Proceedings of the Twenty-Sixth International Conference on
Automated Planning and Scheduling (ICAPS 2016)

Planning Curtailment of Renewable Generation in Power Grids

Sambaran Bandyopadhyay, Pratyush Kumar, Vijay Arya
IBM Research
sambaran.ban89@gmail.com, pratyushkpanda@gmail.com, vijay.arya@in.ibm.com

Abstract

The increasing penetration of renewable sources like solar energy add new dimensions in planning power grid operations. We study the problem of curtailing a subset of prosumers generating solar power with the twin goals of being close to a target collection and maintaining fairness across prosumers. The problem is complicated by the uncertainty in the amount of energy fed-in by each prosumer and the large problem size in terms of number of prosumers. To meet these challenges, we propose an algorithm based on the Combinatorial Multi-Armed Bandit problem with an approximate Knapsack based oracle. With real-data on solar panel output across multiple prosumers, we are able to demonstrate the effectiveness of the proposed algorithm.

Introduction

Operating large power grids involves crucial planning and scheduling decisions. The classical example is the unit commitment of generators, by which generating units (such as thermal, hydro, nuclear, etc.) need to be turned on or off at different times to match demand and supply (Campion et al. 2013). Given the size and importance of power grids, the economic and social impact of such planning is significant.

Power grids are being upgraded into what are popularly called *smart grids*. The first aspect of a smart grid is greater instrumentation to (a) collect a larger volume of data, and (b) actuate a richer set of planning actions. An example is the installation of smart meters in residential and commercial settings. Smart-meters introduce newer planning problems: For instance, when and which consumers must be targeted for incentive schemes to reduce demand based on historical consumption patterns (Chandan et al. 2014).

The second aspect of a smart grid is the push towards green and distributed generation. A popular example here is the installation of roof-top solar panels in residential homes, which can *feed-in* excess energy back to the grid. The characteristic aspect of such distributed energy resources (DER), as compared to the traditional sources, is its uncertainty (Hirth 2013). In the presence of such uncertainty, it is hard to plan unit commitment operations of traditional generators. Specifically, at times of renewable supply excess, the grid operator may enforce that consumers who produce solar power - also called *prosumers* - to decrease their production (Lew et al. 2013). This is referred to as the curtailment, wherein at each time-step the grid operator selectively refuses to buy power generated by a chosen set of prosumers.

Such curtailment must be planned and scheduled carefully with two specific objectives. First, to ensure that demand and supply match, the total collected energy must be close to but not exceed a targeted collection. Second, the curtailment across prosumers must be fair in that the fraction of energy curtailed to the energy transacted must be similar across prosumers. Note that the second objective is across time-steps thereby coupling planning decisions in time.

Curtailment planning is complicated by the uncertainty in the power generated by the prosumers. Based on solar irradiance the amount of energy produced by each panel can be estimated with a tool such as IBM Watt-Sun (Utsler 2014). However, such estimation cannot model (a) localised sources of inefficiency of the panels such as partial shading (Patel and Agarwal 2008) or dust accumulation (Al-Hasan 1998), and (b) the fraction of energy used by the prosumer and thus not fed-in to the grid. To meet this uncertainty, curtailment planning needs to trade-off between exploration and exploitation. Another challenge is the large number of prosumers which have to be simultaneously considered. Planning curtailment for thousands of prosumers can be computationally challenging, especially if the planning granularity in time is small, say 5 minutes.

In this paper, we propose a solution to curtailment planning to meet the twin challenges of uncertainty and problem size. We formulate it as a combinatorial multi-armed bandit (CMAB) problem (Chen, Wang, and Yuan 2013), where arms correspond to each prosumer and playing an arm is equivalent to not-curtailing that prosumer. Its combinatorial because at each time-step multiple prosumers may be curtailed, or equivalently a sub-set of the arms can be played. The reward structure of playing a sub-set of arms models both meeting the targeted collection and being fair. At each time-step, an approximate oracle based on Knapsack algorithm chooses the sub-set of arms that maximizes the reward based on currently estimated parameters. We propose two extensions to consider day-based contexts and daily adjustments, both motivated by observations from real data. We believe the proposed approach is novel, and combines latest results from machine learning to solve a planning problem with application to a relevant problem in the emerging domain of smart grids.

We perform experiments with real-data collected from a popular community website (Pvoutput.org 2015) of solar panels around Melbourne.[1] With various experiments we demonstrate that the proposed algorithm is effective in learning the uncertainty, optimizing for both the objectives, and scaling to large problem instances.

Mathematical Formulation

In this section, we formulate the curtailment problem mathematically and define the metrics of the optimization.

Basic Notation The set of prosumers is denoted as $N = \{1, 2, \cdots, n\}$. The time-steps for curtailment decisions is $\mathcal{T} = \{1, 2, \cdots, T\}$. The time-series of estimated generation for each prosumer $i \in N$ is $g_i = (g_i^1, \cdots, g_i^T)$. The time-series of targeted collection of energy is $D = (d^1, \cdots, d^T)$.

Uncertainty Model As described, the amount of energy that a prosumer wants to feed-in to the grid can differ from the estimation in g. This uncertainty is modelled by the fraction m_i^t, such that the amount of energy fed-in by prosumer $i \in N$ at time-step $t \in T$ if not-curtailed is $m_i^t g_i^t$.

Decision Variables At each time-step, we need to decide which prosumers to curtail. This is modelled by the boolean variables x_i^t which is 0 iff prosumer i is curtailed in time-step t, for all $i \in N$ and $t \in T$.

Constraints and Objectives The total energy collected must be below a targeted collection given by:

$$\sum_{i \in N} x_i^t m_i^t g_i^t \leq d^t, \quad \forall\, t \in T. \tag{1}$$

We can define the total collection at time t as $C^t = \sum_{i \in N} x_i^t m_i^t g_i^t$.

The following defines a fairness metric for prosumer i until time s:

$$F_i^s = \frac{\sum_{i \in N} \sum_{t=1}^{s} x_i^t m_i^t g_i^t}{\sum_{i \in N} \sum_{t=1}^{s} m_i^t g_i^t} - \frac{\sum_{t=1}^{s} x_i^t m_i^t g_i^t}{\sum_{t=1}^{s} m_i^t g_i^t}. \tag{2}$$

In words, the fairness value represents the difference of the ratio of energy collected by energy generated of all prosumers from the same ratio of a specific prosumer. Lower the value greater is the fairness for that prosumer.

Based on collection and fairness definitions, we can derive the objective function for the set of decisions x^t at time-step t as

$$\max C^t + \gamma \sum_{i \in N} x_i^t F_i^t, \tag{3}$$

where γ is a factor that weights the two objectives.

Proposed Planning Algorithm

In this section, we map the curtailment problem to a Combinatorial Multi-Armed Bandit (CAMB). Then, we will adapt an existing algorithm for the CAMB formulation. Finally, we will present extensions of the algorithm to consider time-of-day based context and daily updates.

[1]Data-sets have been included in supplementary material.

Background on CMAB

Multi-armed bandit (MAB) is extensively studied in machine learning. It is modelled as a set of arms, each having an unknown distribution of reward with unknown mean. The goal is to repeatedly play the arms such that the total expected reward closely approximates to the reward when playing the optimal set. In a combinatorial MAB (CMAB), a sub-set of arms, called a super-arm, can be played together (Chen, Wang, and Yuan 2013). Each super-arm has an unkown distribution of reward with unknown mean. Repeatedly a sub-set of arms are played and reward is maximized.

Let N be the number of arms, and let $\mathcal{S}$ be the set of all super-arms, which are 2^N in number, and let $\mathcal{T}$ denote the time-steps for playing the arms. Each arm i is characterised by a set of random variables X_i^t which is revealed if arm i is chosen in round t. The set of random variables $\{X_i^t : t \in \mathcal{T}\}$ are independent identically distributed variables with unknown expectation μ_i. Let $\mu = (\mu_1, \mu_2, \ldots, \mu_N)$ denote the vector of expectations of the arms. When a super-arm $S \in \mathcal{S}$ is played in time-step t a reward $R^t(S)$ is received, which depends on the problem instance definition, the super arm S, and the outcomes of the revealed arms namely X_i^t for all $i \in S$. The expected reward of playing super-arm S is denoted as $r_\mu(S) = \mathrm{E}(R^t(S))$.

A CMAB mapped to an ordinary MAB can be computationally infeasbile. Furthermore, most combinatorial problems are comptuationally hard. For instance in the curtailment problem, due to the constraint of Equation (1), the reward of a super-arm depends on solution to a Knapsack problem which is NP-Hard. To attend to these two issues, an efficient algorithm is proposed as the Combinatorial Upper Confidence Bound (CUCB) algorithm (Chen, Wang, and Yuan 2013). The CUCB algorithm requires an (α, β)-oracle which given a current estimate of the expectation vector μ identifies a super-arm S which is guaranteed to be an $\alpha-$approximate solution with a probability of failure β. For the CUCB algorithm to be valid, we require that the function $r_\mu(S)$ have the following two properties.

1. Monotonicity: If $\mu \leq \mu'$, then $r_\mu(S) \leq r_{\mu'}(S)$ for any super-arm $S \in \mathcal{S}$.
2. Bounded smoothness: There exists a continuous strictly increasing function f with $f(0) = 0$, and for any two μ, μ', we have $|r_\mu(S) - r_{\mu'(S)}| \leq f(\Lambda)$ for any Λ such that $\Lambda \geq \max_i |\mu_i - \mu'_i|$.

Mapping the curtailment problem on to CMAB

Each prosumer is mapped to one arm, playing which corresponds to not-curtailing that prosumer in a particular time-step. The random variables corresponding to the arms denoted X_i^t are mapped on to m_i^t which is the ratio of fed-in energy to the energy generated by prosumer i at time-step t. The expectation μ_i thus models the expectation of the ratio m_i^* for prosumer i. The reward is given by the relation:

$$\begin{aligned} R^t(S) &= -M, && \text{if } \sum_{i \in S} g_i^t > d^t, \\ &= \sum_{i \in S} (m_i^t g_i^t + \gamma F_i^t), && \text{else,} \end{aligned} \tag{4}$$

Algorithm 1 CMAB for the Curtailment Problem

1: For each prosumer i, maintain the variables (1) T_i as the total number of times the prosumer has been selected for solar injection, (2) $\hat{\mu_i}$ as the estimated mean of m_i based on the outcomes for prosumer i, (3) F_i as in eq. 2
2: Initialize the $\hat{\mu_i}'s$ as 1, and $t \leftarrow 0$
3: **While(True)**
4: $t \leftarrow t + 1$
5: For each prosumer i, set $\bar{\mu_i} = \hat{\mu_i} + \sqrt{\frac{3lnt}{2T_i}}$
6: For each prosumer i, update $weight[i] = g_i^t, value[i] = \bar{\mu_i} g_i^t + \gamma F_i$
7: Update $capacity = d^t$, and run the Approximation Knapsack to get the selected set of prosumers as: $S = ApproxKnapsack(weight, value, capacity)$
8: Play S and update all $T_i' s$, and $\hat{\mu_i}'s$
9: **End While**

where M is a large positive number, γ is from Equation (3).

The oracle is designed as an approximate Knapsack solver which solves the following problem: Given a set of items, each with a weight and a value, determine the sub-set of items to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. In our case, each prosumer is an item with weight equal to g_i^t and value equal to $m_i^t g_i^t + CF_i^t$. The weight limit is given by the bound d^t of Equation (1). We use a polynomial time approximation algorithm for Knapsack (Ibarra and Kim 1975) which guarantees an approximation factor of some set α (say 1.05) with probability 1, and thus is a $(\alpha, 1)$ approximation oracle.

We now show that the monotonicity and bounded smoothness properties. For a given S, independent of the expectation vector, it either satisfies eq. (1) or not. If it does not satisfy then $R^t(S)$ is the same for all expectation vectors. If it does satisfy, then the reward is given by $\sum_{i \in S} m_i^t g_i^t + CF_i^t$, where the m_i^t is a random variable with mean μ. This clearly is monotonic in μ. Let us now prove bounded smoothness. As, $\mu_i \leq 1, \forall i \in N$, we have $\max_{i \in N} |\mu_i - \mu_i'| \leq 1$. Let us consider λ_S such that for some $S \in \mathcal{S}$, $\max_{i \in N} |\mu_i - \mu_i'| = \lambda_S$. If $\sum_{i \in S} g_i^t > d^t$, then $|r_\mu(S) - r_{\mu'}(S)| = 0$. Otherwise, $|r_\mu(S) - r_{\mu'}(S)| = \sum\limits_{i \in S} |\mu_i - \mu_i'| g_i^t \leq \lambda_S \sum_{i \in S} g_i^t \leq G\lambda_S$, where $G = \max_{t \in \mathcal{T}} \sum_{i \in N} g_i^t$. Hence there exists a strictly increasing function $f(x) = Gx$ with satsifies the bounded smoothness property.

In the rest of the section, we present two extensions of the algorithm motivated by the specific patterns observed in real-data, as will be discussed in the experimental section.

Extension 1: Time-of-Day Based Contexts So far, we have considered that the ratios m_i^t for some prosumer i are realisations of the same random variable with unkonwn mean μ_i. However, it is practical to consider that the ratio is different for different times of the day. As an example, a prosumer may consume greater energy in the morning or there may be shadow on the panel in the afternoon. To model such variability, we divide a day into multiple contexts each characterised by an independent set of random numbers. Thus, each prosumer i and context j, we estimate the mean $\mu_{(i,j)}$ which characterises the value of ratios m_i^t for times t belonging to that context j.

Extension 2: Daily-Adjustment From real-data, we observe that there are days when collectively prosumers feed-in much lesser energy into the grid. In other words, there is a correlation between the random numbers m_i^t characteristing the feed-in across users. Unless modelled this can affect meeting the targeted collection. To effectively model such correlated reduction, we define a term ρ_t, between 0 and 1, such that the effective m_i^t for each prosumer i has a mean of $\mu_i \rho_t$. At the start of the day, the value ρ_t is initialized to the default value of 1, indicating no correlated reduction in feed-in. Based on the observations through the day, the value of ρ_t is adjusted. This effectively models day-specific patterns: For example, this can model a national holiday when the energy fed-in could be much smaller due to increased consumption.

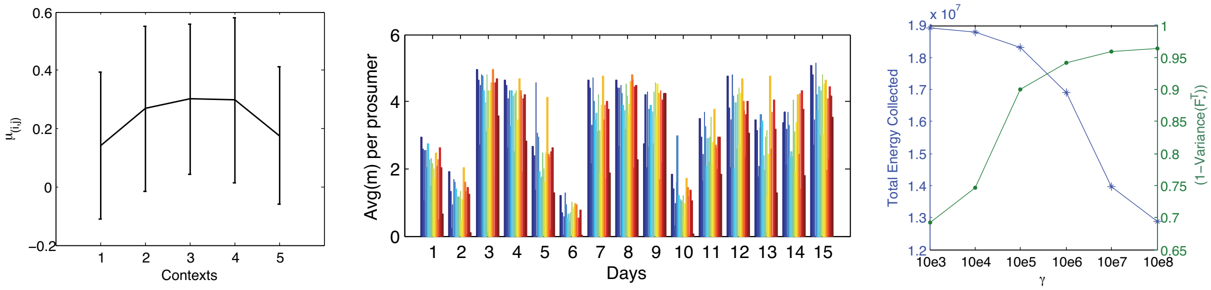
Experimental Evaluation

Data We source data from the publicly available source (Pvoutput.org 2015). This is the leading repository of recorded outputs from solar installations from around the world. As there is no automated data download feature, we manually collected data of 20 prosumers in and around Melbourne for 15 days form 1st to 15th November 2015, at resolutions of 5 minutes each from 7.30 am to 5.30 pm - a total of 36,000 data points. Note that November is a sunny month in Melbourne with significant solar generation. At each time-step we obtain both the power fed-in by the solar panels of each prosumers (denoted by $m_i^t \times g_i^t$) and the estimated power based on the solar irradiance of that day (denoted by g_i^t). In addition, we source the demand of the state of Victoria (of which Melbourne is the capital) for the same 15 days from (Australian-Market-Energy-Operator 2015). We uniformly scale this demand profile and use this as the targeted collection denoted d^t. The curtailment planning is to decide the sub-set of prosumers to curtail in intervals of 5 minutes.

Computation of error To evaluate the proposed algorithms, we compare them against an optimal clairvoyant algorithm that has full knowledge, i.e., it knows $m_i^t g_i^t$ for each time-step and i. Then we compute the ratio of the objective functions (Equation (3)) of proposed algorithm to the optimal one for each time-step. A time-series of the running mean of this ratio is given as η_A^t for algorithm A.

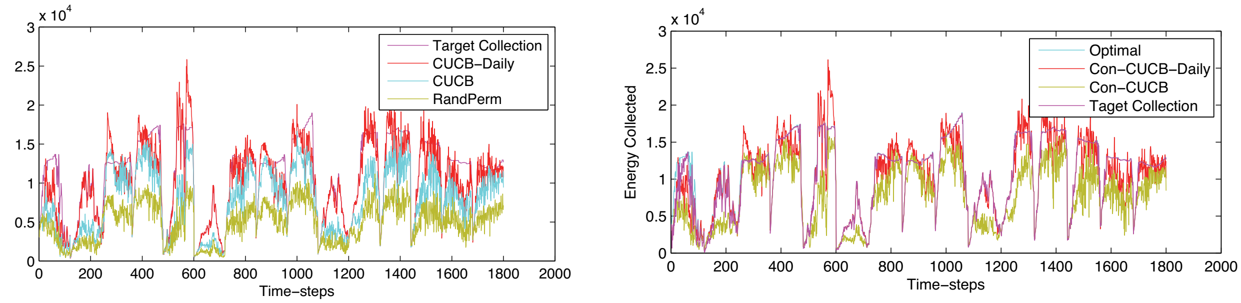
Motivation for time-of-day based contexts For a particular prosumer we plot the mean value of m_i^t where the data is divided into five contexts each of 2 hours from 7.30 am to 5.30 pm. As shown in Figure 1(a), the mean values significantly vary across the contexts.

Motivation for daily-adjustment For each prosumer we compute the mean of m_i^t for each day. As seen from the histogram of these mean values in Figure 1(b), there is a



(a) (b) (c)

Figure 1: Motivation for (a) time-of-day contexts, and (b) daily adjustment. (c) Trade-off between fairness and collected energy based on choice γ



(a) (b)

Figure 2: The collected energy with the 4 proposed algorithms, optimal algorithm, and the RandPerm algorithm.

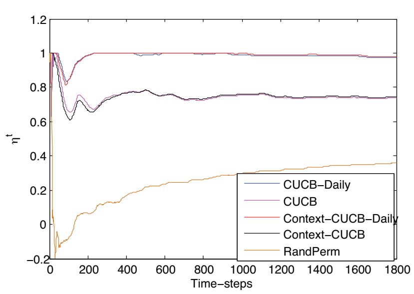


Figure 3: Accuracy of the other algorithms w.r.t. the optimal algorithm.

significant correlation in the mean values of different prosumers across time. This shows that external common factors strongly influence the behaviour of the prosumers.

Choice of γ In Equation (3) we defined γ to weight between the two components of the objective function. We now experimentally evaluate the value of γ such that the choice of prosumers is fair. To this end, we plot $1 - \text{variance}(F_*^T)$ and the total collected energy for the CUCB algorithm for different values of γ in Figure 1(c). Based on this plot we identify $\gamma = 100,000$ as a good trade-off between fairness among customers and total energy collected under the Knapsack constraint.

Performance of all algorithms We have 4 variants of the algorithm by choosing to include or exclude the two extensions for time-based contexts and daily adjustment. These are referred to as CUCB, Con-CUCB, CUCB-Daily, and Con-CUCB-Daily. In addition, we have the optimal clairvoyant algorithm and a RandPerm naive algorithm which randomly permutes the prosumers and sequentially selects as many prosumers as possible satisfying Equation (1). For each algorithm we plot the time-series of the total collection $\sum_i x_i^t m_i^t g_i^t$ against the targeted collection in Figure 2(a), (b). Note that the Optimal algorithm almost always matches the targeted collection. RandPerm on the other hand performs poorly due to no modelling of uncertainty and a greedy approach to meeting the targeted collection. Amongst the 4 variants of our algorithm, expectedly Con-CUCB-Daily most closely matches the targeted collection. W.r.t. the Con-CUCB, addition of the daily-adjustment greatly improves results as is clear in Figure 2(a) for days 6 and 10. A similar difference is observed between CUCB-Daily and CUCB. However, addition of the day-based context does not seem to have a significant impact on the results.

For the 4 algorithm variants and RandPerm we plot in Figure 3 the ratio η^t capturing the time-varying performance w.r.t. the optimal solution. An ideal value of $\eta^t = 1$ indicates that the performance of an algorithm is identical to the optimal algorithm. As noted, RandPerm performs the worst, the variants with daily adjustment perform the best, while the variants with day-based context have a marginal advan-

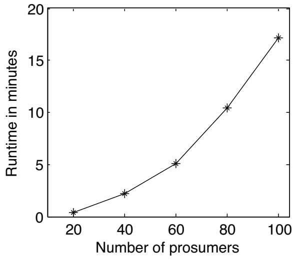


Figure 4: Run-times as a function of problem size

tage. The Con-CUCB-Daily algorithm has $\eta^t > 0.97$ for all $t > 240$, i.e., after 2 days of learning the algorithm is within 3% of the optimal algorithm. With this we conclude that the algorithm effectively learns the uncertainty in modelling the users and optimizes for the considered metrics.

Scaling of runtimes with problem size To evaluate the runtime scaling of the algorithm, we generate artificial prosumers by randomly choosing one of the prosumers and adding random noise to both the estimation g and the actual energy fed-in βg. We repeat this process to generate sets of 40, 60, 80, and 100 prosumers. For each set, we execute the Context-CUCB-Daily and plot the run-times in Figure 4 on MATLAB on a dual-core Intel Core i5 2.6 GHz processor with 8GB RAM. Note that the report run-times are for the 36000 time-steps in the 15 days. Indeed, in practice only one of these time-steps will be considered which can be a manageable run-time for a large set of prosumers.

Conclusions

In this paper, we formulated the curtailment planning of renewable sources as a CMAB. This is effective in modelling the unknown amount of energy fed-in by prosumers while optimizing for meeting the targeted collection and fairness. From real-data we recognized the need to add two extensions, namely, multiple contexts within the day and a daily adjustment. The experimental results demonstrate that the approach is able to actively learn the unknown parameters. We believe that the planning operations in today's smart grids will be dominated by the need to model uncertainty and optimize for large problem sizes, and methods such as the proposed one will be increasingly relevant.

References

Al-Hasan, A. Y. 1998. A new correlation for direct beam solar radiation received by photovoltaic panel with sand dust accumulated on its surface. *Solar Energy* 63(5):323–333.

Australian-Market-Energy-Operator. 2015. Aggregated Price and Demand Data Files. http://www.nemweb.com.au/mms.GRAPHS/DATA/DATACURRENTMONTH_VIC1.csv. Accessed: 2015-11-15.

Bandyopadhyay, S.; Narayanam, R.; Kota, R.; Petra, P. M. I. P. H.; and Charbiwala, Z. 2015. Aggregate demand-based real-time pricing mechanism for the smart grid: A game-theoretic analysis. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2554–2560.

Campion, J.; Dent, C.; Fox, M.; Long, D.; and Magazzeni, D. 2013. Challenge: Modelling unit commitment as a planning problem. In *ICAPS*.

Chandan, V.; Ganu, T.; Wijaya, T. K.; Minou, M.; Stamoulis, G.; Thanos, G.; and Seetharam, D. P. 2014. idr: consumer and grid friendly demand response system. In *Proceedings of the 5th international conference on Future energy systems*, 183–194. ACM.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, 151–159.

Garg, V. K.; Jayram, T.; and Narayanaswamy, B. 2013. Online optimization with dynamic temporal uncertainty: Incorporating short term predictions for renewable integration in intelligent energy systems. In *AAAI*.

Hirth, L. 2013. The market value of variable renewables: The effect of solar wind power variability on their relative price. *Energy economics* 38:218–236.

Ibarra, O. H., and Kim, C. E. 1975. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM* 22(4):463–468.

Lew, D.; Bird, L.; Milligan, M.; Speer, B.; Wang, X.; Carlini, E. M.; Estanqueiro, A.; Flynn, D.; Gomez-Lazaro, E.; Menemenlis, N.; et al. 2013. Wind and solar curtailment. In *Int. Workshop on Large-Scale Integration of Wind Power Into Power Systems*, 1–9.

Linnenberg, T.; Fay, A.; and Kaisers, M. 2015. Bottom-up demand response by following local energy generation voluntarily. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Patel, H., and Agarwal, V. 2008. Matlab-based modeling to study the effects of partial shading on pv array characteristics. *Energy Conversion, IEEE Transactions on* 23(1):302–310.

Pvoutput.org. 2015. PVOutput is a free service for sharing and comparing PV output data. http://pvouput.org. Accessed: 2015-11-15.

Utsler, J. 2014. The Watt-Sun Program Is Poised to Improve Weather Forecasting for Solar Energy. http://www.ibmsystemsmag.com/mainframe/trends/IBM-Research/watt-sun/. [Online; accessed 20-Nov-2015].