

The net cost paid by the generation for serving a load \( x_t \) when a generation \( \theta_t \) is scheduled is given by (2). The agent that schedules the generation, is essentially computing a prediction of the load \( x_t \) and is sometimes also called a forecaster. We would like a forecaster that generates a sequence \( \theta_1, \ldots, \theta_T \) that has low regret, that is low values of

\[
R_T = \max_{\theta^*} \sum_{t=1}^T C^*_n(\theta_t) - C^*_n(\theta^*) \quad (3)
\]

We are thus comparing our forecaster against a hypothetical algorithm that has access to the entire sequence of \( x_t \) but is constrained to select a fixed generation value \( \theta^* \) for the entire duration. We are particularly interested in forecasters that have sub-linear regret so that the average per-slot regret \( \frac{1}{T} R_T \) goes (as quickly as possible) to zero as \( T \to \infty \). In such situations our algorithms are essentially as good as the best fixed forecast in hindsight. We assume that \(|\theta_t| \leq \theta_{max} \) and \(|x_t| \leq X \) as otherwise the regret can be made unbounded.

4.1 Interpreting the cost function

The first observation we make is that the cost function in (2) is convex only if

\[
\lambda^{buy}_t \geq \lambda^{sell}_t \quad (4)
\]

This is also sensible since if the profit from selling is larger than the cost of buying, there is an arbitrage opportunity and infinite profit can be made. This situation should never arise in practice.

In the microgrid will not generate electricity \( \theta \) if

\[
a \theta^2 + b \theta \geq \lambda^{buy}_t \theta \quad (5)
\]

In particular the microgrid will never generate if \( b \geq \lambda^{buy}_t \).

So we assume \( \lambda^{buy}_t \geq b \) and (4) throughout this paper. While the above two conditions are simple, they are important to keep in mind.

We would like to see how the practical aspects of the problem, such as the nature of the cost function and physical constraints, effect the solution and guarantees. The cost function in (2) is a convex loss function which guarantees a single global minima. Actually, it is a strongly convex function, which will have implications on the rate of decrease of regret of our algorithms.

**Definition** A cost function \( C_{net} \) is strongly convex for a certain \( \xi > 0 \) with parameter \( \sigma \) if

\[
C_{net}(u) \geq C_{net}(\theta) + \xi (u - x) + \sigma \| u - x \|^2 \quad (6)
\]

We define,

\[
G \equiv \sup \{ \| \xi \| : \xi \in \partial C_{net}(\theta), 0 \leq \theta \leq \theta_{max} \} \quad (7)
\]

where \( \partial C_{net}(\theta, x) \) is the set of subgradients of \( C_{net} \) at \( \theta \).

**Lemma 1.** The cost function in (2) is strongly convex for \( \xi \leq G \), where \( G \) is given by,

\[
G \leq 2a \theta_{max} + b - \max_t \lambda^{buy}_t股 (8)
\]

and \( \sigma = 2a \) and \( \lambda^{buy}_t \)

For simplicity we focus on a particularly simple online gradient descent type algorithm due to [32] though there are many online algorithms with different properties that may be useful [7].

4.2 Online generation optimization

The algorithm proceeds as follows. At time \( t \) we need to make a decision on how much to generate from the generators at time \( t+1 \). The Zinkevich update [32] suggests that we should generate

\[
y_{t+1} = \theta_t - \eta \frac{\partial C_{net}(\theta)}{\partial \theta} |_{\theta = \theta_t} \quad (9)
\]

projected on to the feasible set. For our cost function (2) this reduces to

\[
y_{t+1} = \begin{cases} 
\theta_t - \eta [2a \theta_t + b - \lambda^{buy}_t] & \text{if } \theta_t \leq x_t \\
\theta_t - \eta [2a \theta_t + b - \lambda^{sell}_t] & \text{if } \theta_t > x_t
\end{cases} \quad (10)
\]

Since the allowed generation lies in ball \( \theta \in K = [0, \theta_{max}] \),

\[
\theta_{t+1} = \min(\max(0, y_{t+1}), \theta_{max}) \quad (11)
\]

For such an update, we can prove that,

**Theorem 2.** The regret of the online generation scheduling algorithm can be bounded as

\[
R_T \leq \frac{G^2}{\sigma} (\log T + O(1)) \quad (12)
\]

where \( G \) is as in (8), \( \sigma = 2a \) and \( D = \theta_{max} \). Thus, the per-slot regret \( \frac{R_T}{T} \) goes to zero as \( O \left( \log \frac{T}{\sigma} \right) \).

**Proof.** While theorems of this form are known in the OCO literature [7], we present the proof for our strongly convex cost function for completeness. Basically, the strongly convex nature of the economic dispatch cost function allows an intelligent choice of the learning rate \( \eta \), which gives us better bounds where the total regret increases only logarithmically with \( T \).

From the strong convexity condition (6), \( \forall \eta \in K \),

\[
\sum_{t=1}^T C_{net}(\theta_t) - C_{net}(u) \leq \sum_{t=1}^T \xi_t (\theta_t - u) - \frac{\sigma}{2} (u - \theta_t)^2 \\
\leq \sum_{t=1}^T \left( \frac{1}{\eta} - \frac{1}{\eta_{t-1}} - \sigma \right) \cdot \frac{1}{2} (u - \theta_t)^2 + \sum_{t=1}^T \eta_t \xi_t^2 \quad (13)
\]
where (13) comes from using Lemma 8. Now, substituting \( \eta_t = \frac{1}{T} \) we conclude
\[
\sum_{t=1}^{T} C^*_{net}(\theta_t) - C^*_{net}(u) \leq \sum_{t=1}^{T} \frac{G^2}{\sigma} \sum_{t=1}^{T} \frac{1}{t} \tag{14}
\]
The theorem follows using the standard bound for the harmonic series. □

This guarantees that in the long run the online scheduling algorithm given by (10) and (11) performs essentially as well as the best fixed generation level in hindsight.

4.3 Ramping constraints

While the algorithm in the previous section is simple and effective, it may become infeasible in practice if rapid variation in the wind cause the forecasts to vary drastically across slots. This is because, in general generators have ramping constraints [10], i.e. constraints of the form
\[
|\theta_{t+1} - \theta_t| \leq R \quad \forall t
\tag{15}
\]
These ramping constraints become important because of the possibility of very high slew rate in wind power availability [19]: “On 11th February 2007, the Irish wind power fell steadily from 415 MW at midnight to 79 MW at 4am”.

This amounts to about 1.5MW/min. In comparison, a typical thermal generator would have a ramping rate of about 5 – 15% of capacity/min.

In order to ensure that the updates in (10) and (11) satisfy the ramping constraints (15), we need that
\[
\eta_t \leq \frac{R}{G} \quad \forall t
\tag{16}
\]
With this constraint on \( \eta_t \) we can state the following result on the regret of generation scheduling with ramping constraints

**Theorem 3.** The regret of the online generation scheduling algorithm, with ramping constraints, can be bounded as
\[
R_T \leq \frac{G^3}{R \sigma} (\log T + O(1)) \tag{17}
\]
where \( G \) is as in (8), \( \sigma = 2a, D = \theta_{\text{max}} \) and \( R \) is the ramping rate as in (15). The per-slot regret \( \frac{R_T}{T} \) goes to zero as \( O \left( \frac{\log T}{G} \right) \).

4.4 Multiple generation sources

One possible solution that has been suggested to enable faster ramping is to have multiple generation sources. While the ramping as a fraction of the total generation remains in the same range, having multiple generators allows faster response to wind events. We show how the online gradient descent algorithm from the previous sub-section can easily be extended to this situation as well.

With multiple generators, \( i = 1, 2, \ldots, N_G \), each with their own cost coefficients \( a^i, b^i, c^i \) the total cost function becomes
\[
C^*_{net}(\theta_t) = \sum_{i=1}^{N_G} a^i \theta^i_{t+1} + \sum_{i=1}^{N_G} b^i \theta^i_{t} + \sum_{i=1}^{N_G} c^i + \lambda^i_{\text{buy}}(x_t - \sum_{i=1}^{N_G} \theta^i_t)^+ + \lambda^i_{\text{sell}}(\sum_{i=1}^{N_G} \theta^i_t - x_t)^+
\tag{18}
\]

Each generator \( i \) has a ramping constraint of the form \( |\theta^i_t - \theta^i_{t+1}| \leq R_i \), \( \forall t \)

**Theorem 4.** The regret of the online generation scheduling algorithm with multiple constrained generators, can be bounded as
\[
R_T \leq \frac{G^3}{R_{\text{min}} \sigma} (\log T + O(1)) \tag{19}
\]
where \( G \) is as in (8), \( \sigma = 2a, D = \theta_{\text{max}} \) and \( R \) is the ramping rate as in (15). The per-slot regret \( \frac{R_T}{T} \) goes to zero as \( O \left( \frac{\log T}{G} \right) \), where \( R_{\text{min}} = \min_i R_i \).

**Proof.** For this cost function, the Zinkevich update (9) for each generator \( i \) reduces to
\[
\theta^i_{t+1} = \begin{cases} 
\theta^i_t - \eta_t [2a \theta^i_t + b - \lambda^i_{\text{buy}}] & \text{if } \sum_{i=1}^{N_G} \theta^i_t \leq x_t \\
\theta^i_t - \eta_t [2a \theta^i_t + b - \lambda^i_{\text{sell}}] & \text{if } \sum_{i=1}^{N_G} \theta^i_t > x_t 
\end{cases} \tag{20}
\]
To account for the fact that generation lies in ball \( \theta^i_t \in [0, \theta^i_{\text{max}}] \), we have
\[
\eta_t \leq \frac{R_{\text{min}}}{G} \quad \forall t \tag{22}
\]
where \( G \) is as in (8), with \( \theta_{\text{max}} = \sum_i \theta^i_{\text{max}} \).

Note that using this approach the regret bound depends on the ramping constraint of the most constrained generator indicating that, at least in the worst case, the benefits of multiple constrained generators is limited. Further analysis using specific statistics of wind or solar power availability would be an interesting direction of further investigation, to identify when multiple generators are an economical decision.

4.5 Simulations

We now consider some simulations to highlight the performance of the algorithms that may be hidden by the proofs. For simplicity we assume that the microgrid operator is interested in using all the wind power generated, and thus schedules for the largest possible wind output. Thus, only shortfalls are possible, so that \( x_t \geq 0 \). This is more realistic in light of recent laws enacted in European countries (especially Germany) and recommendations of the Global Wind Energy Council\(^3\) that require that all the wind power generated be utilized.

We consider a simple ‘ramping’ model of wind availability to demonstrate the effectiveness of the ramp constrained OCO updates with \( \eta_t \) as in (16). In this model, wind event \( i \) occurs after time \( T_i^{\text{up}} \) and continues at a peak power value for \( T_i^{\text{dp}} \). Each event also has a ramp up time \( t_i^{\text{up}} \) and a ramp down time \( t_i^{\text{dn}} \). During the ramp time the wind power changes linearly from initial value to final value.

We simulate the case where \( T_i^{\text{up}} \) and \( T_i^{\text{dp}} \) are drawn from an exponentially distribution with parameter \( \mu^i_{\text{up}} \) and \( t_i^{\text{up}} \) and \( t_i^{\text{dn}} \) are drawn from an exponentially distribution with parameter \( \mu_i \). A typical wind power output sequence is shown in Figure 1.

\(^3\)See for example ftp://ftp.sni.technion.ac.il/events/2011-12-19/levon.pdf
To reduce the number of parameters we fix $\lambda_{buy} = \lambda_{buy}$ (a constant) and $\lambda_{sell} = 0$. This is the special case where the microgrid cannot sell back to the main grid at a profit. In order to understand the effect of wind ramps we fix $T_{rf}$ and vary $\mu_t$ (the exponential parameter for the ramp times of wind events). We also use the thermal generation parameters from Section 3. Finally, in order to remove as much explicit dependence of parameter choices we plot the ratio between the total cost incurred by different algorithms and the cost of the OCO update algorithm.

In Figure 2 we compare the performance of the OCO update, a simple greedy predictor and the best fixed generation level, chosen in hindsight. The greedy algorithm is essentially a persistence based forecaster [22], that schedules the generation optimally assuming that the wind availability in the next slot will be same as the wind availability in the current slot. This naive method has been seen to be hard to beat 1 to 6 hours ahead, but we see how intelligent scheduling leads to substantially improved performance.

5. INTELLIGENT GENERATOR SCHEDULING WITH LOOKAHEAD

The online optimization framework discussed in the preceding section guarantees a low regret with respect to an adversary that chooses a best fixed point (in hindsight) for the entire time horizon. In this section, we consider a much more powerful offline baseline that is free to choose a different point at each time step. We show that the availability of some extra information, in terms of lookahead or a glimpse into the future demands, can be efficiently used to obtain almost as good a performance as this strong offline oracle.

5.1 Generator scheduling with discounted future rewards

Our system with lookahead can be abstracted (without loss of generality) to work as follows. At each step $t$, with a lookahead of $L$, the online algorithm has access to the net loads for each of the next $L$ steps, in addition to that for the current step. Each action incurs a cost, and the objective of the online algorithm is to minimize the average cost (or equivalently, maximize the average reward) over the specified time horizon $T$.

Let there be an expected reward $r_t$ associated with each $\theta_t$ choice. Define a $L$-strategy, during any time step, to be a decision regarding the amount of electricity to be produced in sequence for the next $L+1$ steps. Note that the successive choices prescribed by any strategy must obey the ramping constraint. We now present a $L$-lookahead based deterministic algorithm $ON$ that is asymptotically optimal as $L$ grows. For any strategy that collects $L + 1$ rewards $r_0, r_1, \ldots, r_L$ in the next $L + 1$ slots, we define its (discounted) anticipated reward to be $r_0 + \gamma r_1 + \ldots + \gamma^L r_L$, for some $\gamma \in (0, 1)$. The algorithm $ON$ greedily follows at each time step the strategy with maximum anticipated reward. In other words, at every time step that strategy is chosen for the next $L + 1$ steps whose anticipated reward is maximal, and the online algorithm takes action dictated by this strategy in the current step. Since the algorithm recomputes the maximal strategy at each step, the strategies at successive time steps may be different. In effect, we solve the following optimization problem at each time step $t$:

$$\max_{\theta_t, \theta_{t+1}, \ldots, \theta_{t+L}} r_t(\theta_t, x_t) + \sum_{i=1}^{L} \gamma^i r_{t+i}(\theta_{t+i}, x_{t+i})$$

subject to

$$|\theta_{i+1} - \theta_i| \leq R \ \forall i$$

where,

$$r_i = 1 - \frac{C_{net}(\theta_t)}{C_{max}} \in [0, 1] \ \forall i,$$

and $C_{max} = \max_{t} \max_{\theta_t} C_{net}(\theta_t)$. 

Figure 1: Sample wind events generated from the model.

Figure 2: Cost ratio of OCO and baseline algorithms. A higher ratio shows better performance of OCO. Lower ramp times indicate higher wind volatility.
This, in turn, can be used to solve our problem of interest:

\[
\min_{\theta_t, \theta_{t+1}, \ldots, \theta_{t+L}} C^t_{\text{net}}(\theta_t) + \sum_{i=1}^{L} \gamma^i C^{t+i}_{\text{net}}(\theta_{t+i})
\]

subject to

\[|\theta_{t+1} - \theta_t| \leq R \ \forall i\]

We show that our online algorithm can use the lookahead to achieve a low regret with respect to an optimal offline algorithm that has access to the entire \(x_t\) sequence a priori.

**Theorem 5.** The online algorithm with a lookahead \(L\) on future costs is asymptotically optimal for any \(L = f(T) + \frac{2 \ln T}{\gamma f(T)}\) that satisfies \(f(T) \rightarrow \infty\) as \(T \rightarrow \infty\).

**Proof.** Essentially, we show that there exists an optimal choice of the discounting factor (that depends on the amount of lookahead \(L\)) such that the the (future discounted) online algorithm has good performance in terms of undiscounted total reward. We then bound the sub-optimality of the algorithm. See Appendix B for the details of the proof.

This theorem shows that, for example, even \(L = \log \log T + \frac{2 \ln T}{\gamma f(T)}\) is sufficient. Thus, it is the ratio \(\frac{2 \ln T}{\gamma f(T)}\) that primarily determines the amount of lookahead needed for (asymptotic) optimality, though the rate at which optimality is achieved will increase with increasing \(L\).

### 5.2 Simulations

We now consider some simulations to quantify the benefits of lookahead and also to highlight some interesting differences between the future discounted analysed above and the undiscounted \(\gamma = 1\) version. Recall that the undiscounted version was suggested for generator scheduling in [31].

For the purpose of the simulation we consider a simple wind power availability pattern shown in Figure 3. We consider the performance of 3 algorithms on this pattern: (i) the OCO algorithm from Section 4 (ii) the future discounted algorithm with lookahead from Section 5.1 and (iii) an algorithm that considers the optimal generation schedule with the same lookahead but without a discounting factor. The results of the of the simulation is in Figure 4.

![Figure 3: A simple periodic wind pattern with period 6.](image)

![Figure 4: A comparison of the performance of (i) the OCO algorithm from Section 4 (ii) the future discounted algorithm with lookahead from Section 5.1 and (iii) an algorithm that considers the optimal generation schedule with the same lookahead but without a discounting factor.](image)
optimization algorithms we study for generation scheduling have natural counterparts for the problems faced by DSM agents and exploring these ideas is a promising direction of future work we intend to pursue.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have demonstrated that the theory and algorithms developed for online optimization are useful for generator scheduling problems in the smart grid, with suitable extensions to account for practical constraints such as generator parameters and ramping constraints. We designed simple algorithms, derived guarantees on their performance under mild assumptions and showed that they perform well even when no predictions of the future are available.

We showed how to incorporate predictions of the future renewable availability effectively into online generator scheduling algorithms, and quantified the benefits of lookahead. Interestingly, we showed that discounting the future is useful both as a proof technique and as a strategy for generator scheduling with ramping constraints.

In addition to load scheduling for cost minimization (based on reviewer comments) we are particularly interested in understanding online optimization algorithms for other applications including voltage support, distribution losses, and energy storage management that are particularly important in smart grids with a large penetration of renewable energy.

Finally, on a theoretical side new online algorithms or time varying discounting strategies that have better performance guarantees would be an interesting direction of future research.

Acknowledgements

We would like to thank Shivkumar Kalyanaraman for helpful comments during the preparation of this paper. We would also like to thank the anonymous reviewers who pointed out useful extensions of these ideas that we hope to explore in future work.

8. REFERENCES


Information Sciences Division.


9. ACKNOWLEDGEMENTS

We would like to thank the reviewers for their insightful comments that we hope to address in follow-up work.

APPENDIX

A. ADDITIONAL LEMMAS FOR OCO

We first give a high level overview of the proof structure. Lemma 6 and Lemma 7 together bound the ‘size’ of the update step used by the online gradient descent algorithm. Lemma 8 uses this bound to select an optimal learning rate $\eta_t$, such that the regret grows as slowly as possible. This is used in Theorem 2 to obtain the final regret bound.

**Lemma 6.** For all $t \geq 1$, $u \in \mathcal{K}$, 
\[ \partial C^*_nlt(y_{t+1}) (\theta_{t+1} - u) \leq \frac{(\theta_{t} - u)^2}{2} - \frac{(\theta_{t+1} - y_{t+1})^2}{2} - \frac{(\theta_{t+1} - \theta_{t})^2}{2} \]
Proof. Since $\theta_{t+1}$ is the unconstrained minimizer of $C^*_nlt(y)\frac{(\theta_{t} - y_{t+1})^2}{2}$, $\partial C^*_nlt(y_{t+1}) = (\theta_{t} - y_{t+1})$. We then have
\[ \partial C^*_nlt(y_{t+1}) (\theta_{t+1} - u) = \frac{(\theta_{t} - u)^2}{2} - \frac{(\theta_{t+1} - y_{t+1})^2}{2} - \frac{(\theta_{t+1} - \theta_{t})^2}{2} \]
By combining the middle two terms this gives the lemma \[ \square \]

**Lemma 7.** For all $t \geq 1$ and $u \in \mathcal{K}$,
\[ \partial C^*_nlt(y_{t+1}) (\theta_{t+1} - u) \leq \frac{(u - \theta_{t+1})^2}{2} - \frac{(u - \theta_{t})^2}{2} + \partial C^*_nlt(y_{t+1})^2 \]
Proof. Substitute $u = \theta_{t}$ in Lemma 6 and simplify,
\[ \partial C^*_nlt(y_{t+1}) (\theta_{t+1} - \theta_{t}) \geq (\theta_{t} - \theta_{t+1})^2 \]
But, by Holder’s inequality we have
\[ \partial C^*_nlt(y_{t+1}) (\theta_{t+1} - u) \leq |C^*_nlt(y_{t+1})||(|\theta_{t+1} - u)| \]
Thus, $\partial C^*_nlt(y_{t+1}) (\theta_{t+1} - u) \leq \partial C^*_nlt(y_{t+1})^2$. Combining this with the statement of Lemma 6 completes the proof. \[ \square \]

**Lemma 8.** Relationship between learning rate $\eta_t$ and strong convexity parameter $\xi_t$.
\[ \sum_{t=1}^{T} \xi_t (\theta_{t+1} - u) \leq \sum_{t=1}^{T} \left( \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma \right) \frac{1}{2} (u - \theta_{t})^2 + \sum_{t=1}^{T} \eta_t \xi_t \]
Proof. In Lemma 7, choose $\eta_t = \frac{\partial C^*_nlt(y_{t+1})}{\xi_t}$. We have
\[ \sum_{t=1}^{T} \xi_t (\theta_{t+1} - u) = \sum_{t=1}^{T} \frac{\partial C^*_nlt(y_{t+1})}{\eta_t} (\theta_{t+1} - u) \leq \sum_{t=1}^{T} \frac{1}{\eta_t} \left( \frac{1}{2} (u - \theta_{t+1})^2 \right) \]
Simplifying, and using the non-negativity of the square terms gives us the final result. \[ \square \]

B. ONLINE SCHEDULING WITH LOOKAHEAD

The proof shows that the online algorithm with discounting appropriately considers the possible ‘good’ paths so that the optimal offline algorithm cannot be much better.

Consider the strategies considered by $ON_T$ at time $t$. Let $r_0, r_1, \ldots, r_L$ be the rewards associated with the $L$-lookahead strategy having the maximum anticipated reward at time $t$. Define $ON_T = r_0$ and $ON_T^{LT} = \gamma r_1 + \ldots + \gamma^L r_L$. That is, $ON_T + ON_T^{LT}$ represents the maximum anticipated reward across all $(L + 1)$-length strategies that are available to $ON_T$ at time $t$. Moreover, since $ON_T$ follows this maximum anticipated reward strategy in the current step, a reward of $ON_T$ is amassed by $ON_T$ at time $t$. Fix an optimal offline algorithm, and let $OPT_T$ denote the reward collected by this algorithm at time $t$. Further, let $OPT_T^{LT}$ be a shorthand for $\gamma OPT_T + \gamma^2 OPT_T + \ldots + \gamma^L OPT_T$. Noting that one strategy competing with the strategy having the maximum anticipated reward, at time $t$, is to join the offline algorithm at time $t + 1$ and then follow it for next $L$ steps. As a consequence of ramping, it may not be possible for the online algorithm to mimic the optimal algorithm after one step; in particular, in the worst case, the online algorithm may have to wait for $\Delta = \max_{t \leq T} \Delta_t$ steps (during which, in the worst case, it may not register any reward). Thus, one strategy competing with the strategy having the maximum anticipated reward, at time $t$, is to join the offline algorithm at time $t + \Delta$ and then follow it for next $L$ steps. Therefore,
\[ ON_T + ON_T^{LT} \geq OPT_T^{LT} \\Delta + 1 \]
Another strategy available with $ON_T$ is to follow the strategy that had the maximum anticipated reward during the previous time step $t - 1$. The contribution of the rewards in
Combining (27) and (28), we obtain

\[ ON_i + ON_{i+1}^{*L} \geq \alpha ON_{i}^{*L} \]  

(28)

Combining (27) and (28), we obtain

\[ ON_i + ON_{i+1}^{*L} \geq ON_i^{*L} + \left( 1 - \frac{1}{\alpha} \right) OPT_{t+\Delta}^{L-\Delta+1} \]  

(29)

Summing over all time steps \( t \) (and assuming suitable zero padding), we obtain the following telescopic sum:

\[ \sum_t ON_t \geq \left( 1 - \frac{1}{\alpha} \right) \sum_t OPT_{t+\Delta}^{L-\Delta+1} \]

\[ = \left( 1 - \frac{1}{\alpha} \right) \left( \frac{1}{\alpha^3} + \frac{1}{\alpha^{3+1}} + \ldots + \frac{1}{\alpha^L} \right) \sum_t OPT_t \]

\[ \Rightarrow \sum_t OPT_t \leq \frac{\alpha^{L+1}}{\alpha^{L-\Delta+1} - 1} = g(\alpha, L). \]

(30)

Expressing (30) in terms of cost, we obtain

\[ \sum_t \frac{1 - \frac{OPT_t}{C_{\max}}} {\sum \frac{1 - ON_t}{C_{\max}}} \leq g(\alpha, L) \]

Assuming a time horizon of \( T \), we get

\[ T \sum_{t=1}^{T} \frac{OPT_t}{C_{\max}} \leq g(\alpha, L) \sum_{t=1}^{T} ON_t \]

\[ \Rightarrow \sum_{t=1}^{T} ON_t \leq T C_{\max} \frac{g(\alpha, L)}{g(\alpha, L)} + \sum_{t=1}^{T} \frac{OPT_t}{g(\alpha, L)} \]

Therefore, the total regret can be expressed as

\[ \sum_{t=1}^{T} ON_t - \sum_{t=1}^{T} OPT_t \leq T C_{\max} \frac{g(\alpha, L)}{g(\alpha, L)} + \sum_{t=1}^{T} \frac{OPT_t}{g(\alpha, L)} \]

whence the average per-slot regret can be bounded as shown below:

\[ \frac{T}{T} \sum_{t=1}^{T} ON_t - \sum_{t=1}^{T} OPT_t \leq C_{\max} \left( 1 - \frac{1}{T} \right) \left( 1 - \frac{1}{g(\alpha, L)} \right) \]

In order to minimize the regret, we want to determine a value of \( \alpha \) that minimizes \( g(\alpha, L) \) for \( L \geq \Delta \). It is straightforward to compute the optimal \( \alpha \) using (30):

\[ \alpha^* = \left( \frac{L + 1}{\Delta} \right) \frac{1}{L - \Delta + 1} \]

which, in turn, implies

\[ \frac{1}{g(\alpha, L)} = 1 - \Theta \left( \frac{\log L}{L - \Delta + 1} \right) \]

Hence, the average regret can be expressed in terms of \( L \) as,

\[ \frac{\sum_{t=1}^{T} ON_t - \sum_{t=1}^{T} OPT_t}{T} \leq C_{\max} \left( 1 - \frac{1}{T} \right) \Theta \left( \frac{\log L}{L - \Delta + 1} \right) \]

\[ = C_{\max} \left( 1 - \frac{1}{T} \right) \Theta \left( \frac{\log L}{L - \frac{\theta_{\max}}{R} + 1} \right) \]

(31)

(using the fact that \( \Delta = \theta_{\max} \), since \( |\theta_1| \leq \theta_{\max} \) and \( |\theta_{t+1} - \theta_t| \leq R \forall t \in [T] \)). This gives the theorem.

C. OFFLINE OPTIMIZATION FOR GENERATOR SCHEDULING

We describe how the optimal offline algorithm that has access to perfect predictions of demand and renewable availability would minimize \( \sum_{t=1}^{T} C_{\text{net}}(\theta_t) \). Though this is not realistic, we describe the ideal problem we would like to solve primarily owing to the interesting structure of the solution. The offline generator scheduling optimization problem is

\[ \begin{align*}
\text{minimize} & \quad \sum_{t=1}^{T} \left[ a \theta_t^2 + b \theta_t + c 
+ \lambda_t^{\text{buy}} (x_t - \theta_t)^+ - \lambda_t^{\text{sell}} (\theta_t - x_t)^+ \right] \\
\text{subject to} & \quad |\theta_t - \theta_{t-1}| \leq R, \ t = 1, \ldots, T.
\end{align*} \]

The constraints only couple consecutive generation values \( \theta_t, \theta_{t+1}, \forall t \). Based on this structure, we follow a dynamic programming [3] approach: we work backward from the last time step \( T \) and recursively generate the cost function to be solved at the first time step \( t = 1 \), given the knowledge of the \( x_t \). However, this approach has a major shortcoming in that the solution space is continuous, thereby requiring us to discretize the set of allowable \( \theta_t \). We mention here that the piecewise quadratic nature of the cost function enables us to sidestep the continuous optimization problem, since the optimal point is always guaranteed to be among a finite set of points, and that the number of such points increases only as the number of time steps \( T \).