Distributed Multi-Period Optimal Power Flow for Demand Response in Microgrids

Paul Scott
The Australian National University
NICTA
paul.scott@anu.edu.au

Sylvie Thiébaux
The Australian National University
NICTA
sylvie.thiebaux@anu.edu.au

ABSTRACT

The scalability and privacy preserving nature of distributed optimisation techniques makes them ideal for coordinating many independently acting agents in a microgrid setting. However, their practical applicability remains an open question in this context, since AC power flows are inherently non-convex and households make discrete decisions about how to schedule their loads. In this paper, we show that one such method, the alternating direction method of multipliers (ADMM), can be adapted to remain practical in this challenging microgrid setting. We formulate and solve a multi-period optimal power flow (OPF) problem featuring independent households with shiftable loads, and study the results obtained with a range of power flow models and approaches to managing discrete decisions. Our experiments on a suburb-sized microgrid show that the AC power flows and a simple two-stage approach to handling discrete decisions do not appear to cause convergence issues, and provide near optimal results in a time that is practical for receding horizon control. This work brings distributed control for microgrids several steps closer to reality.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence

Keywords

OPF; ADMM; demand response; distributed control; microgrid; multi-agent systems; scheduling

1. INTRODUCTION

The distinction between generators and loads is fading as households adopt distributed generation, storage and smart devices. We envisage a future where network operators provide a competitive electricity market that anyone can participate in, and where this distinction between generators and loads is removed. This will be of particular importance for the operation of microgrids, which require more finesse to ensure that demand and supply are balanced and that the network is in a safe operating state in each instance.

A different approach is needed from the traditional centralised markets as they were never designed to operate where every customer is an active participant, or to handle their unique time-coupled behaviours. In this new regime demand response (DR) techniques will play a central role in providing incentives, coordination and network support.

The goal of the network operator is to serve power at the lowest cost. Several works [22, 16, 24] have adopted distributed solving techniques in order to solve this problem for many participants. These distributed algorithms greatly parallelise the problem and help to preserve the privacy of participants. As a by-product, they provide a natural market mechanism for fairly allocating payments between consumers and producers. Theoretically, these algorithms require the problem to be convex in order to guarantee convergence to a globally optimal solution. However, the behaviour of many loads within households are discrete in nature [28], and the equations that govern how power physically flows on the network are non-convex.

In this paper, we show that these theoretical problems can in practice be dealt with in the context of microgrids. We show that for a distributed DR algorithm in a microgrid, exact non-convex power flow models perform well compared to inexact convex models, which makes them a valuable candidate in practice. Secondly, we identify that the non-convex nature of discrete household loads is a non-issue, and that in practice simple approaches to handling these discrete loads are effective at the microgrid level. By solving these problems, we show that the use of distributed algorithms for managing the balance of power on a microgrid is in practice not only possible, but also highly effective.

We formulate the problem as a multi-period optimal power flow (OPF) problem to account for multiple time steps over a day, which can be used as part of a day-ahead pricing scheme or, as we propose, a receding horizon control algorithm. We solve the multi-period OPF problems in a distributed manner by adapting the alternating direction method of multipliers (ADMM) approach presented in [22]. We experiment with a range of power flow models of varying degrees of accuracy, to compare their relative behaviour in a distributed algorithm. We then introduce and compare several approaches layered on top of ADMM which manage the introduction of discrete variables into the problem. Technically, our contributions can be summarised as:

e-Energy'15, July 14–17, 2015, Bangalore, India.

ACM 978-1-4503-3609-3/15/07 ...$15.00.
DOI: http://dx.doi.org/10.1145/2768510.2768534.

- A comprehensive experimental comparison of the convergence of five commonly used power flow models when used for distributed OPF in a microgrid context.
- The identification that the exact non-convex power flow model in practice not only converges in this context, but also finds near-optimal solutions in a timely fashion relative to other models.
- The introduction and comparison of three simple but effective approaches to managing the discrete shiftable loads that are typically found within households.

Combined, these results show that distributed DR using ADMM can achieve near optimal solutions in a time frame that is practical for receding horizon control in this challenging microgrid setting, regardless of the theoretical limitations. This work brings distributed DR closer to the point where it can be deployed in a real microgrid.

In the next section we discuss the related work and how our contribution is unique. In Sections 3–4 we formulate the problem and present the distributed algorithm we use to solve it. The test microgrid is introduced in Section 6 before presenting our results in Sections 7–8 on power flows and discrete decisions.

2. RELATED WORK

Much of the existing work on demand response (DR) has focused on using real-time pricing (RTP) as a control signal [28, 25, 7, 30, 32, 15]. In these methods, participants receive a RTP signal and individually optimise their own behaviour, so as to minimise a combination of monetary and discomfort costs. Other approaches have utilised non-pricing control signals, which are simpler to implement, but are limited in the types of loads that they can model [33, 31].

These approaches implement a form of open loop control, because the agent that sets the control signal (RTP or otherwise) at best can only estimate how consumers will respond to it. In order to reduce the amount of guesswork and improve solutions, a closed loop approach to RTP was presented by Gatsis et al. [16]. In this scheme, the prices are iteratively updated by a central agent, with consumers communicating their best responses to the price prior to acting. Mohsenian-Rad et al. [24] introduce an alternative iterative procedure not based on RTP, where consumers cooperate to reduce total generation costs in a distributed manner.

The approaches discussed so far do not model the electricity network, so cannot account for real power losses, reactive power, voltage limits or line thermal limits. Without these considerations, we cannot be sure that the DR outcome is efficient, safe or even possible. Many of the works on distributed algorithms which explicitly model the network have used ADMM as a solving technique, due to its ease in decomposition, and its convergence guarantees on a wide range of problems [6]. However, most of these works have focused on more traditional OPF problems rather than demand response in a microgrid context.

One of the first authors to apply ADMM to power networks was Kim et al. [21], who decomposed a convex approximation of the OPF problem into regions, and compared the results to two other approaches. They found it to have a significant speed improvement over a centralised approach, and that it preserved privacy between regions. Erseghe [12] also performed region-based decomposition of the network and found exact local solutions to the OPF problem. Instead of decomposing on the network structure, Phan et al. [27] decomposed across scenarios in a security-constrained OPF. The recent work by Magnússon et al. [23] decomposes the network to a greater extent than these other methods, and they solve the underlying non-convex OPF by taking sequential convex approximations. One thing all these works have in common is that they are focused on the more traditional OPF problem, whereas in our work we consider a microgrid where distributed participants act independently.

Region-based decomposition was also used by Dall'Anese et al. [9] to control distributed generation on radial feeders. They used ADMM to solve an unbalanced OPF problem using a semidefinite programming (SDP) relaxation. In our work we consider each customer to be independent, for privacy reasons, and we also allow for meshed microgrid topologies. Šulc et al. [34] use the relaxed DF (SOCP) equations to perform reactive power control on radial networks. For a similar problem, Peng et al. [26] provide closed-form solutions for ADMM subproblems, greatly reducing the computational requirements. Again these works focus on radial networks.

The work that is closest to ours is that presented by Kraning et al. [22], and indeed we build on their approach. They decompose all network components for a multi-period OPF problem using a quadratic power flow approximation. This procedure is effectively a principled method for settling RTPs for each bus, also known as locational marginal prices. Their experiments showed that very large problems could be solved efficiently in a parallel environment.

All these works have taken different approaches to modelling the power flows on the network. There is no comparison of the relative performance between these different power flow models in a distributed algorithm for microgrids, which is what we achieve in this paper for five different models. Our results in this area indicate that exact local methods can produce close to optimal solutions in a competitive number of iterations relative to other models. In addition, to the best of our knowledge, we are the first to incorporate discrete decisions into a distributed demand response mechanism that models the network. Our work brings ADMM to the point where it can be considered a practical approach for efficiently balancing power in a microgrid setting.

3. PROBLEM FORMULATION

The overall objective of the demand response problem is to minimise the average long-term cost of supplying electricity. We formulate this as a series of multi-period OPF problems embedded within a receding horizon control process, which enables time-coupled components to be accurately controlled in an uncertain environment. A multi-period OPF is first solved over a horizon of $n \in \mathbb{N}$ time steps, the decision in the first step is acted on, and then the process repeats with the window shifted forward by one. In this paper we focus on solving the multi-period OPF within a single horizon, and the actions that agents take to implement the first decision.

Note that the formulation that we use breaks away from the standard in power systems in order to decompose the network and distribute the problem. Fig. 1 highlights the difference between a standard line diagram and our formulation. In our model, a network N consists of a set of components C, terminals T and connections L. Each component

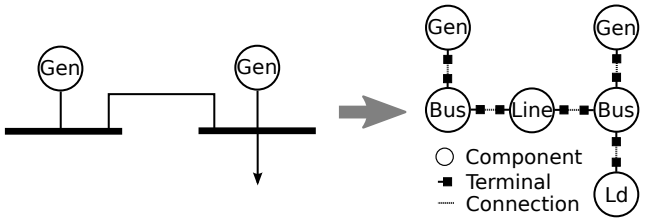

Figure 1: Conversion from a standard line diagram to the component orientated representation.

$c \in C$ (e.g., bus, line, generator, load) has a set of terminals $T_c \subseteq T$ which can be connected to the terminals of other components, where the T_c sets partition T. Each connection $l \in L$ is a pair of terminals, i.e. $L \subseteq T \times T$.

3.1 Connections

Connections exist between the terminals of two different components. We use the quantities of real power, reactive power, voltage and voltage phase angle ($p, q, v, \theta \in \mathbb{R}^n$ respectively) to model the flow of power into a component through a terminal. These are vectors in order to capture each time step in the horizon. For convenience, we use a parent vector $y_i \in \mathbb{R}^{4n}$ to represent all variables for a terminal $i \in T$, where $y_i := (p_i, q_i, v_i, \theta_i)^\mathsf{T}$. When two terminals are connected together, $(i, j) \in L$, we pose the following constraints:

$$p_i + p_j = 0, \quad q_i + q_j = 0, \quad v_i - v_j = 0, \quad \theta_i - \theta_j = 0$$

The first two constraints ensure that for a connected pair of terminals, at each time step, any power that leaves one terminal must enter the other. The second two constraints ensure that the connected terminals have the same voltage and phase angle. This duplication of variables is necessary in order to decompose the problem for our distributed algorithm. To avoid confusion, recall that connections and terminals are different from lines and buses (see Fig. 1).

We rewrite these constraints as $y_i + Ay_j = 0$ for y, where A is the appropriate $4n \times 4n$ diagonal matrix. Further, we define the connection function $h : \mathbb{R}^{4n} \times \mathbb{R}^{4n} \mapsto \mathbb{R}^{4n}$ as the LHS of this constraint for convenience: $h(y_i, y_j) := y_i + Ay_j$.

3.2 Components

At a high level, each component $c \in C$ has a variable vector $x_c \in \mathbb{R}^{a_c}$, an objective function $f_c : \mathbb{R}^{a_c} \mapsto \mathbb{R}$, and a constraint function $g_c : \mathbb{R}^{a_c} \mapsto \mathbb{R}^{b_c}$, where $g_c(x_c) \leq 0$. For a component $c \in C$, the vector x_c includes all terminal variables for that component: $y_i, \forall i \in T_c$.

The objective function is used to model any costs or preferences that a component may have other than the direct payments they make to the market for their consumption. For a generator this can be the fuel costs, for a house this might be temperature comfort preferences, and other components like a line will not have any costs.

In the following sections we describe at a lower level the models used for the components in our experiments. When necessary, we use $t \in \{0, \ldots, n\}$ to index vectors by time, otherwise we imply standard vector operations. The index where $t = 0$ is used to represent the value of the variable at the beginning of the current horizon, which we assume is known.

3.2.1 Bus

A bus has a variable number of terminals which depends on how many other components connect to it. For example, a bus might be connected to a generator, a load and 3 lines for a total of 5 terminals. Regardless of the number of terminals, the constraints take the form:

$$\sum_{i \in T_c} p_i = 0 \quad \sum_{i \in T_c} q_i = 0$$
$$\forall i, j \in T_c : v_i = v_j, \ \theta_i = \theta_j$$

The first two constraints are an expression of Kirchhoff's current law (KCL) in terms of power flows. The remaining constraints ensure that all terminal voltages and phase angles are the same.

3.2.2 Line

A line is a two terminal component which transports power from one location to another, typically from bus to bus. We model a line as having a constant conductance $g \in \mathbb{R}_+$, susceptance $b \in \mathbb{R}$ and maximum apparent power $s \in \mathbb{R}_+$. The AC power flow equations are derived from Ohm's law, where $\forall i, j \in T_c, i \neq j$:

$$p_i = gv_i^2 - gv_iv_j\cos(\theta_i - \theta_j) - bv_iv_j\sin(\theta_i - \theta_j) \quad (1)$$
$$q_i = -bv_i^2 + bv_iv_j\cos(\theta_i - \theta_j) - gv_iv_j\sin(\theta_i - \theta_j) \quad (2)$$
$$s^2 \geq p_i^2 + q_i^2, \quad \underline{v} \leq v_i \leq \bar{v}, \quad \theta_i - \theta_j \leq \bar{\theta} \quad (3)$$

These constraints are identical for each time step, so we have left out the indexing by time to improve clarity. These equations are non-convex, so they are often either approximated or relaxed, as we will discuss further in Section 7.

3.2.3 Generator

A generator is a single terminal component which produces real and reactive power. In our formulation the generator has a floating phase angle and voltage. A generator has lower and upper real and reactive power limits such that $p_{i,t} \in [\underline{p}, \bar{p}]$ and $q_{i,t} \in [\underline{q}, \bar{q}]$, a ramping rate $p^r \in \mathbb{R}_+$ and a quadratic cost function f for generation costs:

$$f(x) = p_i^\mathsf{T}\Psi p_i - \psi^\mathsf{T}p_i$$
$$\forall t \in \{1, \ldots, n\} : -p^r \leq p_{i,t} - p_{i,t-1} \leq p^r$$

where $\Psi \in \mathbb{R}_+^{n \times n}$ is a diagonal matrix and $\psi \in \mathbb{R}_+^n$. More advanced generator models with non-convex start up costs and minimum outputs can be modelled in this framework but are not considered here. They will be investigated in future work to see how they impact the distributed algorithm.

3.2.4 Shiftable Load

A shiftable load is a single terminal component used to model electrical loads like dish washers and clothes dryers. A household has some flexibility on when these loads can run, and will schedule them to minimise the costs they pay for the electricity. These loads must start running between an earliest and a latest start time: $t^e, t^l \in \mathbb{N}$. To model this we introduce binary variables $u \in \{0, 1\}^n$ for the horizon. A value of 1 indicates that the component starts at the given time. A component runs for a duration of $d \in \mathbb{N}$ consecutive time steps, during which it consumes a load of $p^{\text{nom}} \in \mathbb{R}$.

$$p_{i,t} = p^{\text{nom}} \sum_{t'=t-d+1}^{t} u_{t'}, \qquad \sum_{t=t^e}^{t^l} u_t = 1$$

$$\forall t \notin \{t^e, \ldots, t^l\} : u_t = 0$$

A convex relaxation of this component can be obtained by relaxing the integrality requirement: $u \in [0,1]^n$. Shiftable loads with more complex time-varying power consumptions can be modelled as in [30]. We expect the results presented here will carry over to this more complicated model, but leave a thorough check to future work.

3.2.5 Other Components

A whole range of other components can easily be modelled within this framework, for example, batteries, inverters, solar PV, electric vehicles, HVACs and voltage regulators (see [30] for additional models). Indeed we have experimented with batteries and solar PV in our implementated algorithm, but in this paper we focus on the more difficult to handle shiftable loads.

3.3 Optimisation Problem

Now that we have the component models and the relations between them, we can write down the multi-period OPF problem for one horizon. The objective is to minimise the sum of all component cost functions, subject to component and terminal connection constraints. This is a utilitarian view of the problem.

$$\min_x \sum_{c \in C} f_c(x_c) \tag{4}$$

$$\text{s.t. } \forall c \in C : g_c(x_c) \leq 0 \tag{5}$$

$$\forall (i,j) \in L : h(y_i, y_j) = 0 \tag{6}$$

4. DISTRIBUTED ALGORITHM

The next step is to show how this multi-period optimisation problem can be solved in a distributed manner. The end result is an iterative algorithm where each component (household, generator and network device) selfishly optimises its own consumption/production profile for the currently standing prices. These profiles are then communicated amongst connected components and the prices are modified in order to encourage agreement and consistency.

In order to distribute and solve the problem in this way we use the alternating direction method of multipliers (ADMM) algorithm. ADMM is a variation of the standard augmented Lagrangian method that enables problem decomposition [6, 11, 14]. The augmented Lagrangian relaxation applied to the connection constraints (6) is:

$$\mathcal{L}(L, y, z, \lambda, \rho) := \sum_{c \in C} f_c(x_c) + \sum_{(i,j) \in L} \left[\frac{\rho}{2} \| h(y_i, z_j) \|_2^2 + \lambda_{i,j}^\mathsf{T} h(y_i, z_j) \right]$$

where $\rho \in (0, \infty)$ is a penalty parameter and $\lambda_{i,j} \in \mathbb{R}^{4n}$ are the dual variables for the connection constraints.

These dual variables represent the locational marginal prices in our problem, or put another way, connection dependent RTPs. These prices are used to charge (or pay) components for the power that they exchange through their terminals. For example, a component with a terminal i (connected to terminal j) will pay, or be paid an amount equal to:

$$\lambda_{i,j,p}^\mathsf{T} p_i + \lambda_{i,j,q}^\mathsf{T} q_i + \lambda_{i,j,v}^\mathsf{T} v_i + \lambda_{i,j,\theta}^\mathsf{T} \theta_i \tag{7}$$

Where we have split up the dual variables so that it is clear how they associate with each physical power quantity. These prices are based on not just the cost of generation, but also account for line losses and adjust to prevent congestion. They provide a natural market mechanism for the fair distribution of payments from consumers to producers.

4.1 Algorithm

A single iteration of the ADMM algorithm consists of two phases followed by a dual variable update. Components are each allocated to one of the two phases. The component sets C_1 and C_2, and the variable vectors x_1 and x_2 represent this allocation.

The connections are split into three parts: L_1, L_2 and $L_{1,2}$. The intra-phase connections L_1 (L_2) are those that are between components in C_1 (C_2). The inter-phase connections $L_{1,2}$ are those where one component is in C_1 and the other is in C_2. The augmented Lagrangian relaxation is only applied to the inter-phase connections.

The superscript $k \in \mathbb{N}$ is used to indicate the k-th iteration. At the start of the algorithm all terminal and dual variables are initialised to some values $y_i^{(0)}$ and $\lambda_{i,j}^{(0)}$. For the k-th iteration ADMM proceeds as follows:

1. Optimise for x_1, holding x_2 constant at its $k-1$ value
2. Optimise for x_2, holding x_1 constant at its k value
3. Update the dual variables λ

For our optimisation problem this becomes:

$$x_1^{(k)} = \arg\min_{x_1} \mathcal{L}(L_{1,2}, y, y^{(k-1)}, \lambda^{(k-1)}, \rho^k) \tag{8}$$

$$\text{s.t. } \forall c \in C_1 : g_c(x_c) \leq 0$$

$$\forall (i,j) \in L_1 : h(y_i, y_j) = 0$$

$$x_2^{(k)} = \arg\min_{x_2} \mathcal{L}(L_{1,2}, y^{(k)}, y, \lambda^{(k-1)}, \rho^k) \tag{9}$$

$$\text{s.t. } \forall c \in C_2 : g_c(x_c) \leq 0$$

$$\forall (i,j) \in L_2 : h(y_i, y_j) = 0$$

$$\forall (i,j) \in L_{1,2} : \lambda_{i,j}^{(k)} = \lambda_{i,j}^{(k-1)} + \rho^{(k)} h(y_i^{(k)}, y_j^{(k)}) \tag{10}$$

In the simple case when ρ is constant, f_c and g_c are convex, and h is affine, ADMM converges to a global optimum [6].

If a component has no intra-phase connections, then it can be separated from the optimisation problem for its phase, and can therefore be solved independently. We adopt the partitioning scheme where C_2 contains all buses and C_1 the rest of the network. This allows us to fully separate all components within phases, since buses will never connect to other buses ($L_2 = \emptyset$) and non-bus components will never connect to other non-bus components ($L_1 = \emptyset$). In this way each component acts as an independent agent and communicates only to other directly connected agents. As an additional benefit, some components are simple enough when separated that they have closed-form solutions that can be calculated at each iteration, instead of invoking an optimisation routine [26]. We adopt such closed-form solutions for buses as proposed in [22].

4.2 Residuals and Stopping Criteria

As in [22], we use primal and dual residuals to define the stopping criteria for our algorithm. The primal residuals represent the constraint violations at the current solution.

We combine the residuals of all connections into a single vector r_p. By indexing into the inter-phase connections $L_{1,2} = \{(i_1, j_1), (i_2, j_2), \ldots\}$, the primal residuals are:

$$r_p^{(k)} := (h(y_{i_1}^{(k)}, y_{j_1}^{(k)}), h(y_{i_2}^{(k)}, y_{j_2}^{(k)}), \ldots)^\mathsf{T}$$

The dual residuals give the violation of the KKT stationarity constraint at the current solution. We collect the dual residuals for each connection into the vector r_d. For ADMM, the dual residuals are (see [6] for derivation):

$$r_d^{(k)} := \rho(Ay_{j_1}^{(k)} - Ay_{j_1}^{(k-1)}, Ay_{j_2}^{(k)} - Ay_{j_2}^{(k-1)}, \ldots)^\mathsf{T}$$

These residuals approach zero as the algorithm converges to a KKT point. We consider that the algorithm has converged when the scaled 2-norms of these residuals are smaller than a tolerance ϵ: $\frac{1}{\sqrt{M}}\|r_p^{(k)}\|_2 < \epsilon$, $\frac{1}{\sqrt{M}}\|r_d^{(k)}\|_2 < \epsilon$. Here M is the total number of inter-phase terminal constraints $4n|L_{1,2}|$ minus the number of terminal constraints that are trivially satisfied (e.g., floating voltages and phase angles for generators). It is used to keep the tolerance independent of problem size.

5. IMPLEMENTATION

We developed an experimental implementation of the above approach in C++ using Gurobi [17] and Ipopt [35, 19] as backend solvers for subproblems. Gurobi is used for mixed-integer linear or quadratically constrained problems, and Ipopt for more general nonlinear problems. CasADi [1] was used as a modelling and automatic differentiation front end to Ipopt. This implementation was designed with flexibility in mind, so that a wide range of experiments could be conducted.

In a fully distributed real-world implementation every house, generator, bus, line, and other component could have its own collocated computational node. However, from a practical point of view it might make more sense to have the computational parts of the network located separately from their components and even grouped together. For example, all the buses and lines of a single feeder could be managed by a single node, which communicates to downstream houses and the upstream substation. A whole range of practical factors such as speed, communications, costs, robustness and maintenance would need to be considered before a decision could be made on the right architecture.

Our experimental setup is a sequential implementation of the ADMM algorithm, however we timed the slowest component at each iteration to get an idea of how long a fully distributed implementation would take.

The experiments were run on machines with 2 AMD 6-Core Opteron 4184, 2.8GHz, 3M L2/6M L3 Cache CPUs and 64GB of memory.

6. TEST MICROGRID

Our experiments are based around a modified 70 bus 11kV benchmark distribution network [10] (shown in Fig. 2), which was chosen because it has a comparable size to that of a suburb. We close all tie lines in the network in order to change it from a radial to a meshed configuration. We expect microgrids to take on more of a meshed network structure to improve reliability and efficiency, and to better utilise distributed generation.

The benchmark comes with a static PQ load at each bus, which we replace with a number of houses (around 50 on average) that depends on the size of the load. The houses are connected directly to the 11kV buses as we have no data on the low voltage part of the network. We assume that the power bounds we place on each household will be sufficient to prevent any capacity violation of the low voltage network.

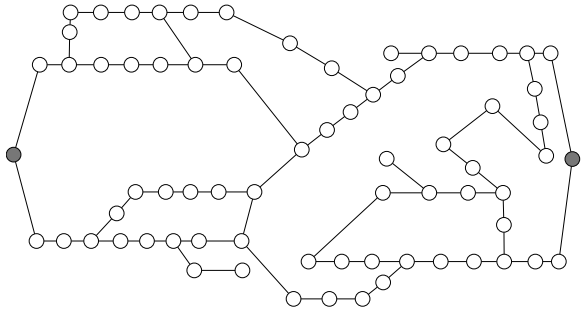

Figure 2: 70 bus network showing buses, lines and the generators/substations in grey.

A house is an independent agent that manages subcomponents. For our experiments these include an uncontrollable background power draw and two shiftable loads. A house has a single terminal through which it can exchange real and reactive power with the rest of the network. Each house has an apparent power limit of $s = 10$kVA.

We develop a typical house load profile l_t by modifying an aggregate Autumn load profile for the ACT region in Australia (data from [2]). We assume that households consume on average 20kWh per day. This provides the basis for all uncontrollable household background loads. For the purposes of these experiments, we assume that the static PQ loads in the benchmark were recorded when loads were at 75% of their peak. We divide the benchmark static real power at each bus by how much power a typical house consumes at 75% of its peak power (1.45kW). Rounding down this number gives us an estimate of the number of houses which would be located at a given bus. This approach produces a total of $h = 3674$ houses for the network, about the size of an Australian suburb.

We place two generators in the network where the distribution system connects to upstream substations. These can be thought of as either dispatchable microgenerators or as representing the cost of importing power into the microgrid.

We randomise some of the generator and household load parameters to produce different problem instances, as can be seen in Table 1. The time horizon spans 24 hours with 15 minute time steps, which produces a problem instance with over 2 million variables per horizon. The experiments were run with a primal and dual stopping tolerance of $\epsilon = 10^{-4}$ and a fixed penalty parameter of $\rho = 0.5$. To improve numerical stability, we scale the system to a per-unit representation with base values at 11kV and 100kVA. This means that a real power residual of 10^{-4} translates to 10W for a connection, or about 1% of the average household load.

The starting values for the distributed algorithm are the same for all terminals and all time steps. All are zero except for the voltage magnitudes $v_t = 1$ and the real power constraint dual variables $\lambda_t^p = 5$, which translates to a price of 200 \$/MWh. This is a naive (or cold) starting point as it uses no information about the particular network instance.

Table 1: Component parameters.

Comp	Param	Value	Units
Gen	ψ_t	$\max(4, \sim \mathcal{N}(40, 8^2))$	\$/MWh
Gen	$\Psi_{t,t}$	$\max(1, \sim \mathcal{N}(10, 2^2))$	\$/MWhMW
Gen	$\underline{p}, \bar{p}$	$-s \times h/2, 0$	kW
Gen	$\underline{q}, \bar{q}$	$0.2\underline{p}, -0.2\underline{p}$	kvar
House	p_t	$\sim \mathcal{N}(l_t, (0.2l_t)^2)$	kW
House	q_t	$0.3p_t$	kvar
Shift 1	d	$\max(15, \sim \mathcal{N}(90, 18^2))$	min
Shift 1	p^{nom}	$\max(0.3, \sim \mathcal{N}(3, 0.6^2))$	kW
Shift 2	d	$\max(15, \sim \mathcal{N}(60, 12^2))$	min
Shift 2	p^{nom}	$\max(0.1, \sim \mathcal{N}(1, 0.2^2))$	kW
Shift	t^e, t^l	$0, n-d$	

In addition to the 70 bus microgrid, we also ran a series of experiments on randomly generated networks similar to those described in [22]. These randomly generated networks ranged in size from 20 to 2000 buses, and were designed to be highly congested. We will occasionally mention some of the results from these random networks when they differ from those of our 70 bus microgrid.

7. IMPACT OF POWER FLOW MODELS

In this section we investigate how the ADMM method performs with different power flow models. We assess 5 different models, of varying degrees of accuracy and complexity, in order to establish the relative trade-offs when used as part of a distributed algorithm.

7.1 Power Flow Models

Due to their non-convex nature, the AC power flow equations (1–3) are often either relaxed or approximated. Convex relaxations include a quadratic constraint (QC) model [18], a semi-definite program (SDP) [3], the dist-flow (DF) relaxation [13, 4] and an equivalent SOCP relaxation [20, 5]. Approximations include the linear DC (DC) model that uses p and θ [29], the LPAC model [8] and the quadratic formulation (K) proposed by Kraning et al. [22].

The relaxations provide a lower bound on the globally optimal solution while the approximations can produce results with an objective higher or lower than the global optimal. Both the relaxations and approximations often produce solutions that are not feasible for the exact model. These alternative models, however, are often much simpler to compute and their solutions can be used as a heuristic or the bounds can be used for calculating optimality gaps. This is why they are often used with difficult network optimisation problems, for example, OPF, OPF with line switching, capacitor placement and expansion planing.

As shown in the related work section, some of these models have been used with the ADMM algorithm. What is lacking is a comparison of the relative strengths and weaknesses between the different models when used in this context. In this section we compare how the distributed ADMM algorithm performs when using the AC, QC, DF, DC, and K line models. We compare the differences in the solution quality, feasibility, processing time and number of iterations for our test network. What we find is that even though the AC equations are non-convex, in practice they converge and perform well compared to the other approaches. We also find that there is the potential to obtain faster convergence using the K model, but at the expense of accuracy.

Table 2: Iterations and parallel solve time for line models.

	Iterations (std.)	Time in sec (std.)
AC	1945 (17)	148 (12)
QC	1951 (14)	546 (33)
DF	1933 (26)	110 (8)
DC	4140 (50)	244 (8)
K	1027 (52)	15 (1)

We generate 60 random instances of our test microgrid with the binary variables for the shiftable devices relaxed. These are then solved using the distributed algorithm described in Section 4, for each of the 5 different power flow models. In the first part of this section we discuss the convergence, and in the second part we discuss the quality and accuracy of the solutions.

7.2 Convergence

For all 60 instances and all 5 power flow models the algorithm converged. This was expected for all the convex models, but we had no guarantee for the non-convex AC model. This gives us confidence that the exact AC model, even though non-convex, can in practice be used within distributed algorithms.

Table 2 provides the number of iterations and time taken to converge in the form of means and standard deviations. The parallel solve time is the amount of time required to solve the problem in a fully distributed implementation. This was measured by summing together the time of the slowest component at each iteration. In absolute terms, the parallel solve time is relatively small despite the fact that our implementation was designed with flexibility in mind, not performance. That said the K model is significantly faster relative to the other models. It converges in half the number of iterations required by the next fastest model, but as we will see in the next section, it gives us an inaccurate result.

The congested random networks produce similar results. One difference is that for a number of instances the K model was infeasible (would not converge) due to its tendency to exaggerate line losses, where we had a valid AC solution. It is expected that the DC model, and the relaxations to some degree, will exhibit the reverse effect: returning a solution when there is no feasible solution for the exact model. However, we did not come across such a scenario in our experiments with the microgrid and congested random networks.

Fig. 3 shows an example of the primal residual convergence for different line models (the dual residuals are similar). The AC, QC and DF models overlap. One unintuitive result is the fact that the DC model converges poorly when it is in fact a very simple linear model. Large oscillations build up during the solution process which slows the rate of convergence. We performed a series of experiments in order to get a better understanding of this effect. The explanation appears to be that the DC model behaves like an undamped system, as it has no line losses and only linear constraints. The DC model will have a stronger response for a given change in its terminal dual variables. The net effect is that oscillations build up across the network during the solving process. On the other hand the K model overestimates line

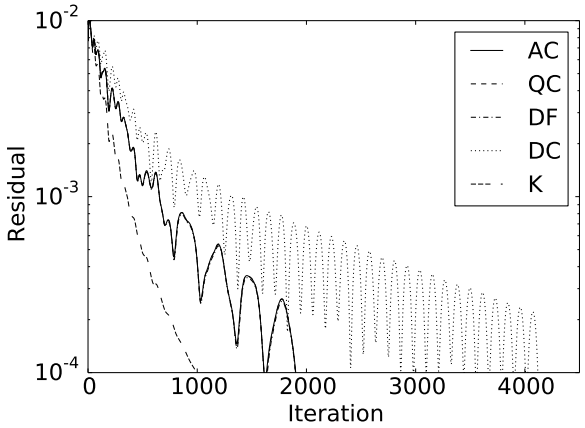

Figure 3: Convergence of primal residuals.

losses, which means it is much less sensitive and no oscillations form. The AC, QC and DF models are somewhere in between these two extremes.

7.2.1 Warm Starting

It is important to point out that we are giving the algorithm a naive starting point for both the primal and dual variables, as described in Section 6. In practice, the receding horizon control scheme will provide an excellent warm starting point, because the values from the previous horizon can be used for all but one time step. As a sanity check, we performed warm starting experiments for the AC model. Similar to what was done in [22], we duplicate a problem instance and then randomly resample the household background power and shiftable device power parameters according to the rule: $p \sim p\mathcal{N}(1, \sigma^2)$. We used the solution of the original instance as a starting point for the modified instance. For $\sigma = 0.2$ the warm started run only needed 11% of the original iterations on average. In a second experiment we fully correlated the resampling step, which could represent a correlated change in solar panel output for many households. With $\sigma = 0.2$, only 29% of the original iterations were required on average.

7.2.2 Communication

In reality, communication delay will play a major part in the total solve time for the algorithm. The communications could be done over existing internet infrastructure, or dedicated wired or wireless communications could be built for the system to enable more direct communications. Regardless of what technology is used, for each iteration messages need to be communicated from the first phase components to the second phase components and then back again. If we assume each of these hops takes 60ms, then 1000 iterations would require up to 2 minutes of communication time. For this reason, in certain circumstances it may be beneficial to cut down the total number of iterations, even if it requires more processing time per iteration.

We expect the mechanism can be designed to be quite robust to intermittent drops in communication. For example, if a component fails to receive a message from another connected component, then they can continue working by using the last received message. If a connection is dropped for an extended period, then the system could fall back to load predictions based on historical data and some conservative pricing scheme could come into place.

7.3 Solution Quality

Next we show the solution qualities for the different line models. For each model we calculate the percentage difference in objective value relative to the best known AC solution: $100 \cdot (f - f_{best})/f_{best}$. The means and standard deviations of the 60 instances are:

QC	-0.031% (0.008%)	DF	0.039% (0.018%)
DC	-3.541% (0.072%)	K	4.726% (0.090%)

Because the AC equations are non-convex, we don't have a guarantee that the solutions they produce are globally optimal. However, they provide a feasible upper bound on the global optimal. On the other hand, the QC and DF models are convex relaxations of the AC equations, so they provide a lower bound on the global optimal. Therefore the global optimal solution resides somewhere between the values of the AC solution and the QC and DF solutions.

With this in mind, we find that the AC, QC and DF models all produce solutions which are very close to each other. The difference is within the margin of error of the objective function afforded by our stopping criteria, which we estimate to be 1% (see Section 6). This indicates that the AC, QC and DF models produce solutions that are within 1% of the global optimal. They may in fact be closer than this, but we would need to run the experiments with tighter tolerances in order to check. On a limited number of instances we did just this, and found the gap between the objective of the AC model and its convex relaxations to further shrink into insignificance.

These results give us confidence that the non-convex AC model, which is the only one that guarantees Ohm's law is satisfied, produces solutions that are very close to optimal. The QC and DF models produce results with an objective that is very close to the AC model, but even with this small difference, there is the risk that the solutions violate constraints in the exact AC model. Other work has come to a similar conclusion, but in a more traditional OPF setting [18, 27, 12].

There is quite a different story for the approximate models. The DC model underestimates the optimal value by around 3.5% while the K model overestimates it by around 4.7%. Part of the reason for this is that the DC model completely ignores line losses while the K model overestimates them. Even though the K model has fast convergence, it is unlikely to useful on its own in a realistic setting due to its poor accuracy. However, it might be useful in hybrid approaches where line models are swapped, e.g., from K to AC, part way through the solution process in order to speed up convergence.

These results show the feasibility of using the non-convex AC power flow equations for solving a distributed OPF problem in a microgrid context. The K model adopted in [22] converges much faster, but it is unlikely to be usable in a realistic setting, as it ignores voltages and reactive power, and produces overly high costs.

8. DISCRETE DECISIONS

We now want to solve the multi-period OPF for the test microgrid where the binary variables in the shiftable loads are no longer relaxed. In order to do this we extend the

algorithm so that it can manage discrete decisions. The focus here is on the scheduling of shiftable loads within households, but discrete decisions can also occur in some generator models and for network switching events.

We identify 3 different approaches to managing discrete decisions. Although they are quite simple, they are nonetheless very effective at managing the shiftable loads within households.

8.1 Methods

We investigate 3 tractable methods for dealing with integer variables which have no global optimality guarantees. Just as we did for the AC equations, we will compare our result to a lower bound in order to get an understanding of the optimality gap. We categorise these methods as:

- Relax and price (RP)
- Relax and decide (RD)
- Unrelaxed (UR)

The RP and RD approaches are broken up into 2 and 3 stages respectively. The first stage, called the negotiation stage, is common to both methods. All integer variables are relaxed and the distributed algorithm is run until convergence, just like what was done for the power flow experiments. At this point the integer variables may take on fractional values, and this solution gives a lower bound on the global optimal solution. In the second stage each component makes a local decision in order to force any fractional values to integers. Recall from Section 3.2.4 that shiftable devices have a binary variable u_t for each time step, only one of which can take on the value 1 to indicate the starting time.

8.1.1 Relax and Price

In the second stage of the RP method, each house performs a local optimisation to determine how to enforce integer feasibility of u_t. We designed a range of cost functions which penalise a component if it changes its terminal values from those that were negotiated in the first stage. For a given cost function each house solves a Mixed-Integer Program (MIP) to obtain an integer-feasible solution. The two most effective cost functions that we identified are:

$$f_0(y, \hat{y}, \hat{\lambda}) = \hat{\lambda}^\mathsf{T} y + \alpha h(y, \hat{y})^\mathsf{T} h(y, \hat{y}) \tag{11}$$

$$f_3(y, \hat{y}, \hat{\lambda}) = \hat{\lambda}^\mathsf{T} A\hat{y} + \alpha h(y, \hat{y})^\mathsf{T} \Lambda h(y, \hat{y}) \tag{12}$$

where, for a given house to bus connection, $\hat{y}$ is the negotiated terminal values for the bus and $\hat{\lambda}$ the negotiated dual variables. We use Λ to represent the diagonal matrix where $\Lambda_{i,i} := |\hat{\lambda}_i|$ and α is a penalty parameter.

The first function charges households at the negotiated price for what they *actually* consume, but they are also charged a quadratic penalty for operating away from the negotiated consumption. The second function requires the household to pay for all power that was negotiated in the first stage. As with the first function, a penalty is charged for operating away from the negotiated operating point, however the penalty is scaled by the dual variables, which can vary with time.

After this local optimisation step, we check that the solution is feasible and what the overall cost is. In order to do this we need to put some degrees of freedom back into the problem. In power networks the dispatch of generators are established in advance, in response to an estimated demand. This forecast is never perfect, so a certain number of generators are paid to perform frequency regulation in order to balance demand in real time. In our experiments we employ both our generators for this use by allowing them to adjust their output. For these experiments we assume the same cost function and prices for both dispatch and frequency regulation.

8.1.2 Relax and Decide

In the second stage of the RD method, the largest u_t value of each shiftable component is chosen to be fixed at 1 and the rest set to 0. In the third stage the distributed algorithm is restarted in order to converge to a new solution that is integer feasible.

8.1.3 Unrelaxed

The final approach, UR, consists of a single stage where it attempts to enforce integrality satisfaction at each iteration of the distributed algorithm. We have already foregone theoretical convergence guarantees by our adoption of the non-convex AC equations. Here we push the ADMM algorithm even further by allowing discrete variables into the algorithm (8–10), where Gurobi solves MIPs for houses, and Ipopt NLPs for lines.

We ran experiments on 60 random instances of our test microgrid for each of the three approaches. We use the AC line model for each experiment and a penalty of $\alpha = 10$ for the RP approach. In the following sections we discuss the convergence of the methods and the quality of the solutions.

8.2 Convergence

None of the approaches are guaranteed to find an integer feasible solution if one exists, however, in practice they all converged to feasible solutions for all experiments on our test microgrid.

The RP method only marginally increases the solve time above the results in Section 7. The RD method requires a small amount of extra time as it performs a warm restart of the distributed algorithm. The UR method takes 1.7 times longer on average, which is a result of the fact that it solves MIPs during each iteration.

8.3 Solution Quality

In order to assess the solution quality for each method, we compare the change in objective value relative to the relaxed version of the problem. The results are shown in Fig. 4, where we have separated the objective into terms for the cost of generation and the charge to households. The charge is the sum of household objective functions, which represents the amount of money they pay for their electricity. For the RP methods this is given by the cost functions in the previous section. For the RD and UR methods the charge is simply the final $\lambda^\mathsf{T} y$ for each house.

For the relaxed problem itself, the true cost of generation can be different from the amount households are charged, as we are dealing with marginal prices. In addition to this, network congestion typically generates additional revenue above the cost of generation itself. An increase in cost for the integer feasible solution relative to the relaxed problem is an indication of the additional cost to the generators for balancing supply. When household charge increases rela-

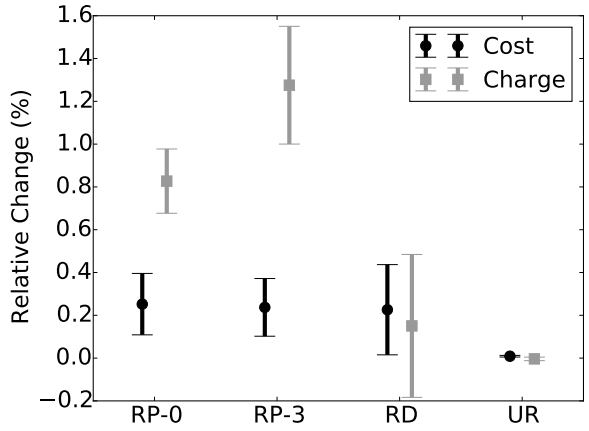

Figure 4: Change in generator cost and household charge relative to relaxed solution.

tive to the relaxed solution, this indicates that households were forced to change their consumption from the negotiated amount to ensure integer feasibility of the shiftable loads.

All methods produce costs that are within 1% of the relaxed problem, and hence also the global optimum. There is no significant difference between the methods as they reside within our estimated margin of error based on our stopping tolerance. What these results suggest is that we have a tight relaxation of the integer problem. A contributing factor is that each shiftable load only contributes a tiny amount to the overall power demand.

By artificially increasing the size of the shiftable loads by more than an order of magnitude, and heavily congesting the network, we do find instances where there is a significant gap between the relaxed solution and the candidate. However, for the realistically sized residential shiftable loads as utilised in our test microgrid, the relaxation was tight.

The charges to households are significantly higher for the RP method without gaining any benefit in terms of reduced costs. We ran the same experiments with a much smaller α, which all but eliminated charges without any increase to costs. When battery storage is introduced, we expect households will have even more flexibility in how they reach their negotiated consumptions, therefore further reducing incurred charges. This suggests that for the sole purpose of managing shiftable loads, there is no need to have a strong penalty. However, the penalty may serve a purpose for managing the effects of uncertainty in the network, and to prevent agents from lying during the negotiation stage.

All of the methods we have presented provide an efficient means for dealing with the discrete decisions in a household. Other factors such as the way they can handle uncertainty and the need for a penalty for agents gaming the system will affect the choice between these methods.

9. CONCLUSION AND FUTURE WORK

We have presented a distributed demand response mechanism for operating a microgrid. It can coordinate a whole range of distributed agents with time-coupled behaviours, whilst preserving network constraints. It also provides a natural way of pricing power in the network.

Using this mechanism we have successfully compared the performance of a range of power flow models in a meshed microgrid, and introduced simple but effective approaches to handling the shiftable loads within households. We developed a suburb-sized test microgrid, and found that the full non-convex AC equations produce close to optimal solutions in short solve times. All three of our methods for handling household shiftable loads produce close to optimal solutions with only a moderate increase in solve times.

Our work has shown that in practice distributed algorithms are not only feasible, but also highly effective at performing demand response within a microgrid context.

In future research we will investigate alternative distributed solving techniques with the aim of further improving the rate of convergence. There are opportunities for finding closed-form solutions for the exact AC equations, and to further parallelise the problem by decomposing certain components across time. It might also be possible to build a frequency regulation market into the distributed algorithm.

We need further experiments to investigate if our results carry over to larger discrete decisions, for example, those related to large industrial plant, generator start-up costs, and line switching. We also plan to answer the important question of how susceptible this mechanism is to gaming in practice, and if this is a problem, what can be done about it.

Acknowledgments

We would like to thank Hassan Hijazi for his helpful technical assistance. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

10. REFERENCES

[1] J. Andersson. *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[2] Australian Energy Market Operator (AEMO). www.aemo.com.au.

[3] X. Bai, H. Wei, K. Fujisawa, and Y. Wang. Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6âĂŞ7):383 – 392, 2008.

[4] M. Baran and F. Wu. Optimal sizing of capacitors placed on a radial distribution system. *Power Delivery, IEEE Transactions on*, 4(1):735–743, 1989.

[5] S. Bose, S. H. Low, T. Teeraratkul, and B. Hassibi. Equivalent relaxations of optimal power flow. *CoRR*, abs/1401.1876, 2014.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, Jan 2011.

[7] Z. Chen, L. Wu, and Y. Fu. Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization. *Smart Grid, IEEE Transactions on*, 3(4):1822 –1831, dec. 2012.

[8] C. Coffrin and P. V. Hentenryck. A linear-programming approximation of ac power flows. *Informs Journal on Computing*, May 2014.

[9] E. Dall'Anese, H. Zhu, and G. B. Giannakis. Distributed optimal power flow for smart microgrids. *Smart Grid, IEEE Transactions on*, 4(3):1464–1475, 2013.

[10] D. Das. Reconfiguration of distribution system using fuzzy multi-objective approach. *International Journal of Electrical Power & Energy Systems*, 28(5):331 – 338, 2006.

[11] J. Douglas, Jim and J. Rachford, H. H. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):pp. 421–439, 1956.

[12] T. Erseghe. Distributed optimal power flow using admm. *Power Systems, IEEE Transactions on*, 29(5):2370–2380, Sept 2014.

[13] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy. Inverter var control for distribution systems with renewables. In *SmartGridComm*, pages 457–462. IEEE, 2011.

[14] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17 – 40, 1976.

[15] N. Gast, J.-Y. Le Boudec, and D.-C. Tomozei. Impact of demand-response on the efficiency and prices in real-time electricity markets. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, pages 171–182, New York, NY, USA, 2014. ACM.

[16] N. Gatsis and G. Giannakis. Residential load control: Distributed scheduling and convergence with lost ami messages. *Smart Grid, IEEE Transactions on*, 3(2):770–786, June 2012.

[17] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2014.

[18] H. L. Hijazi, C. Coffrin, and P. Van Hentenryck. Convex quadratic relaxations for mixed-integer nonlinear programs in power systems. *NICTA Technical Report* `http://www.optimization-online.org/DB_HTML/2013/09/4057.html`, March 2014.

[19] HSL Archive. A collection of fortran codes for large scale scientific computation, 2014.

[20] R. Jabr. Radial distribution load flow using conic programming. *IEEE Transactions on Power Systems*, 21(3):1458–1459, Aug 2006.

[21] B. Kim and R. Baldick. A comparison of distributed optimal power flow algorithms. *Power Systems, IEEE Transactions on*, 15(2):599–604, 2000.

[22] M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, 1(2), 2014.

[23] S. Magnússon, P. C. Weeraddana, and C. Fischione. A Distributed Approach for the Optimal Power Flow Problem Based on ADMM and Sequential Convex Approximations. *ArXiv e-prints*, Jan. 2014.

[24] A. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *Smart Grid, IEEE Transactions on*, 1(3):320 –331, dec. 2010.

[25] A.-H. Mohsenian-Rad and A. Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *Smart Grid, IEEE Transactions on*, 1(2):120 –133, sept. 2010.

[26] Q. Peng and S. H. Low. Distributed Algorithm for Optimal Power Flow on a Radial Network. *ArXiv e-prints*, Apr. 2014.

[27] D. Phan and J. Kalagnanam. Some efficient optimization methods for solving the security-constrained optimal power flow problem. *Power Systems, IEEE Transactions on*, 29(2):863–872, March 2014.

[28] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Agent-based control for decentralised demand side management in the smart grid. In Tumer, Yolum, Sonenberg, and Stone, editors, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems - Innovative Applications Track (AAMAS 2011)*, pages 330–331, Taipei, Taiwan, May 2011.

[29] F. Schweppe and D. Rom. Power system static-state estimation, part ii: Approximate model. *power apparatus and systems, IEEE transactions on*, (1):125–130, 1970.

[30] P. Scott, S. Thiébaux, M. van den Briel, and P. Van Hentenryck. Residential demand response under uncertainty. In *International Conference on Principles and Practice of Constraint Programming (CP)*, pages 645–660, Uppsala Sweden, sep. 2013.

[31] M. Shinwari, A. Youssef, and W. Hamouda. A water-filling based scheduling algorithm for the smart grid. *Smart Grid, IEEE Transactions on*, 3(2):710–719, June 2012.

[32] H. Tischer and G. Verbic. Towards a smart home energy management system - a dynamic programming approach. In *Innovative Smart Grid Technologies Asia (ISGT), 2011 IEEE PES*, pages 1 –7, nov. 2011.

[33] M. van den Briel, P. Scott, and S. Thiebaux. Randomized load control: A simple distributed approach for scheduling smart appliances. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Beijing, China, August 2013.

[34] P. Šulc, S. Backhaus, and M. Chertkov. Optimal Distributed Control of Reactive Power via the Alternating Direction Method of Multipliers. *ArXiv e-prints*, 2014.

[35] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.