Coordination of Prosumer Agents via Distributed Optimal Power Flow: an Edge Computing Hardware Prototype

Demonstration

Daniel Gebbran[1], Gregor Verbič[1], Archie C. Chapman[2], Sleiman Mhanna[3*]
[1]School of Electrical and Information Engineering, University of Sydney, NSW, Australia
[2]School of Information Technology and Electrical Engineering, University of Queensland, QLD, Australia
[3]Department of Electrical and Electronic Engineering, University of Melbourne, VIC, Australia

ACM Reference Format:
Daniel Gebbran[1], Gregor Verbič[1], Archie C. Chapman[2], Sleiman Mhanna[3]. 2020. Coordination of Prosumer Agents via Distributed Optimal Power Flow: an Edge Computing Hardware Prototype. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

With the rapid rise of distributed energy resources (DER), such as rooftop photovoltaic (PV), electric vehicles and smart buildings in general, the traditional role of electrical energy consumer is shifting to one of a *prosumer* role [1]. However, the uncoordinated participation of DER may result in network problems, as violations of voltage limits or transformer capacity. One approach to implementing large-scale resource coordination is to solve an AC (alternating current) optimal power flow (OPF). This allows a network agent to optimally allocate generation resources to meet the electricity demand, respecting constraints of the physical network.

This demo presents a Raspberry Pi-based hardware prototype that coordinates participating DER agents in a low-voltage electrical network by solving a distributed optimal power flow (DOPF), which respects network constraints. We decompose the problem at the prosumer level, using the alternating direction method of multipliers (ADMM) to solve the problem in a distributed fashion. The demonstration will graphically present coordination benefits, computation times, and network status. Participants in the demonstration will be able to choose prosumers' data for the simulation, the network configuration, and different communications technologies to simulate real-world behavior in the algorithm.

Conventionally, the OPF is computed centrally, after collecting data about all agents in the network. However, the inclusion of prosumers poses a challenge for this OPF formulation - many more controllable devices are involved, and their owners have differing preferences and energy requirements. A centralized problem with hundreds or thousands of prosumers is likely to be coputationally intractable. Moreover, it violates privacy of prosumers, in a sense that each of these agents have to disclose all its information for the central, coordinating agent. A distributed OPF (DOPF) problem, instead, allows for the integration of prosumers' DER in a way that satisfies network constraints, better respecting privacy of the agents, and reducing total computation time [4, 5].

*Corresponding author's email: daniel.gebbran@sydney.edu.au.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

In our application, we decompose the OPF problem at the prosumer level. That is, instead of solving the OPF centrally, we separate it between the network agent (i.e., the aggregator)[1] that solves a single network subproblem (which minimizes network and prosumers' costs, subject to network constraints), and prosumer agents, each with one subproblem that minimizes individual costs, subject to their own constraints. These subproblems are coupled by common variables, and can be solved using the *alternating direction method of multipliers* (ADMM) [3]. Only recently has this decoupling method been developed for DOPF problems [7]. Accordingly, DOPF studies have not rigorously examined the actual distributed hardware required for each agent involved.

In order to explore this space, we develop a Raspberry Pi (RPi) edge computing prototype, using a cluster of five RPi 3B+ acting as prosumer agents. In our demonstration, a laptop acts as the aggregator, also hosting extra prosumer agents. We envisage the aggregator running its network optimisation routines on edge computing hardware located physically close to the customers, possibly incorporated into existing power network assets (e.g., substations).

To the best of our knowledge, our prototype is the first edge computing work that implements a DOPF on using multi-agent system (MAS) methods. This better reflects conditions for actual deployment of DOPF, respecting privacy of the agents to a better degree and considering communication and computation requirements.

2 PROBLEM FORMULATION

In an AC-OPF problem, the objective is to find the lowest-cost dispatch of generators that satisfies the system demand. The problem abides by physical laws (i.e. Kirchhoff's and Ohm's Laws) and technical restrictions, such as limits on bus voltages, transformer capacity and line ampacity. Additionally, including DER in the problem introduces intertemporal couplings, since they often have energy storage capacity which may act over a time horizon $\mathcal{T} = \{0, 1, ..., T\}$. This gives them flexibility to provide support to the grid and reduce the total operation cost over $\mathcal{T}$. A multiperiod OPF seeks the lowest-cost dispatch of generators and DER over $\mathcal{T}$ while accounting for operational constraints on flexible devices and all underlying $T + 1$ OPF problems. Since the DER are prosumer-owned, we model them as independent home energy management (HEM) formulations [2], controlled by a HEM computational agent. These agents form subproblems within the OPF problem, seeking to minimize their own energy cost.

[1]We envision the network agent in charge to be a *Distributed System Operator* (DSO).

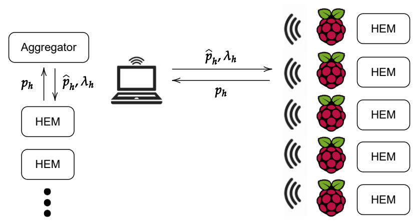

Figure 1: Demonstration flowchart for each iteration k.

Formally, a centralized, multiperiod OPF has the form [6]:

$$\underset{\boldsymbol{x} \in \mathcal{X},\, \{\boldsymbol{z_h} \in \mathcal{Z}_h\}_{h \in \mathcal{H}}}{\text{minimize}} \quad f(\boldsymbol{x}) + \sum_{h \in \mathcal{H}} g_h(z_h), \tag{1}$$

where f is the objective function associated with the aggregator agent; $\boldsymbol{x}$ is a set of aggregator variables, in a feasible set $\mathcal{X}$ defined by network constraints (e.g., voltage and power limits, nodal and power flow equations); g_h is the objective function and constraints associated with each prosumer agent $h \in \mathcal{H}$; and $\boldsymbol{z}_h$ is a set of prosumer variables, in a feasible set $\mathcal{Z}_h$ defined by constraints of each prosumer (e.g., battery constraints, power balance equations). Here, the objective function of the OPF is to minimize the connection cost $f(\boldsymbol{x})$ (i.e. a quadratic function for the sake of generality), plus the sum of all prosumers' costs $g(z_h)$.

However, solving this problem centrally incurs: (i) lack of privacy: the aggregator would have all information for each distributed agent $\boldsymbol{z}_h$, for each and every connected device used in the respective HEM; and (ii) problem intractability: the problem computation time grows rapidly with an increase in the number of prosumers.

To enable distributed computation of (1) within an edge computing framework, we need to duplicate variables that appear in between feasible sets [6]. In this formulation, we duplicate the net power profile of each prosumer for both the aggregator agent and the prosumer agent. This duplication of variables enables us to rewrite problem (1) as:

$$\underset{\hat{\boldsymbol{x}} \in \hat{\mathcal{X}},\, \{\boldsymbol{z_h} \in \mathcal{Z}_h\}_{h \in \mathcal{H}}}{\text{minimize}} \quad f(\boldsymbol{x}) + \sum_{h \in \mathcal{H}} g_h(z_h), \tag{2a}$$

$$\text{subject to:} \qquad \hat{p}_{h,t} = p_{h,t}, \ \forall\, h \in \mathcal{H},\ t \in \mathcal{T}, \tag{2b}$$

where $\hat{p}_{h,t}$ is a copy for the network problem, and $p_{h,t}$ is a copy for the prosumer problem, $p_{h,t}$; $\hat{\boldsymbol{x}}$ is the original set of problem variables in (1) with the addition of $\hat{p}_{h,t}$; and $\hat{\mathcal{X}}$ is the new feasible region of the network problem. Now, the sets of variables $\hat{\mathcal{X}}$ and $\mathcal{Z}_h$ are decoupled, and (2a) is separable if (2b) is relaxed.

The aggregator, therefore, no longer has or needs all the information about all agents in the system (only their aggregated power profile) but retains a full view of its own electrical network. In addition, HEM agents are self-interested and self-directed, and maintain a high degree of autonomy even if abiding by the aggregator coordination, which is driven by price signals.

We exploit this structure to solve (2), since ADMM exploits the decomposable structure of Problem (2) by executing alternating minimization operations over $\hat{\mathcal{X}}$ and $\mathcal{Z}$, using the two decoupled sets of variables. On a given iteration k, using a iterate set $(\boldsymbol{x}, z^k, \boldsymbol{\lambda}^k)$, the aggregator calculates its next iterate $\boldsymbol{x}^{k+1}$ by solving the aggregator subproblem (until desired convergence is obtained):

$$\begin{aligned}\hat{\boldsymbol{x}}^{k+1} := \underset{\hat{\boldsymbol{x}} \in \hat{\mathcal{X}}}{\arg\min} \quad & f(\boldsymbol{x}) + \sum_{h \in \mathcal{H}} \Big(g(z_h) + \\ & \sum_{t \in \mathcal{T}} (\lambda_{h,t}^k (\hat{p}_{h,t} - p_{h,t}^k) + \frac{\rho}{2} \left\| (\hat{p}_{h,t} - p_{h,t}^k) \right\|_2^2) \Big),\end{aligned} \tag{3}$$

where ρ is a penalty parameter and $\lambda_{h,t}$ is a dynamic price, which is the dual variable associated with each coupling constraint.

The aggregator then communicates $[\hat{\boldsymbol{p}}_h^{k+1}, \boldsymbol{\lambda}_h]_{h \in \mathcal{H}}$ to each prosumer h, respectively. The message passing between agents and their respective subproblems is illustrated in Figure 1. Next, each HEM calculates an update for $\boldsymbol{z}_h^{k+1}$ using the iterate set $(\boldsymbol{x}_h^{k+1}, \boldsymbol{z}_h, \boldsymbol{\lambda}_h^k)$:

$$\begin{aligned}z_h^{k+1} := \underset{z_h \in \mathcal{Z}_h}{\arg\min} \quad & g(z_h) + \\ & \sum_{t \in \mathcal{T}} (\lambda_{h,t}^k (\hat{p}_{h,t}^{k+1} - p_{h,t}) + \frac{\rho}{2} \left\| (\hat{p}_{h,t}^{k+1} - p_{h,t}) \right\|_2^2),\end{aligned} \tag{4}$$

and communicates the resulting $\boldsymbol{p}_h^{k+1}$ back to the aggregator. The final step at iteration k is the dual update:

$$\lambda_{h,t}^{k+1} := \lambda_{h,t}^k + \rho(\hat{p}_{h,t}^{k+1} - p_{h,t}^{k+1}) \quad \forall\, h \in \mathcal{H},\ t \in \mathcal{T}, \tag{5}$$

which takes place at the aggregator.

On one hand, under the ADMM scheme the HEM agents better preserve their privacy because the only variable they share is their aggregate power consumption profile[2]. Moreover, they are in control of their own subproblem, with its scope clearly defined as a self-interest optimization problem. On the other hand, using this edge computing framework greatly alleviates the computational burden imposed at the aggregator (3).

Literature analyzing computation and communication requirements for real-world hardware deployment of this technique is scarce. Our demonstration innovates in implementing this MAS in distributed hardware with autonomous HEM agents.

3 DEMONSTRATION

The test cases are composed of twenty-five or fifty prosumer agents and one aggregator agent. A 24-hour receding horizon time window is used for the DOPF, with $T = 48$ time steps. Each prosumer has fixed loads, PV generation and a battery, and is subject to a time-of-use (ToU) and feed-in (FiT) tarrifs. Five prosumer agents are modeled in RPis (each with 1 GB RAM, 1.4 GHz) and execute subproblem (4) within the RPi. The laptop (16 GB RAM, 2.80 GHz) executes the aggregator subproblem (3), the dual update (5), and any remaining prosumers.

Attendees will be able to interact by choosing two different power networks; modifying the PV power generated; and selecting different communications technologies (e.g., 3G, 4G). This will allow for attendees to explore different problems' complexity, PV penetration, and the impact of latency in the distributed algorithm.

The output of the simulation will show, in graphical format: (i) number of iterations and total algorithm execution time; (ii) prosumers' tariffs; (iii) system power flow; (iv) prosumers' power consumption; and (v) end-of-line voltage profile over time.

[2]There is no disclosure of sensitive information such as individual appliances, or battery and vehicle charging rates or states.

REFERENCES

[1] AEMO, Energy Networks Australia. 2018. *Open Energy Networks*. Technical Report.

[2] Donald Azuatalam, Kaveh Paridari, Yiju Ma, Markus Förstl, Archie C. Chapman, and Gregor Verbič. 2019. Energy management of small-scale PV-battery systems: A systematic review considering practical implementation, computational requirements, quality of input data and battery degradation. *Renewable and Sustainable Energy Reviews* 112 (2019), 555–570. https://doi.org/10.1016/j.rser.2019.06.007

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3 (Jan 2011), 1–122. https://doi.org/10.1561/2200000016

[4] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. In press. Differential Privacy for Power Grid Obfuscation. *IEEE Transactions on Smart Grid* (In press).

[5] Sleiman Mhanna, Gregor Verbič, and Archie C. Chapman. 2017. A Component-Based Dual Decomposition Method for the OPF Problem. *CoRR* abs/1704.03647 (2017).

[6] Sleiman Mhanna, Gregor Verbič, and Archie C. Chapman. 2019. Adaptive ADMM for Distributed AC Optimal Power Flow. *IEEE Transactions on Power Systems* 34, 3 (May 2019), 2025–2035. https://doi.org/10.1109/TPWRS.2018.2886344

[7] Paul Scott, Dan Gordon, Evan Franklin, Laura Jones, and Sylvie Thiébaux. 2019. Network-Aware Coordination of Residential Distributed Energy Resources. *IEEE Transactions on Smart Grid* (2019). https://doi.org/10.1109/TSG.2019.2907128