Distributed Rate Control for Smart Solar Arrays

Stephen Lee, Srinivasan Iyengar, David Irwin, Prashant Shenoy
College of Information and Computer Sciences
University of Massachusetts Amherst

ABSTRACT

Continued advances in technology have led to falling costs and a dramatic increase in the aggregate amount of solar capacity installed across the world. A drawback on increased solar penetration is the potential for supply-demand mismatches in the grid due to the intermittent nature of solar generation. While energy storage can be used to mask such problems, we argue that there is also a need to explicitly control the rate of solar generation of each solar array in order to achieve high penetration while also handling supply-demand mismatches. To address this issue, we present the notion of smart solar arrays that can actively modulate their solar output based on the notion proportional fairness. We present a decentralized algorithm based on Lagrangian optimization that enables each smart solar array to make local decisions on its fair share of solar power it can inject into the grid, and then present a sense-broadcast-respond protocol to implement our decentralized algorithm into smart solar arrays. Our evaluation on a city-scale dataset shows that our approach enables 2.6× more solar penetration, while causing smart arrays to reduce their output by as little as 12.4%. By employing an adaptive gradient approach, our decentralized algorithm has 3 to 30× faster convergence. Finally, we implement our distributed algorithm on a Raspberry Pi-class processor to demonstrate its feasibility on grid-tied solar inverters with limited processing capability.

ACM Reference format:
Stephen Lee, Srinivasan Iyengar, David Irwin, Prashant Shenoy. 2017. Distributed Rate Control for Smart Solar Arrays. In *Proceedings of e-Energy '17, Shatin, Hong Kong, May 16-19, 2017,* 11 pages.
DOI: http://dx.doi.org/10.1145/3077839.3077840

1 INTRODUCTION

The cost of solar energy continues to decline rapidly due to both advances in solar module efficiency and economies-of-scale in manufacturing. Today, the total average cost of energy from solar photovoltaics (PV) in the U.S. is estimated at 12.2¢ per kilowatt-hour (kWh) [31, 32], which is nearly equivalent to the average retail electricity rate of 12¢ per kWh [31]. Some have predicted that, based on current trends, the marginal cost of solar modules will eventually fall to near zero [24]. These declining costs, combined with subsidies from various states, are driving significant increases in the number and size of solar deployments. As the cost of solar module hardware decreases, solar energy cost will be dictated largely by "balance of system" costs, which capture the indirect costs incurred by utilities to incorporate renewables despite their intermittent nature. These costs include inverters, charge controllers, and energy storage devices, such as batteries, among others.

Conventional wisdom holds that there is a limit to the amount of the solar penetration, i.e., the maximum fraction of demand satisfied by solar power that the grid can handle. Since solar generation is intermittent, utilities must offset any large increases or decreases in solar output by decreasing or increasing output from other sources to compensate. However, with high penetration and variable weather conditions, fluctuations in aggregate solar output may occur too quickly to be offset from mechanical generators to offset, resulting in supply-demand mismatches. Consequently, current regulations strictly limit the number and size of grid-connected solar deployments that use net metering.

The problem faced by the grid is reminiscent of problems faced by the early Internet. Early transport protocols for network data transmissions did not include congestion control and allowed users to inject data into the Internet at arbitrarily high rates. Since network capacity was fixed, too many users sending data at excessively high rates drove the network to near congestion collapse. The imminent threat of congestion collapse led the design of TCP, a transport protocol that uses congestion and rate control to gracefully adapt sending rates upon detecting congestion to maximize aggregate goodput, prevent congestion collapse, and fairly share the Internet's available bandwidth among active flows [14].

Today's "dumb" electric grid and solar arrays are akin to the early Internet—it permits grid-tied solar systems to generate and transmit large amounts of power into the grid without regard for its current state and available excess transmission capacity. For example, on a sunny day, the cumulative output of solar deployments throughout the grid could cause a supply-side surplus that exceeds demand and causes grid "congestion". In contrast, on a cloudy day, the grid may be able to accept additional power from many solar systems that are currently forced off-grid due to strict caps.

To address this problem, we present the notion of smart "active" solar arrays that can intelligently control their solar power output—in contrast to today's passive solar arrays that simply inject the maximum amount of power they can generate at each instant based on current weather conditions. Smart solar arrays have the ability to accept signals from the grid and can increase or decrease their output ("solar rate") in response to these signals—similar to TCP which can modulate its sending rate based on congestion signals. Recent research on software-controlled smart solar inverters [28] can be used as a building block for our smart solar arrays. Our contributions are as follows.

Proportional-share Solar Rate Control: We formulate the problem of *solar rate control* that allocates the available solar capacity

e-Energy '17, Shatin, Hong Kong

978-1-4503-5036-5/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3077839.3077840

using the notion of *proportional fairness*. Our approach enables utilities to control the aggregate amount of solar output across its users by setting a limit, or weight, for each array. Each smart array then generates solar power in proportion to its weight. The key challenge for utilities is determining a *fair* weight for hundreds-to-thousands of deployments in a distributed fashion without continuously gathering fine-grained solar data from each array.

Decentralized solar rate control algorithm: We present a decentralized algorithm based on Lagrangian optimization that enables each solar array to compute its fair solar rate locally and in a distributed manner using grid signals. We also present a sense-broadcast-respond protocol to implement our decentralized algorithm into smart solar arrays while also enabling fast convergence of our algorithm to the fair rate.

Implementation and Evaluation: We evaluate our approach using a city-scale dataset and show that our distributed rate control algorithm performs similar to a centralized approach that requires full system knowledge. Our results show that our approach enables 2.6× more solar penetration, while causing smart arrays to reduce their output by as little as 12.4%. By employing an adaptive gradient approach, our decentralized algorithm has 3 to 30× faster convergence. Finally, by implementing our decentralized algorithm on a Raspberry Pi-class processor, we demonstrate its feasibility on grid-tied solar inverters with limited processing capabilities.

2 BACKGROUND

2.1 Solar Arrays

Solar panels installed on buildings can be connected to the grid through *net metering*. These grid-tied solar panels support local loads inside a building and feed the surplus power into the grid, effectively selling it back to the utility. However, solar energy generation is intermittent and highly weather dependent (see Figure 1). For example, on sunny days the amount of solar generated by a panel is at its maximum, but on overcast days the amount of solar generation may be relatively low. Thus, the amount of solar power "net metered" to grid depends on: (i) local demand from loads (ii) the solar radiation incident on the panel, which is weather dependent.

Injecting large amounts of solar power is problematic as grid operators must continuously balance supply and demand. If the total output from intermittent solar arrays fluctuates too rapidly, it can cause supply and demand mismatches. Furthermore, as solar penetration grows, the impact of intermittent solar energy makes balancing supply and demand ever more challenging.

To avoid using "excessive" amount of solar power from being injected into the grid, many governments strictly regulate grid solar connections [2]. Many states in the US set hard limits by passing laws to regulate the number of solar panel connections. Limiting the solar capacity limits the stochasticity seen from these distributed sources, which in turn makes matching supply and demand a more manageable problem despite intermittency. For example, while the state of Virginia has a cap of 1%, a similar law exists in Massachusetts that caps the solar at 2% of the total power generation. Importantly, these caps are generally based on the rated maximum capacity of a solar installation, regardless of what it actually generates. That is, the caps assume a solar panel is generating at its maximum capacity all the time.

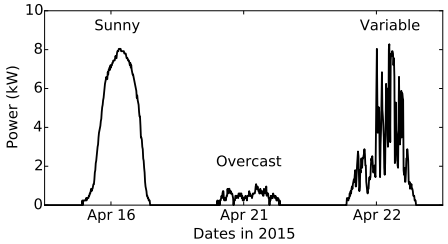


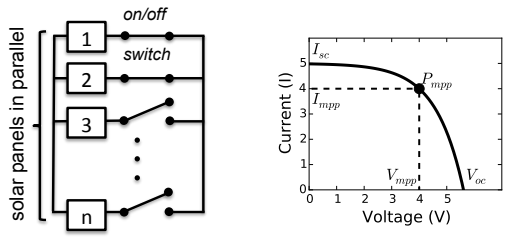
Figure 1: Solar power output varies based on time of day and local weather conditions.

In this paper, we propose an alternate approach — smart solar-powered arrays that are capable of self-regulating their output in a grid-friendly fashion. Our smart solar arrays can control their generation rate by backing off when supply exceeds demand (more precisely, the aggregate solar output is greater than some threshold), and increasing the rate when needed. The idea is similar to rate control of network flows in TCP, where sources back off when there is congestion in the network and increase the rate when network capacity is available. While network rate is given by the bandwidth and measured in Mbps, solar rate is given by the solar power output and measured in kilowatts (kW). We argue that solar rate control has the potential to permit a much larger solar capacity to be installed, thereby increasing solar penetration. Solar rate control also provides grid operators with an additional control "knob" when continuously matching supply and demand.

Relation to energy storage: An alternate solution to managing high solar intermittency is to use energy storage. Energy storage, such as lithium-ion batteries, can absorb surplus energy from solar arrays and feed the excess power back to the grid when there is a deficit [15, 21]. Today, the cost of energy storage remains high, and large-scale energy storage deployments remain economically infeasible. However, technology improvements will make energy storage feasible in the future. It is important to note that energy storage and solar rate control are *complementary approaches* for handling high solar penetration. Both technologies can coexist with one another, and neither obviates the need for the other. For example, even with large scale storage deployments, solar rate control is necessary—since storage batteries, which have finite capacity, may reach full charge and require solar rate control to temporarily reduce excess output. This is similar to "supply-side" demand response, where solar output is temporarily reduced on rare occasion when supply exceeds demand and batteries can not absorb the surplus. Similarly, even with widespread solar rate control deployment, energy storage can be used to locally store excess output that can not be net-metered to the grid. Finally, smart solar arrays also offer a form of "reserve capacity" where their output can be ramped up if there is a sudden increase in demand, a role that energy storage can also play. While energy storage-based techniques have received significant attention in recent years [3, 9, 23], solar rate control is a newly emerging topic that has not seen much attention and is the focus of this paper.

2.2 Why is Solar Rate Control Feasible?

Interestingly, practically every solar panel today, as well as solar arrays, have the ability to control their solar output. At an array



(a) Discrete Rate Control (b) Continuous Rate Control

Figure 2: Rate control approaches in solar panels.

scale, this can be trivially done in discrete steps by dynamically connecting and disconnecting individual panels. Figure 2(a) shows an array where panels are connected in parallel and a program switch can be used to dynamically disconnect k out of n panels, thereby providing *discrete* control[1].

Even at the granularity of a single panel, it is possible to control the output of the panel. The output of photovoltaic solar is given by its *I-V curve* depicted in Figure 2(b). Given a certain amount of solar irradiance, the I-V curve shows all possible operating points of the panel for that solar irradiation. Specifically, any voltage on the curve can be chosen and the panel will then produce the corresponding current. Since power is defined as the product of current and voltage i.e. $P = I \cdot V$, the panel actually can provide a different power output based on the choice of voltage. In general, panels operate at a voltage V at the knee of the curve, which yields the maximum output. The point where the panel generates the maximum power is called the maximum power point (MPP).

However, there is no particular reason to operate a solar panel at its maximum power point. It is possible to pick other values of V [12, 28], using buck-boost converter, which are akin to "backing off" and producing an output less than the output at MPP. Thus, any solar panel's output can be altered by changing its operating voltage. Our smart solar panels are built on this idea. We assume the presence of software controls that enable the output to be lowered below the MPPT, and thus control the power output of the panel. This mechanism enables *continuous* rate control to limit the power injected to the grid.

Modern inverters are beginning to offer more configurability and in the long run, we expect them to expose rate control mechanisms [28]. Both the discrete control above, or the continuous control can be used to regulate the rate. Given smart solar panels connected to the grid, our goal is to control the solar output in order to provide higher control over distributed solar-powered systems.

3 SOLAR RATE CONTROL

The problem of controlling solar power is similar to the rate control problem in communication networks [16, 20]. This body of work proposes an optimization framework for determining the rates allocated to different network flows given network capacity constraints. These ideas from network rate control were first applied to the power grid scenario by Ardakanian et. al. albeit in a different context—controlling the rate of electric vehicle charging [4]. In our case, we use these principles from networking [16, 20] to address the problem of *solar rate control*. Next, we present the problem of solar rate control. We then outline our design objectives and assumptions.

3.1 Centralized Problem

We first formulate our solar rate control problem as a centralized optimization problem. The centralized problem requires knowledge of the load at the feeders/transformers level and the current generation rate of individual solar installations in order to compute the solar allocation rate while adhering to certain grid constraints. The allocation rate should not only maximize the individual user's output but also maximize the overall grid utilization.

Intuitively, we want to limit the aggregated distributed solar generation to a certain *capacity*. This leads to the problem of apportioning the capacity among different solar arrays to determine the generation rate for each array. Note that the grid demand and solar generation output is time varying, and may change over the day. Thus, at each time t, the optimization problem needs to recompute the capacity and the allocation rate for each solar array . For simplicity, we describe the optimization formulation for a single time step.

We consider a distributed grid transmission network with a set of transmission feeders F, transformers K and smart solar arrays S. Electric power is transmitted from the power station to substations at high voltages. At the distribution substation i.e. low voltage (LV) feeder, voltage is stepped down and distributed to transformers, wherein it is further stepped down before it is transmitted to residential users. Thus, the smart solar arrays are connected to the LV feeder via a transformer. Formally, we say that the smart solar array s is connected to a LV feeder f, if $s \in S(k)$ and $f = F(k)$, where k is the transformer located in between s and f. We model the key characteristics of our problem as follows:

Transformer constraint: Power flow at the transformer level can be bi-directional and the maximum power flow at the transformer is dependent on the transformers rating C. The transformer rating is between $-C$ to C kVA, where the negative sign indicates reverse power flow from the transformers to the feeders. Usually, the transformers are right-sized to ensure that the load at the transformers does not exceed its rating. However, high solar penetration in residential homes may cause reverse power flow and the following constraint must be satisfied to maintain grid stability.

$$\sum x_s \leq load_k + C_k \quad \forall s \in S(k) \text{ and } k \in K \tag{1}$$

where x_s is the solar generation rate of the smart solar array $s \in S(k)$ and $load_k$ is the aggregate load from the residential homes in transformer k.

Feeder constraint: Most residential LV feeders are not equipped with infrastructure to allow reverse power flow i.e. electricity does not flow from an LV feeder to a medium voltage transmission line and thus obeys the following constraint

$$\sum x_s \leq load_f, \quad \forall s \in S(f) \text{ and } f \in F \tag{2}$$

where $load_f$ is the load at the feeder f, and $S(f)$ are the smart solar arrays in feeder f.

[1]Typical rooftop solar installation is 5kW (20 panels). Thus, we can control the power output in 5% (250W) increments.

Grid capacity constraint: The grid utility may cap solar output to reduce variability in grid or due to legislative reasons [2]. The aggregate solar generation output may be capped at a fraction of the aggregate grid demand

$$\sum x_s \leq capacity, \qquad \forall s \in S \tag{3}$$

where *capacity* is defined as a fraction of the total power demand at the grid level.

Solar PV constraint: The maximum power generated by a solar panel lies in the interval $[0, x_s^{mppt}]$, where x_s^{mppt} is the MPPT rate of the solar PV and is defined as

$$0 \leq x_s \leq x_s^{mppt} \quad \forall s \in S \tag{4}$$

Note that (1), (2) and (3) can be combined and represented as a single inequality

$$R\mathbf{x} \preceq \mathbf{c} \tag{5}$$

where $R \in \mathbb{R}^{m \times n}$ matrix, with m combined constraints from (1), (2) and (3) and n smart solar arrays; $\mathbf{x} \in \mathbb{R}^{n \times 1}$ vector is the set of smart solar arrays; $\mathbf{c} \in \mathbb{R}^{m \times 1}$ vector captures the capacity constraints; and finally, $\preceq$ represents the generalized inequality of vectors. R can be represented as:

$$R_{is} = \begin{cases} 1 & \text{if } s \in S \text{ is present in the } i^{th} \text{ constraint} \\ 0 & \text{otherwise} \end{cases}$$

Remember our goal is to take some aggregate capacity and apportion it among individual solar installations. Thus, our objective is to maximize the total utility of the individual smart solar arrays $U_s(x_s)$; subject to constraints (4) and (5). To summarize, our optimization problem can be defined as:

$$\begin{aligned} \max_{x_s} \quad & \sum_{s \in S} U_s(x_s) \\ \text{subject to:} \quad & R\mathbf{x} \preceq \mathbf{c} \qquad \text{and,} \\ & 0 \leq x_s \leq x_s^{mppt} \qquad \forall s \in S \end{aligned}$$

We refer to the above problem as the *primal* problem. We assume that the utility function is strictly concave, increasing and twice differentiable. Since each constraint is convex, a unique maximizer exists and solving the optimization problem generates a solar allocation that is optimal.

The centralized optimization problem discussed earlier is mathematically tractable. However, solving the optimization necessitates a prohibitively high communication overhead, as it requires two-way communication infrastructure between the smart solar arrays and the control center. Moreover, an increase in solar array deployments will increase the coordination overhead between the control center and smart solar arrays to compute the solar allocation rate. Hence, in section 4, we formulate a distributed approach that solves the above optimization problem to mitigate some of the issues in the centralized approach.

3.2 Design Objectives

3.2.1 Maximize utility to end-users and grid. Solar panels are *net-metered* and the amount of electricity supplied to the grid earns residential customers billing credits. To model the benefit of net metering, we attribute a utility function $U_s(x_s)$ to the user for generating solar output at rate x_s. From the user's perspective, each user would like to maximize its own utility. However, from the grid perspective, the utility function should also maximize the overall utilization of the network.

We explore two utility functions, *non-weighted* and *weighted*, described in Kelly et. al. [17], which maximizes both the grid and the user's utility function. The non-weighted utility function, $U_s(x_s) = \log(x_s)$, provides equal utility regardless of the size of the solar panel. Since, $\log(x_s)$ is a strictly increasing function, an increase in solar output x_s denotes an increase in the *utility*. On the other hand, the weighted utility, $U_s(x_s) = w_s \log(x_s)$, provides additional benefit to users for installing larger solar panel, where weight w_s represents the weight corresponding the size of the solar panel. Both the utility functions are increasing, strictly concave and continuously differentiable.

3.2.2 Fairness in solar rate allocation. We are interested in an allocation that is fair to the user. In our paper, we use a utility function that provides *proportional fairness* and *weighted proportional fairness*. Any feasible allocation vector $\mathbf{x}$ is proportionally fair, if for any other feasible rate vector $\mathbf{y}$, the aggregate of proportional change is non-positive i.e.

$$\sum_{s \in S} \frac{y_s - x_s}{x_s} \leq 0 \tag{6}$$

Similarly, any feasible allocation vector $\mathbf{x}$ is *weighted proportionally fair*, if for any other feasible vector $\mathbf{y}$ the following holds.

$$\sum_{s \in S} w_s \frac{y_s - x_s}{x_s} \leq 0 \tag{7}$$

As shown in [16], the logarithmic utility function discussed above achieves proportional fairness and the allocation vector obeys the fairness property (6). In addition, it is shown that proportional fairness is *pareto optimal*, since increasing a user's allocation will decrease allocation of another user.

4 DISTRIBUTED RATE CONTROL

The centralized problem discussed in the previous section has three key drawbacks in practice. First, it requires full knowledge of the maximum generation output (MPP) of all grid-connected smart solar arrays. Second, the control center requires knowledge of the grid's network topology in order to compute the solar rate. Third, a two-way communication needs to be established between the control center and smart solar arrays for controlling the solar rate. Hence, we reformulate the centralized optimization problem to an equivalent distributed optimization problem, which can then be solved locally by smart solar arrays and eliminate some of the disadvantages of the centralized approach. In contrast to the centralized approach, the distributed algorithm does not require knowledge of the grid's network topology and eliminates the need to share local information.

4.1 Dual decomposition

We use the *dual decomposition* approach to divide the centralized optimization problem into smaller subproblems. Note that the optimization problem has a coupling constraint (5), which prevents

solving each subproblem independently. Clearly, without the coupling constraint, each user can maximize its utility independent of each other, thus maximizing the aggregate objective function. Below, we present the Lagrangian dual problem, which relaxes the coupling constraint using *control prices* (Lagrangian multipliers) and thus allows solving the problem as independent subproblem.

We define the Lagrangian of our optimization problem and consider *control* prices λ to relax the coupling constraint

$$\begin{aligned}\mathcal{L}(x,\lambda) &= \sum_{s\in S} U_s(x_s) - \sum_{l\in L}\lambda_l(y_l - c_l)\\ &= \sum_{s\in S}(U_s(x_s) - x_s q_s) + \sum_{l\in L}\lambda_l c_l\end{aligned}$$

where l denotes the row number and L is the total number of constraints in matrix R; and

$$y_l = \sum_{s\in S} R_{ls}x_s \qquad \forall l \in L$$

$$q_s = \sum_{l\in L} R_{ls}\lambda_l \qquad \forall s \in S$$

Thus, the *Lagrangian dual* problem can be formulated as:

$$\mathbf{D}(\lambda): \min_{\lambda\geq 0} \quad \sum_{s\in S} V_s(x_s,\lambda_s) + \sum_{l\in L}\lambda_l c_l \tag{8}$$

$$\text{subject to:} \quad \lambda_l \geq 0 \qquad \forall l \in L \tag{9}$$

where,

$$V_s(x_s,\lambda_s) = \max_{0\leq x_s\leq m_s}(U_s(x_s) - x_s q_s) \qquad \forall s \in S \tag{10}$$

As discussed earlier, the utility function (U_s) is strictly concave. Since the sum of concave function U_s is concave and the linear constraints are concave, strong duality holds i.e. the primal and the dual solutions are equal. Hence, solving the dual problem solves our original primal problem.

We solve the dual problem using the gradient projection method. Note that for a fixed λ, the dual problem is completely *separable* in x_s and each subproblem in x_s can be maximized independently by each smart solar array using (10). In particular, for a given price λ, a unique maximizer exists that maximizes (10). Since the utility function U_s is continuously differentiable, using the Karush-Kuhn-Tucker (KKT) theorem[2], the unique maximum x_s^* is given by

$$x_s^* = \min\{\max\{1/U_s'(x_s), 0\}, x_s^{mppt}\} \tag{11}$$

where U_s' is the derivative of the utility function U_s.

The control prices (λ) manage the subproblems and are computed by the master algorithm that solves the dual problem. The master algorithm computes the prices by determining λ that minimizes the objective function in (8). This is done by updating λ using the gradient $\nabla D(\lambda)$ given by

$$g_l = \frac{\partial}{\partial\lambda_l}D(\lambda) = c_l - y_l \tag{12}$$

The gradient projection algorithm solves the dual problem iteratively. At each iteration, each subproblem is solved parallely, and the master algorithm updates the control prices in opposite direction of the gradient such that

$$\lambda_l(t+1) = \max\Big\{\lambda_l(t) - \gamma(c_l - y_l), 0\Big\}, \quad \forall l \in L \tag{13}$$

where, $\gamma > 0$ is an appropriate step size.

4.2 Choosing step size

Our algorithm is similar to the distributed algorithm described in [4] and guarantees to converge as ∇D is Lipschitz continuous[3] and bounded, provided the step size is appropriately selected. In other words, the convergence of the distributed algorithm is sensitive to the step size used for updating the control prices. While a big step size may cause the algorithm to oscillate around the optimal solution, a small step size may increase the number of iterations required to converge to the solution. Here, we discuss two approaches we used to select a step size to solve the dual problem.

4.2.1 Fixed gradient. At each iteration, the master algorithm updates the control prices using the gradient controlled by a fixed step size parameter using (13). As shown in [4], the solution generated by the distributed algorithm converges to the primal-dual optimal when the step size satisfies the following condition

$$0 < \gamma < 2/\bar{\alpha}\bar{L}\bar{S} \tag{14}$$

where $\bar{\alpha} = max_s\{-1/U_s''(x_s)\}$; $\bar{L} = max_s\{\sum_{l\in L} R_{ls}\}$ and $\bar{S} = max_l\{\sum_{s\in S} R_{ls}\}$.

4.2.2 Adaptive gradient (AdaGrad). In contrast to the fixed gradient, the adaptive gradient modifies the step size as a function of time and updates the control prices $\forall l \in L$ as follows

$$\lambda_l(t+1) = \max\left\{\lambda_l(t) - \frac{\gamma}{\sqrt{G_l(t)+\epsilon}}\cdot g_l(t), 0\right\} \tag{15}$$

where $G_l(t) = \sum_{i=1}^{t} g_l^2(i)$ is the sum of the squares of the gradients w.r.t. λ_l up to iteration t; and $\epsilon = 1e^{-8}$ is a smoothing term to avoid division by zero error. Note that the accumulated sum $G_l(t)$ grows with the number of iterations, which in turn causes the step size to shrink. The benefit of Adagrad is it is not very sensitive to the initial step size, and any appropriate step size converges in reasonable amount of time. The convergence guarantees of Adagrad is well studied and the algorithm converges to the optimal solution [11]. Empirically, Adagrad converges faster than the fixed gradient approach and we evaluate both of them in our distributed algorithm.

4.3 System Design

Having presented the distributed algorithm that solves our rate control problem, we next describe our assumptions and the *Sense-Broadacast-Respond* protocol — a round-based protocol. We assume that power flows unidirectionally from the power station to the feeders. However, below the feeder power flow is bi-directional in transformers. Further, we assume the solar arrays have the capability to receive control signals and adjust its rate accordingly.

In our proposed protocol, each round maps to the iterations the distributed algorithm takes to converge to the optimal solution. In each round, prices are computed using (13) and sent to individual

[2]KKT conditions are first order necessary conditions for a nonlinear program to yield a solution that is optimal

[3]Lipschitz continuous guarantees existence and uniqueness of a solution

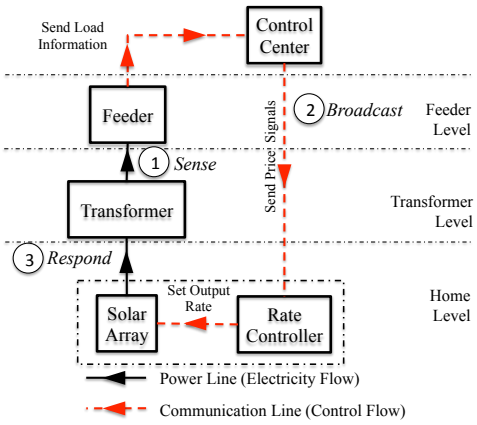


Figure 3: *Sense - Broadcast - Respond* protocol communication among the feeder/transformer level sensors, the control center and the smart solar arrays.

smart solar arrays to modulate their power outputs. To better illustrate our *Sense-Broadcast-Respond* protocol, we describe the steps on how the control center communicates with the smart solar arrays to rate control its power output (see Figure 3).

4.3.1 Sense. Sensors at the feeder and transformer capture the load at each time interval. The feeder then communicates the captured information to the grid's control center using Algorithm 1. Note that the aggregate load sensed at the feeder is the combination of the uncontrolled load from buildings and the regulated power from solar panels and is equivalent to the gradient $(c_l - y_l)$ presented in (12).

Algorithm 1 Feeder/Transformer's algorithm

1: **while** True **do**
2: **sense** $load_f$
3: **send** $load_f$ information to the control center
4: **wait** for the next clock tick
5: **end**

4.3.2 Broadcast. The utility's control center receives the load from the feeder or transformer and computes the control prices using Algorithm 2. The control prices is adjusted using (13) or (15). Next, the computed control prices are broadcasted to all smart solar arrays.

Algorithm 2 Utility's control algorithm

Input: γ
1: **while** True **do**
2: **receive** load from feeders/transformers $\forall f, k$
3: **compute** gradient g_l based on the load
4: $\lambda_l := \max\{(\lambda_l - \gamma * g_l), 0\}$ ▷ update control prices
5: **broadcast** prices to solar $s \in S(l)$, in constraint l
6: **wait** for the next clock tick
7: **end**

Characteristics	Value
Num. of Electric meters	11,186
Electric meter granularity	5 minutes
Num. of Feeders	29
Num. of Transformers	1108
Transformers rating(kVA)	5 to 750
Duration	12 months

Table 1: Key characteristics of the dataset

4.3.3 Respond. The smart solar array consists of an identifier pair that associates the array with its parent feeder/transformer. When a smart solar array receives the broadcasted control prices, it computes the rate using (11). The identifier aids in associating the prices relevant to the smart solar array. After the rate is computed, the smart solar array sets its generation rate.

Algorithm 3 Smart solar array's algorithm

1: **while** True **do**
2: **receive** control price vector λ
3: $q_s := \sum_{l \in L} R_{ls} \lambda_l$ ▷ aggregate price in l
4: $x_s := \text{argmax}_{0 \le x_s \le m_s} (U_s(x_s) - x_s q_s)$
5: **set** solar generation rate for x_s
6: **wait** for the next clock tick
7: **end**

5 EVALUATION

In this section, we describe the dataset and experimental setup for evaluating our distributed algorithm with different utility functions.

5.1 Dataset

For evaluation, we use the smart meter data gathered from a small city in the New England region of the United States. The dataset consists of smart meter data from 11,186 residential homes. Apart from electricity consumption, we also have the electric grid distribution network information — consisting of the feeders-to-transformers-to-meters connections. Table 1 shows a brief description of the dataset characteristics and was obtained from the authors of [13].

The dataset also contains solar power generated from a single residential home. To generate solar power dataset for multiple homes, we first normalize the solar power output using its maximum output for the year. Second, we assume the solar installation sizes to be in the range of 4 to 10 kW. Next, we scale the normalized solar output with the uniformly generated points for all homes from this range.

5.2 Experimental Setup

We run our evaluation for three days in the month of April that consists of different solar profiles (see Figure 1) unless otherwise stated. These solar profile patterns are representative of the different fluctuations observed over a year. Along with the solar profiles, we use the load profile from the corresponding dates as an input to our distributed algorithm.

Our distributed approach takes step size γ as an input to the parameter. For the fixed gradient approach, we use $\gamma = 2/\bar{\alpha}\bar{L}\bar{S} - \epsilon$,

as this is the maximum step size to guarantee convergence (14). As discussed earlier, the adaptive gradient (Adagrad) is insensitive to the initial step size. We use $\gamma = 0.5$ as the step size for the Adagrad approach. For our experiments, we limit the solar capacity to 15% of the aggregate demand observed at grid level. The time step size is 5 minutes (granularity of the dataset). In addition, instead of reinitializing the control prices at every time step, we use the control prices of the previous time step as an input for the next time step.

We use the *cvxpy* library — a python based convex optimization library — to solve the centralized formulation. Internally, the *cvxpy* solver uses *cvxopt* solver to find the optimal solution. Separately, for the distributed scenario we use python to simulate the environment.

5.3 Metrics

5.3.1 Fairness Metric. To assess the fairness of our algorithms, we use the *Gini coefficient* to measure the inequality in allocation distribution. The Gini coefficient is a widely used metric in economics to show the distribution (inequality) of income among the residents of a country. The value for the coefficient is between 0 (perfect equality) and 1 (perfect inequality). Mathematically, it is given by (G),

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2\sum_{i=1}^{n}\sum_{j=1}^{n}x_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2\cdot n\sum_{i=1}^{n}x_i} \tag{16}$$

where, in our case, x_i is the rate allocated to user i and n is the total number of grid-tied solar installations.

5.3.2 Variability Metric. Due to solar intermittency, volatility of the load profile observed at the grid level increases with the introduction of solar energy. This increased volatility makes grid operation of matching the demand with supply more challenging, thereby reducing power quality (i.e. more voltage fluctuations). This volatility can be reduced by controlling the solar output. We use *variability* metric ($\mathcal{V}$) to determine the impact of controlling the rate of solar output and is measured by taking the standard deviation of the successive difference of the power values

$$\mathcal{V} = \sigma(\Delta P) \tag{17}$$

where, P is a vector representing the power generated during the day; ΔP represents the difference between successive values in P; and σ represents the standard deviation function. Higher value indicates more variability.

5.4 Experimental Results

5.4.1 Impact on grid demand. We assume 5% solar penetration at each feeder i.e. 5% of residential homes have solar panel installations. We compare our approach against no rate control scenario i.e. each solar panel generates power at its maximum value (MPP).

Figure 4 shows the impact of our distributed rate control on the aggregate grid demand. The aggregate grid demand profiles usually have two peaks — one in the morning and the other in the evening (Figure 4 (a)). The aggregate grid demand with increased solar penetration with no rate control resembles a sitting duck — also known as the *duck curve* — and causes *ramp up* and *ramp down* problems [1]. Our algorithm ensures that the net demand with solar power never crosses the solar cap set by the grid. The solar cap alleviates the ramp down and ramp up problems in power generation due to high solar penetration, thereby reducing the need for expensive peaking power plants. This is clearly seen in Figure 4(a), where the ramping up/down need is cut in half.

Load Profile	Apr 16	Apr 21	Apr 22
Grid	0.079	0.076	0.069
Grid + No rate control	0.09	0.079	0.226
Grid + Rate control	**0.084**	**0.079**	**0.145**

Table 2: Variability metric for different days in 2015

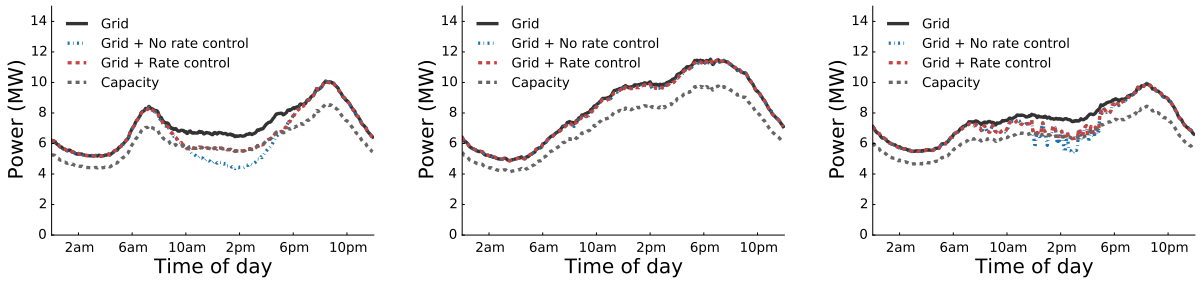
Usually, solar generation on an overcast day is low. Hence, the overall solar energy generated never exceeds the capacity mandated at the grid level (Figure 4(b)). In contrast, Figure 4(c) depicts a demand profile with variable solar generation, with generation greater or less than the capacity during the different times of the day. Our distributed algorithm adjusts the rate such that it doesn't exceed the solar capacity or the solar array's maximum generation rate .

We observe a similar behavior at the feeder level (see Figure 5). Apart from the results shown here, we also ran our simulation for solar penetrations higher than 5%. Even when the maximum solar generation capacity exceeds the local demand, our algorithm limits the rate such that local feeder constraints are met.

Next, we show the impact on variability with and without rate control mechanisms. We compute the variability in the demand curve using (17). We observe that the net demand seen by the grid with rate control is less variable compared to no rate control mechanisms. Table 2 shows the variability metric for three representative solar profiles Figure 4. Note that introduction of solar energy (regulated or unregulated) increases the variability — as shown by the increased values of the variability metric. However, the variability is much lower with rate control than without it. Moreover, with rate control the load profile at the grid level is either less or equally variable compared to no solar scenario.

Result: *Our distributed approach limits the aggregate solar generation output to available solar capacity. Moreover, it decreases the variability in the aggregate grid demand*

5.4.2 Impact of utility function on solar rate. We analyze the behavior of weighted and non-weighted utility functions of our rate control algorithm on different panel sizes. Clearly, at the grid level, the output of both the utility functions remain similar as it maximizes both the grid's and user's utility simultaneously. However, the rate allocation generated by the utility functions for individual solar panels would differ based on the size of the solar panel. This is trivially true for the weighted scenario as the allocation is proportional to the size of the panel. In the non-weighted scenario, a smaller sized panel might have reached its maximum generation capacity, thereby allowing larger panels to generate more power. We plot the rate allocation observed on a sunny day for different sized panels (see Figure 6). As expected, in the weighted scenario, we observe each panel *backs off* its generation rate proportional to the panel size. Whereas, in the non-weighted scenario, each



(a) Sunny day (Apr 16) (b) Overcast day (Apr 21) (c) Variable day (Apr 22)

Figure 4: Impact of rate control on the aggregate grid demand for different days. With 5% solar penetration and no regulation the solar output exceeds the solar capacity. Our distributed rate control algorithm caps the power output to the desired level.

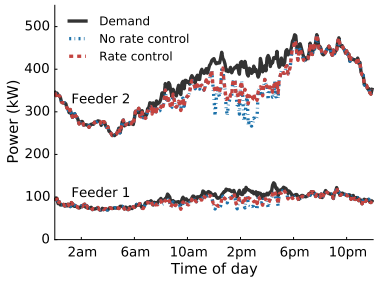
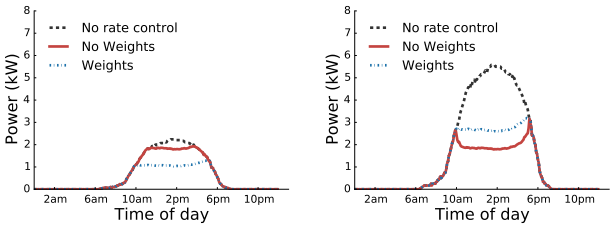


Figure 5: Impact of rate control in two feeders with 5% solar penetration. Feeder 1 has fewer homes comparatively.



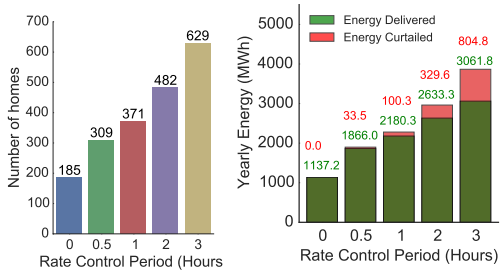
(a) Small-sized panel (b) Large-sized panel

Figure 6: Rate allocation for different sized panel. In weighted allocation, each panel backs off its rate proportional to their panel size. However, a non-weighted allocation treats each panel equally and a small-sized panel may generate power equal to a larger sized panel.

panel generate power at a similar rate (unless its maximum rate is reached for smaller panels).

Result: *Small sized panels benefit more with non-weighted utility, while weighted utility is favorable to bigger panels*

5.4.3 Impact of solar power control policies. As discussed in section 1, several states in the US have enforced hard limits on the amount of solar energy net metered into the grid. However, these hard caps are quite conservative and do not exploit the available solar potential. Moreover, these policies limit the adoption of solar by residential homeowners. Here, we analyze the change in the



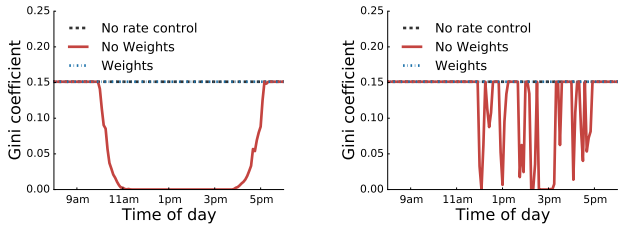
(a) Number of homes (b) Solar energy

Figure 7: Impact of average daily rate control period.

number of homes adopting solar installations and the amount of solar energy generated with different rate control policies. Unlike other experiments, we also assume all panels to be of equal size (5 kW) and evaluate for the entire year 2015. We define the rate control policy as the average hourly curtailment of solar energy per day. For this experiment, we choose rate control policies between 0 to 3 hours.

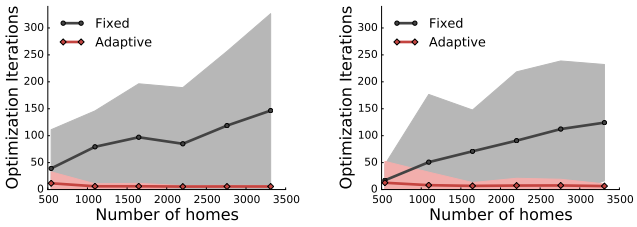
Figure 7 (a) shows the number of homes that can install solar panel systems with different rate curtailment policies. With no daily curtailment, a maximum of 185 homes may be permitted to install solar panels of size 5 kW. However, if we allow just 30 minutes of average daily curtailment, the number can be increased to 309 homes. As we increase the rate curtailment to an hour, we can double the number of homes adopting solar panel systems. Furthermore, with 2 and 3 hours of average daily curtailment we can have 2.6× and 3.4× increase in the number of homes having solar panel systems respectively.

Figure 7(b) quantifies the amount of energy delivered to and curtailed by the grid with different rate curtailment policies. As discussed earlier, a maximum of 185 homes can install solar panel system when the total installation size is limited to the minimum load observed for the entire year. The total solar energy supplied to the grid from these distributed sources is around 1137 MWh. However, increasing the average daily curtailment period to 30 minutes, the solar energy delivered to the grid increases by 64%, with solar energy curtailment of just over 1.8%. Furthermore, increasing the curtailment period to an hour, the installed panels can contribute



(a) Sunny day (Apr 16) (b) Variable day (Apr 22)

Figure 8: Fairness comparison between no rate control mechanism, weighted and non-weighted utility functions.



(a) Sunny day (Apr 16) (b) Variable day (Apr 22)

Figure 9: Convergence plots of fixed and adaptive gradient using our distributed algorithm.

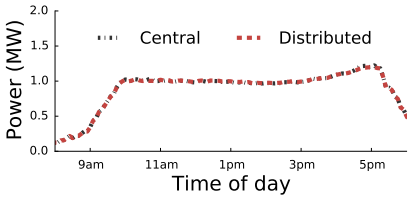


Figure 10: Aggregated solar power comparison between the centralized and distributed algorithms.

almost doubles the amount of energy to the grid with solar energy curtailment of 4.6%. Similarly, with 2 and 3 hours of average curtailment period, installed solar panels contributes around 2.3× to 2.7× to the grid, with energy curtailment of around 12.5% to 26.2% respectively. Clearly, increasing the rate control period increases the solar energy utilization in the grid provided a small fraction of curtailment is allowed. Intuitively, a solar panel only reaches its peak generation capacity around noon on a clear sunny day. For most periods, the power output is a fraction of the total installation size. Thus, increasing the aggregate installation size increases the amount of solar energy utilized by the grid.

Result: *Increasing the rate control period, increases the overall solar utilization in the grid. In particular, an average curtailment of 2 hours enables 2.6× more solar penetration, while causing smart arrays to reduce their output by as little as 12.4%.*

5.4.4 Fairness in solar rate allocation. Our allocation scheme ensures that generation rates of all net-metered solar arrays are assigned in a fair manner — even when solar generation and grid's capacity vary. We use Gini coefficient, a metric for statistical dispersion, to measure the fairness of our proposed approach.

We compare the two utility functions — weighted and non-weighted — with a solar panel generating power at its maximum capacity (MPP) i.e no rate control. We evaluate for three days with 5% solar penetration at each feeder level (see Figure 8). With no rate control, all panels will generate power at its maximum rate, wherein the rate is proportional to its installation size. Thus, the Gini coefficient is a constant value, that indicates the inequality in the distribution of the panel sizes. Similarly, in the weighted scenario, the rate allocated would be proportional to the size of the panel. Thus, the Gini coefficient does not change with time and is similar to the MPP scenario.

In contrast, the Gini coefficient will not be constant in the non-weighted scenario as depicted in Figure 8. As shown in Figure 8 (a), until 10 am, the Gini coefficient is equivalent to the weighted scenario. This is because even when all the panel generates power at its maximum rate it is not able to meet the total available solar capacity. However, as the day progresses, the total generation exceeds the maximum solar capacity and all the panels are allocated equal rate, which causes the Gini coefficient to reach zero. On an overcast day (not shown in figure), the maximum available capacity is never reached as all the panels operate at MPP. Hence, Gini coefficient is constant. Separately, on a variable day(see Figure 8 (b)), the Gini coefficient varies as it depends on the amount of available capacity met by the generated solar discussed earlier.

Result: *Both weighted and non-weighted utilities can be used to achieve fairness in rate allocation.*

5.4.5 Convergence of our distributed approach. As discussed earlier, the convergence of the distributed algorithm is dependent on the step size. Theoretically, a large step size will oscillate and not converge to the optimal solution, while a small step size will take a long time to reach the optimal solution. Here, we empirically, compare the performance of two step-size selection methods — i) Fixed gradient, and ii) Adaptive gradient (AdaGrad). We select step sizes and evaluate for all three days as described in the experimental setup section. Moreover, we assume that the distributed approach has converged if the objective function's output within two consecutive iteration is less than $1e^{-5}$.

Figure 9 shows the convergence results of the distributed algorithm using different step size methods. Note that the distributed algorithm is run for each time instance of a day. The shaded area highlights the range of iteration counts executed by the algorithm to converge over the day. In the fixed gradient method, the mean and the standard deviation of the number of iterations increases linearly with the number of homes with solar panels.

In contrast, the adaptive gradient takes smaller number of iterations — almost 3× to 30× fewer — to converge compared to the fixed gradient approach. Moreover, the adaptive gradient is more reliable, as the standard deviation of the iterations over the day is small. Further, compared to fixed gradient, the number of iterations doesn't grow linearly in the number of homes with solar panels. However, we notice that on an overcast day, the number of iterations required for both the fixed and the adaptive gradient is almost identical. Due to overcast conditions, the maximum solar generation rate is small, which results in faster convergence.

We also compare the performance of our distributed approach with the centralized approach. Note that the centralized approach has full knowledge about the routing topology and the maximum

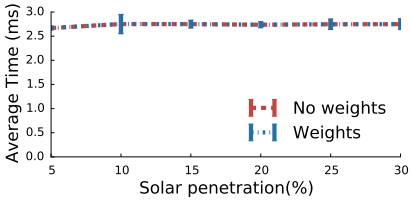


Figure 11: Time taken to compute solar rate using control prices in a Raspberry Pi 3.

power point (MPP) of each solar installation. As seen in Figure 10, the performance of both the centralized and the distributed approach is similar. On an average, we observe that the distributed solution converges to 98.3% of the centralized solution. Moreover, the maximum absolute difference between the distributed and centralized is 0.029 MW, while the average difference is 0.005 MW.
Result: *In comparison to fixed gradient, Adagrad requires 3× to 30× fewer iterations to converge. Moreover, our approach performs similar to the centralized approach i.e. within 98.3% on average.*

5.4.6 Distributed solar rate computation. We assume that a smart solar-powered arrays will have a Raspberry Pi class processor to receive control prices and control its solar rate at every iteration. Thus, we analyze the average time Raspberry Pi takes to complete a single iteration of the distributed algorithm on un-optimized python code. Note that the solar rate computed depends on the size of the control prices which varies based on the size of the distribution network (number of feeders and transformers). However, the number of feeders and transformers change infrequently for a given grid network (once in every few months or years). Thus, the time taken to compute the rate should theoretically remain the same. Figure 11 shows the empirical average time taken to execute the algorithm on Raspberry Pi 3. We observe that the execution time per iteration varies between 2.5 to 3 ms. If we assume the average communication time between the control center and the smart solar arrays to be 10 ms, with 20 iterations (AdaGrad) per convergence (for 5% solar penetration), the distributed algorithm should take less than 0.3 seconds to converge.
Result: *With 5% penetration, our distributed approach takes less than 0.3 sec to find the optimal rate allocation.*

6 RELATED WORK

A detailed assessment of distributed solar impact on the grid highlights the need for generation flexibility in managing solar variability [7]. The specific phenomenon of solar over-generation during the day causes large ramp up of power generation through peaking generators, which has been shown to pose operational challenges and put a tremendous amount of stress on the grid [1]. Prior work on controlling distributed solar generation include *demand side management* using storage or load matching [27] and *solar regulation* through curtailment or cutoff. Separately, other research work has focused on distributed generation control [18] and shown distributed and centralized voltage control have similar potential in increasing capacities of distributed generation [30].

Numerous studies on solar regulation through curtailment exist [5, 19, 26, 29]. Tonkoski et. al. presents an active power curtailment technique to increase the overall distributed solar capacity at the low-voltage feeder [29]. Rongali et. al. describes a voltage-based curtailment where the solar rate is reduced if the sensed voltage is higher than normal [26]. Lo et. al. presents a discrete curtailment approach by completely disconnecting the solar units through control signals from the utility's command center [19]. In contrast, we present a distributed algorithm that apportions the available solar capacity to individual smart solar arrays through a proportional fairness scheme.

In demand side management, user's demand and solar generation profile is either scheduled intelligently or shifted using energy storage —- to avoid the risk of excess solar supply. Zhao et. al. presents control algorithms for electric vehicle charging to mitigate the impact of renewable energy integration to the grid [33]. Palensky et. al. discusses different approaches to control demand side load [22]. Energy storage absorbs excess energy generated from solar and acts as a buffer for large variations in the output [6, 10]. However, energy storage costs are high and when energy storages are full, excess solar may still need to be curtailed. Our distributed approach is complementary to the energy storage and provides more control over distributed solar generation.

Distributed approach for rate control has been widely studied in the networking literature [16, 20]. However, these approaches are now being studied in the context of rate control of electric vehicles [4, 8, 25]. Carvalho et. al. discusses different fairness protocols to mitigate congestion in the grid caused by electric vehicles [8]. Our distributed formulation is similar to the approach proposed in [4]. However, unlike [4], which explores rate control for electric vehicles — we explore rate control in the context of distributed solar and explicitly model electricity distribution network constraints. Moreover, we explore different approaches for faster convergence of our distributed algorithm.

7 CONCLUSION

In this paper, we addressed the problem of growth in solar deployments that could cause supply-demand imbalance due to intermittency in power generation. We designed a decentralized rate control algorithm to allocate generation rate of individual smart solar arrays and apportion the aggregate grid solar capacity through a proportional fairness scheme. Our proposed decentralized algorithm made decisions local to a solar deployment to compute its solar rate without any need for explicit communication with the utility. We evaluated our rate control algorithm on a city-scale electric distribution network and showed that it generates a fair allocation. We observed that a dynamic rate control achieves significantly higher solar penetration with negligible energy curtailment compared to the current hard caps placed on solar deployments. We also presented convergence results that exhibit tractability of our algorithm. Further, we assessed the feasibility of our approach on a Raspberry Pi-class processor and showed that it executes in 0.3 seconds for a solar penetration level of 5%. In future, we plan to prototype the smart solar rate controller that implements our distributed algorithm.

Acknowledgements. We thank all the reviewers and our shepherd Zhenhua Liu for their feedback. This research is supported by NSF grants IIP-1534080, CNS-1405826, CNS-1253063, CNS-1505422, and the Massachusetts Department of Energy Resources.

REFERENCES

[1] 2016. What the Duck Curve Tells us about Managing a Green Grid. https://goo.gl/YvI2uc. (Feb 2016).

[2] 50states 2015. The 50 States of Solar: A Quarterly Look at America's Fast-Evolving Distributed Solar Policy Conversation. https://goo.gl/bXRy3p. (November 2015).

[3] O Ardakanian, S Keshav, and C Rosenberg. 2016. Integration of renewable generation and elastic loads into distribution grids. In *SpringerBriefs in electrical and computer engineering*.

[4] O. Ardakanian, C. Rosenberg, and S. Keshav. 2013. Distributed control of electric vehicle charging. In *Proceedings of the fourth international conference on Future energy systems*.

[5] S Bandyopadhyay, P Kumar, and V Arya. 2016. Planning Curtailment of Renewable Generation in Power Grids.. In *Proccedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling*.

[6] J. Barton and D. Infield. 2004. Energy storage and its use with intermittent renewable energy. *IEEE transactions on energy conversion* (2004).

[7] J. Bebic. 2008. *Power system planning: emerging practices suitable for evaluating the impact of high-penetration photovoltaics*. NREL.

[8] R. Carvalho, L. Buzna, R. Gibbens, and F. Kelly. 2015. Critical behaviour in charging of electric vehicles. *New Journal of Physics* 17, 9 (2015).

[9] C K Chau, G Zhang, and M Chen. 2016. Cost minimizing online algorithms for energy storage management with worst-case guarantee. *IEEE Transactions on Smart Grid* (2016).

[10] P. Denholm, E. Ela, B. Kirby, and M. Milligan. 2010. The role of energy storage with renewable electricity generation. (2010).

[11] J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011).

[12] DP Hohm and ME Ropp. 2000. Comparative study of maximum power point tracking algorithms using an experimental, programmable, maximum power point tracking test bed. In *Photovoltaic Specialists Conference, 2000. Conference Record of the Twenty-Eighth IEEE*.

[13] S. Iyengar, S. Lee, D. Irwin, and P. Shenoy. 2016. Analyzing Energy Usage on a City-scale using Utility Smart Meters. In *Proceedings of the ACM International Conference on Systems for Energy Efficient Build Environments*.

[14] V. Jacobson and M. Karels. 1998. Congestion Avoidance and Control. In *SIGCOMM*.

[15] Y Kanoria, A Montanari, S Tse, and B Zhang. 2011. Distributed storage for intermittent energy sources: Control design and performance limits. In *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE.

[16] F. Kelly, A. Maulloo, and D. Tan. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society* (1998).

[17] F. Kelly and E. Yudovina. 2014. *Stochastic networks*. Vol. 2. Cambridge University Press.

[18] Na Li, Changhong Zhao, and Lijun Chen. 2016. Connecting automatic generation control and economic dispatch from an optimization view. *IEEE Transactions on Control of Network Systems* (2016).

[19] C. Lo and N. Ansari. 2012. Alleviating solar energy congestion in the distribution grid via smart metering communications. *IEEE Transactions on Parallel and Distributed Systems* (2012).

[20] S. Low and D. Lapsley. 1999. Optimization flow control-I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking* 7, 6 (1999).

[21] A Mishra, R Sitaraman, D Irwin, T Zhu, P Shenoy, B Dalvi, and S Lee. 2015. Integrating energy storage in electricity distribution networks. In *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*.

[22] P. Palensky and D. Dietrich. 2011. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics* (2011).

[23] J Qin, Y Chow, J Yang, and R Rajagopal. 2014. Modeling and online control of generalized energy storage networks. In *Proceedings of the 5th international conference on Future energy systems*.

[24] J. Rifkin. 2015. The Rise of the Internet of Things and the Race to a Zero Marginal Cost Society. https://goo.gl/smcHB0. (November 10th 2015).

[25] E S Rigas, S D Ramchurn, N Bassiliades, and G Koutitas. 2013. Congestion management for urban EV charging systems. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE.

[26] S. Rongali, T. Ganuy, M. Padmanabhan, V. Arya, S. Kalyanaraman, and M. Petra. 2016. iPlug: Decentralised dispatch of distributed generation. In *COMSNETS*.

[27] P Samadi, V WS Wong, and R Schober. 2016. Load scheduling and power trading in systems with high penetration of renewable energy resources. (2016).

[28] A. Singh, S. Lee, D. Irwin, and P. Shenoy. 2017. SunShade: Software-defined Solar Power. In *Proceedings of the 8th ACM/IEEE International Conference on Cyber-Physical Systems*.

[29] R. Tonkoski, L. Lopes, and T. El-Fouly. 2011. Coordinated Active Power Curtailment of Grid Connected PV Inverters for Overvoltage Prevention. *IEEE Transactions on Sustainable Energy* (April 2011).

[30] P N Vovos, A E Kiprakis, A R Wallace, and G P Harrison. 2007. Centralized and distributed voltage control: Impact on distributed generation penetration. *IEEE Transactions on Power Systems* (2007).

[31] When 2016. When Will Rooftop Solar Be Cheaper Than the Grid? https://goo.gl/h1Ayy5. (March 31st 2016).

[32] Wind 2016. Wind and Solar Boost Cost-competitiveness versus Fossil Fuels. https://goo.gl/XlcZQP. (October 5th 2016).

[33] S. Zhao, X. Lin, and M. Chen. *Robust Online Algorithms for Peak-Minimizing EV Charging under Multi-Stage Uncertainty*. Technical Report. Available online: https://goo.gl/vtNHq0.