Cooperative Load Scheduling for Multiple Aggregators Using Hierarchical ADMM

Preprint

Xiangyu Zhang, Dave Biagioni, Peter Graf, and Jennifer King

National Renewable Energy Laboratory

Presented at the 2020 IEEE Conference on Innovative Smart Grid Technologies (IEEE ISGT)
Washington, D.C.
February 17–20, 2020

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-2C00-76096
March 2020

Cooperative Load Scheduling for Multiple Aggregators Using Hierarchical ADMM

Preprint

Xiangyu Zhang, Dave Biagioni, Peter Graf, and Jennifer King

National Renewable Energy Laboratory

Suggested Citation
Zhang, Xiangyu, Dave Biagioni, Peter Graf, and Jennifer King. 2020. *Cooperative Load Scheduling for Multiple Aggregators Using Hierarchical ADMM: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-2C00-76096. https://www.nrel.gov/docs/fy20osti/76096.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-2C00-76096
March 2020

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the Autonomous Energy Systems project funded by the National Renewable Energy Laboratory's Laboratory Directed Research and Development program. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Cooperative Load Scheduling for Multiple Aggregators Using Hierarchical ADMM

Xiangyu Zhang, Dave Biagioni, Peter Graf, Jennifer King
National Renewable Energy Laboratory, Golden, Colorado, USA 80401

***Abstract*—Demand response (DR) serves an important role in improving the efficiency and stability of power systems. In recent years, with advances in communication and smart device technologies, many aggregators have emerged to facilitate end customer participation in DR programs. These aggregators, equipped with customized optimal control algorithms, are capable of providing various grid services. Among them is load scheduling during DR events, namely following a load signal provided by the utility company while minimizing overall customer discomfort. However, as the number of aggregators keeps increasing, it becomes challenging for utility companies to conduct load scheduling for multiple aggregators and generate reference signals for each of them. This paper proposes an optimization framework using hierarchical alternating direction method of multipliers (H-ADMM) to optimally generate load following signals for multiple aggregators. Under this framework, utility and multiple aggregators work in a cooperative manner, aiming at minimizing an overall system cost from different levels of the power system hierarchy, while protecting user privacy. A case study has been conducted in a system with multiple aggregators, based on control of HVAC loads. Experimental results validate the effectiveness of the proposed algorithm.**

***Keywords*—Demand response, aggregator, load scheduling, ADMM.**

I. Introduction

Demand response (DR) is an effective approach to increase power grid operation efficiency [1], [2]. Examples of benefits include avoiding the operation of less efficient and more expensive generation units, and balancing intermittent renewable generation [3]. Research on implementing demand response has become popular; in single customer scenarios, for example, the authors of [4] discussed load scheduling at device level given day-ahead price and an algorithm for home energy management system is designed in [5]. On the other hand, to facilitate residential and commercial customers to participate in DR programs, a great number of aggregators have emerged in recent years. Authors of [6] studied a distributed direct load control for an aggregated residential load. The study in [7] proposed a coordination of electric vehicles charging among multiple aggregators. In the present day, aggregators are well equipped with proprietary optimal control algorithms that can control the device cluster to follow the load signal provided by the utility company, with minimized customer discomfort incurred. With an increasing number of such aggregators entering the marketplace, there is an emerging opportunity for utility companies to exploit demand responsiveness to effect coordinated control in the form of load following. This important problem has not been well addressed in the literature.

We propose an optimization framework that allows a utility company and multiple aggregators to solve the power allocation sharing problem in a cooperative effort. Assuming each aggregator has multiple device clusters, the sharing problem is solved iteratively using a three-layer tree structure (utility/aggregators/clusters). The proposed algorithm, which we call hierachical-ADMM (H-ADMM), is organized according to this tree structure and, importantly, shares information between layers so as to protect end user and aggregator privacy. H-ADMM consists of one *upper* and multiple *lower* control levels and is characterized by a specific combination of block-separable and shared terms in the global objective that naturally lead to the hierarchical decomposition. In addition, the primal and dual updates performed at each level have a simple but meaningful interpretation in terms of the various agents that can be easily understood in the context of cooperative control. For ease of exposition, we focus on finite-horizon control with perfect forecasting, but note that the formulation is easily extended to incorporate receding-horizon model predictive control. Finally, while the proposed control hierarchy consists of only two control levels, the scheme can be generalized to arbitrary depth for suitable problems.

II. Formulation of Load Scheduling

Fig. 1 shows a utility company and N aggregators, each with multiple device clusters. The aggregators have the autonomy to determine how to group the cluster, e.g., by device type or geographical feeders, and how to implement optimal cluster control for individual devices within each cluster, e.g., via greedy control, reinforcement learning or other optimal control algorithms) To capture the current reality, the detail of cluster level control is assumed to be proprietary and unknown to the utility. For a control horizon with T discrete steps, we assume $x_{ij} \in \mathcal{R}_+^T$ is the power allocated to Cluster j in Aggregator i, and $x_i = \sum_{j=1}^{N_i} x_{ij}$ is the power consumed by Aggregator i. Here $\mathcal{R}_+^T$ denotes the space of T-dimensional real-valued vectors with nonnegative components. The nonnegative vector $\beta \in \mathcal{R}_+^T$ represents the amount of power procured from the day-ahead wholesale market, in order to satisfy aggregator demand. For the total power to track the load signal β, costs are incurred at all three levels and our objective is to minimize this total cost.

The total cost minimization is formulated as:

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad \sum_{i=1}^{N} f_i(x_i) + \epsilon \left\| \sum_{i=1}^{N} x_i - \beta \right\|_2^2 . \tag{1}$$

Here $f_i : \mathcal{R}_+^T \to \mathcal{R}$ represents the costs from Aggregator i and its constituent clusters and the second term represents

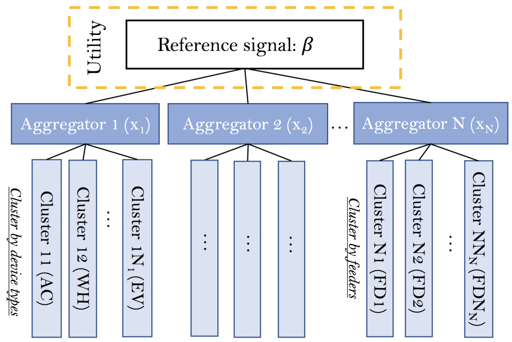


Fig. 1: Overview of load scheduling problem.

the utility-level penalty for deviation from the aggregate load signal with per-unit penalty ϵ. We assume that aggregators have two (possibly conflicting) objectives: to minimize the penalty for drawing power beyond a predefined limit, and to minimize customer discomfort. Specifically,

$$f_i(x_i) = \sum_{j=1}^{N_i} f_{ij}(x_{ij}) + \eta(\sum_{j=1}^{N_i} x_{ij} - \tau_i)^+ \tag{2}$$

where $f_{ij}(x_{ij})$ represents overall customer discomfort within a cluster, τ_i is the upper limit of power consumption agreed upon between Aggregator i and the utility, and η is the associated per-unit cost of exceeding this limit. The function $(\cdot)^+$ represents element-wise projection onto the nonnegative real line, i.e., $(y)^+ = \max\{0, y\}$. In practice, when the optimal x^*_{ij} is obtained, it is sent as a load following signal to the corresponding cluster and the existing proprietary controller at cluster level will use x_{ij} for device level control.

III. Hierarchical ADMM

In this section we describe H-ADMM and highlight both the hierarchical, decentralized nature of the algorithm as well as its ability to protect private state information via limited information passing between the layers.

We first review ADMM in the context of "sharing" problems. ADMM is a convex optimization algorithm that has recently gained popularity in problems where the structure of the objective function (and constraints encoded therein) admits a decomposition into subproblems that can be iteratively solved to arrive at a global solution via a convergent sequence of primal-dual updates. One such class of problems have objectives of the form $\sum_{i=1}^{N} f_i(x_i) + g(\sum_{i=1}^{N} x_i)$, and these problems are often referred to as "sharing" problems since they combine separable terms, $f_i(x_i)$, with a joint term, $g(\sum_{i=1}^{N} x_i)$, that depends on the cumulative (shared) value of the individual decisions. We refer the reader to [8] for details about this decomposition and the associated primal-dual algorithm.

In order to derive the hierarchical scheme, we first note that the upper level optimization problem (1) is, in fact, a sharing problem: each aggregator has its own local objective encoded via f_i in addition to a cooperative objective to follow the net load signal. Ignoring for the moment that each aggregator decision is a sum over its clusters' decisions, the ADMM algorithm for this problem is expressed via the following primal-dual (scaled) updates,

$$x_i^{k+1} = \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}}(f_i(x_i) + \frac{\rho_u}{2} \left\| x_i - x_i^k + \bar{x}^k - \bar{z}^k + u^k \right\|_2^2) \tag{3}$$

$$\bar{z}^{k+1} = \underset{\bar{z}}{\operatorname{argmin}}(\epsilon \left\| N\bar{z} - \beta \right\|_2^2 + \frac{N\rho_u}{2} \left\| \bar{z} - \bar{x}^{k+1} - u^k \right\|_2^2) \tag{4}$$

$$u^{k+1} = u^k + \bar{x}^{k+1} - \bar{z}^{k+1}. \tag{5}$$

Here the superscript k denotes the ADMM iteration number, $\bar{x} = \sum_i^N x_i$, and $\rho_u > 0$. Note that $\bar{z}$ is an intermediate optimization variable; the overbar here reflects the fact that $\bar{z}$ is an estimator of $\bar{x}$.

The key observation of this paper is that, considering $x_i = \sum_{j=1}^{N_i} x_{ij}$ and definition (2), the aggregator update rules x_i^{k+1} in (3) themselves have the structure of a sharing problem: the quadratic penalty term is a function of $x_i = \sum_{ij} x_{ij}$ while all other quantities $x_i^k, \bar{x}^k, \bar{z}^k, u^k$ are simply iteration parameters. For this reason, we consider solving the x_i-update in the upper level via another level of ADMM,

$$x_{ij}^{k+1} = \underset{x_{ij} \in \mathcal{X}_{ij}}{\operatorname{argmin}}(f_{ij}(x_{ij}) + \frac{\rho_l}{2} \left\| x_{ij} - x_{ij}^k + \bar{x}_i^k - \bar{w}_i^k + y_i \right\|_2^2) \tag{6}$$

$$\bar{w}_i^{k+1} = \underset{\bar{w}_i}{\operatorname{argmin}}(h(N_i \bar{w}_i) + \frac{N_i \rho_l}{2} \left\| \bar{w}_i - \bar{x}_i^{k+1} - y_i^k \right\|_2^2) \tag{7}$$

$$y_i^{k+1} = y_i^k + \bar{x}_i^{k+1} - \bar{w}_i^{k+1} \tag{8}$$

where $\rho_l > 0$. In (7), the function h encodes the shared terms that include both aggregator power limit penalty as well as the quadratic penalty from the upper level iteration,

$$h(v) = \eta(v - \tau_i)^+ + \frac{\rho_u}{2} \left\| v - x_i^k + \bar{x}^k - \bar{z}^k + u^k \right\|_2^2. \tag{9}$$

Taken together, the single set of upper level updates (3)-(5) and $\sum_i N_i$ lower level updates (6)-(8) comprise a bilevel hierarchical control scheme. We note that while aggregator decisions x_i in the top level updates (3) are ultimately passed down to the individual devices in the lower level, the control scheme is indeed hierarchical. In particular, one can interpret the upper level $\bar{z}$- and u-updates as the action of the utility to reconcile the joint objectives of the aggregators with the system-wide objective of following the load signal. It does so by adjusting both the target net power consumption (4) as well as the cost of deviating from the target (5). The lower level operates analogously, with each aggregator now attempting to reconcile the cluster objectives with its own load following objective. In practice, (4) and (5) are completed at the utility level; (7) and (8) are completed at the aggregator level and (6) is completed at the cluster level, only limited and non-private data are shared between levels. Fig. 2 shows the computational flow chart.

IV. Case Study

In this section, the efficacy of the proposed H-ADMM algorithm is first verified using a simplified cluster model and compared against two baseline control algorithms. Numerical

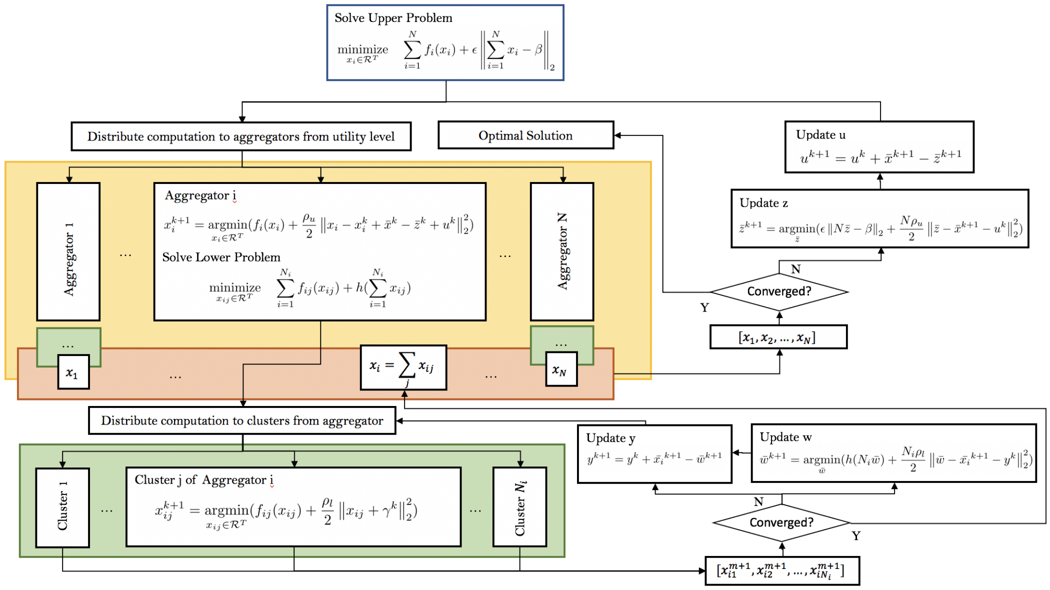


Fig. 2: Distributed Architecture for the Hierarchical ADMM.

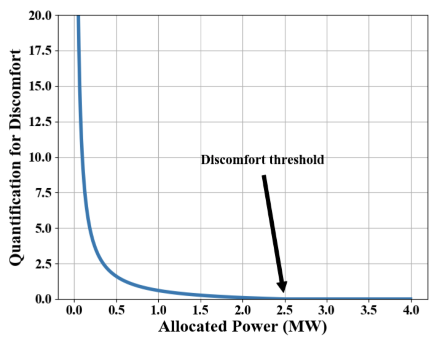


Fig. 3: Example of a discomfort function for simplified aggregator model.

results shown in this section are based on Python implementation of the proposed algorithm using open source libraries 'ipopt/Pyomo' and 'pyswarms'.

A. Simulation Setup

1) Simplified cluster discomfort model: For simplicity, we first assume for any step in control horizon, the discomfort can be computed using (10), where a_{ij} is discomfort threshold.

$$f_{ijt}(x_{ij}[t]) = \begin{cases} \frac{1}{x_{ij}[t]} - \frac{1}{a_{ij}} & (0 < x_{ij}[t] < a_{ij}) \\ 0 & (x_{ij}[t] \geq a_{ij}) \end{cases} \tag{10}$$

The rationale behind this assumption is that typically the discomfort of cluster decreases with the increase of x_{ij} and increase drastically when x_{ij} is insufficient. Figure 3 is an example of such discomfort function for a cluster, where $a_{ij} = 2.5$. Based on this assumption, the discomfort for the whole control horizon can be expressed via the convex function $f_{ij}(x_{ij}) = \sum_{t=1}^{\mathcal{T}} f_{ijt}(x_{ij}[t])$.

2) Scenario setup: We consider optimizing in 15-minute intervals over a two hour horizon ($\mathcal{T} = 8$) and that the utility-level load signal to be followed is given by: $\beta = [28., 16., 20., 22., 18., 24., 15., 18.]$ in units of MW. In this problem, the system consists of three aggregators and each of them has two clusters. The power limit at aggregator level is $\tau = [9.0, 8.0, 11.0]$ and the discomfort threshold for each cluster in all three aggregators are $[[2.0, 5.0], [3.0, 2.0], [3.0, 5.0]]$. Unit penalty for exceeding aggregator-level power limit is $\eta = 5.0$ and the overall load deviation cost is $\epsilon = 1.0$. Maximum iteration number of upper and lower problems are 100 and 200 respectively, but iteration stops early when update is less than 5e-3. Parameters ρ_u and ρ_l are chosen to be 1.0 and 0.5 respectively.

B. Baseline algorithms

1) Baseline 1: Proportional power allocation: In contrast to the proposed framework, Baseline 1 does not allow cooperation and sharing among all aggregators/clusters. Instead, the utility company allocates power to each aggregator and cluster in an arbitrary but reasonable way: by dividing the power proportionally as

$$x_i[t] = \beta[t] \cdot \frac{\tau_i}{\sum_{i=1}^{N} \tau_i} \tag{11}$$

$$x_{ij}[t] = x_i[t] \cdot \frac{a_{ij}}{\sum_{j=1}^{N_i} a_{ij}} \tag{12}$$

By comparing H-ADMM with Baseline 1, it can be seen whether introducing the flexibility of allowing cooperation among aggregators/clusters provides any advantage.

2) Baseline 2: Centralized optimization: By plugging (10) into (1) and (2), a centralized version of the proposed optimization problem is formulated as (13). Comparison of H-ADMM with Baseline 2 provides a measure of accuracy since Baseline 2 is mathematically equivalent and can be readily solved to global optimality due to the convexity of global problem. We note that Baseline 2, however, assumes complete knowledge of all state and decision variables and thus does not support privacy protection or distributed solution.

$$
\begin{aligned}
\underset{x_{ij}}{\text{minimize}} \quad & \sum_{i=1}^{N}\left[\sum_{j=1}^{N_i}\sum_{t=1}^{T} d_{ij} + \eta \sum_{t=1}^{T}(x_i[t]-\tau_i)^{+}\right] \\
& + \epsilon \left\| \sum_{i=1}^{N} x_i - \beta \right\|_2^2 \qquad\qquad (13) \\
\text{subject to} \quad & d_{ij} \geq 0, d_{ij} \geq \frac{1}{x_{ij}} - \frac{1}{a_{ij}}, \\
& x_i = \sum_{j=1}^{\mathcal{N}_i} x_{ij}\,(\forall i)\,, x_{ij} \in \mathcal{R}^{+}
\end{aligned}
$$

C. Simulation Results

Table I shows the comparison between H-ADMM and two baseline approaches. First, the comparison between proportional power allocation and H-ADMM shows that the introduction of cooperation among aggregators/clusters in our proposed framework can reduce the total cost. Cost reduction is attributed to the decrease of discomfort cost. Table II shows the change of cluster power consumption over the control horizon. The fluctuation of cluster's power reflects the cooperation and resource sharing among aggregators. Mathematically speaking, the power is directed to clusters with the highest marginal discomfort. Second, the comparison between the centralized optimization with H-ADMM shows the total cost (C_{total}) obtained by both algorithms are very close with the difference attributable to the early stopping criteria of H-ADMM. Fig. 4 compares the power allocated to each aggregator by H-ADMM and its centralized version, it demonstrates the results from H-ADMM closely agrees with those from the centralized optimization. Fig. 5 shows that H-ADMM can coordinate these six clusters and three aggregators to closely follow the utility-level load.

TABLE I: PERFORMANCE COMPARISON

Algorithm	C_{total}	C_{dis}	C_{lf}	C_{agg}
Proportional Allocation	1.6280	1.6280	0.0000	0.0000
Centralized Optimization	0.8453	0.8361	0.0092	0.0000
H-ADMM	0.8557	0.8422	0.0098	0.0036

TABLE II: LOAD PERCENTAGE FOR SIX CLUSTERS

Cluster	Baseline 1	H-ADMM			
	Fixed (%)	Max (%)	Min (%)	Mean (%)	(Max-Min)/Mean (%)
1	9.18	14.12	9.51	12.11	38.14
2	22.96	23.33	18.11	20.27	25.75
3	17.14	20.00	14.91	16.88	30.14
4	11.43	15.07	10.00	12.72	39.85
5	14.73	18.71	12.88	16.41	35.54
6	24.55	25.43	17.90	21.76	34.63

It is worth noting that the centralized optimization problem above is only used to verify the optimality of H-ADMM solution; in a real life application, the centralized problem usually cannot be formulated and solved because 1) aggregators are not willing to expose their system discomfort models which may be both proprietary and contain sensitive data and 2) the cluster discomfort models might not be convex or, worse, even have an explicit form.

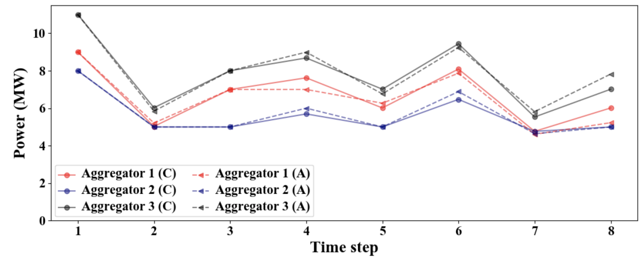


Fig. 4: Comparison of load allocation from two algorithms (centralized optimization (C) and hierarchical ADMM (A)) at aggregator level.

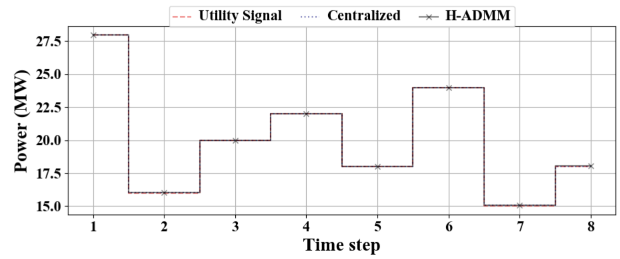


Fig. 5: Results of load following by two algorithms.

D. Preliminary Results on Detailed Cluster Discomfort Model

As previously mentioned, in real applications $f_{ij}(x_{ij})$ can be expected to be more complicated than (10), and might not even have an explicit form. In this section, preliminary results are presented for exactly such a system; we refer to the corresponding model as the *detailed* cluster model.

1) Detailed Cluster Discomfort Model: In this section, we assume all clusters consist of only HVAC devices and they are controlled via a greedy control algorithm, as shown in Algorithm 1. For simplicity, all AC units are assumed to have the same power consumption of P_{ac}. Z represents the set of all AC units in the cluster and $\mathcal{F}_\zeta$ is the thermal dynamic model for AC unit ζ ($\forall \zeta \in Z$), which is known to the cluster. Evidently, $f_{ij}(x_{ij})$ in this case does not have an explicit form and we investigate a gradient-free method to optimize (6).

Algorithm 1 $f_{ij}(x_{ij})$ Evaluation using Cluster Greedy Control

Input: x_{ij}

```
f_ij(x_ij) = 0
for t = 1 to T do
    Determine number of AC units to be turned ON:
    ψ = int(x_ij[t]/P_ac)
    Evaluate thermal comfort margin for all AC units:
    m_ζ^t = T_set,ζ + ΔT_tolerance − T_ζ^t
    Sort AC units by m_ζ (from least margin to max margin)
    and select Top ψ AC units in the priority queue to cool
    (s_i^t = 1), otherwise (s_i^t = 0).
    for ζ = 1 to |Z| do
        Update indoor temperature: T_ζ^{t+1} = F_ζ(T_ζ^t, s_ζ^t)
        f_ij(x_ij) = f_ij(x_ij) + max(0, −λ m_ζ^{t+1})
    end for
end for
return f_ij(x_ij)
```

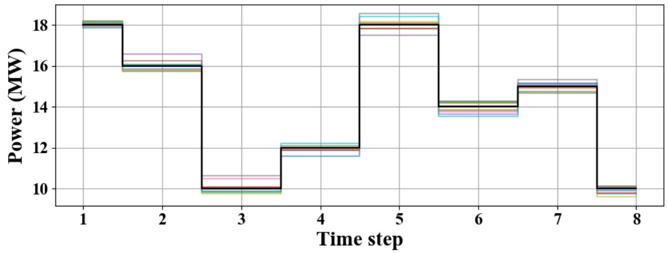


Fig. 6: Load following results at utility-level when using detailed cluster model. Black line shows the signal, and 10 colored lines are actual load profiles as H-ADMM solutions.

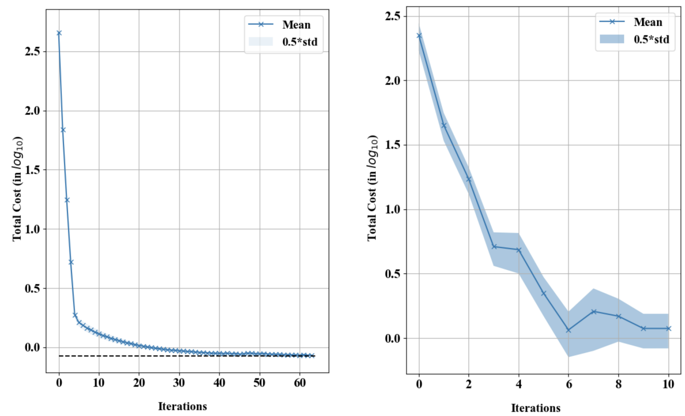
2) Scenario setup and results: Similar to earlier simulation, we assume there are three aggregators and each one has two clusters. The numbers of AC units in all clusters are shown in $[[300, 500], [400, 500], [500, 500]]$. Each unit has 8kW power consumption when cooling. The utility-level load signal to be followed is : $\beta = [18., 16., 10., 12., 18., 14., 15., 10.]$ in units of MW. The power limits at the aggregator level are $\tau = [6.0, 6.5, 7.0]$. The initial indoor temperatures of all houses are sampled from a Gaussian distribution ($T_\zeta^0 \sim \mathcal{N}(72.0, 1.0)$, in $^\circ F$). In this study, particle swarm optimization is used to solve (6) and the number of upper iteration is limited to 10. Fig. 6 shows the optimized load following results at the utility level. Due to the stochastic feature of heuristic optimization, 10 trials are conducted. Fig. 6 shows in all 10 trials the utility level loads are closely following the signal given. Two reasons for the small amount of deviation: first, at some times load following is sacrificed in exchange for cluster comfort, in order to minimize the overall objective of (1); second, due to the early stopping of iterations, the final solution, though close to the optimal solution, is sub-optimal.

E. Algorithm Convergence

Previous studies have proven the convergence of ADMM under convexity assumptions on the objective function and its domain. When using the simplified discomfort model, (10) is convex which guarantees the convergence of H-ADMM when solving the problem. Fig. 7 (a) shows the convergence of total cost (1) in log10 for the case with simplified cluster model (a). In (a), the black dashed line is the optimal objective value from the centralized solver, to which the H-ADMM solution is shown to converge. Fig. 7 (b) shows the convergence of gradient-free optimization using the detailed cluster model. Given the discrete nature of the detailed objective, the problem is not convex and thus not guaranteed to converge. However, empirical evidence suggests the algorithm may provide a reasonable heuristic solution; further investigation of these issues will be provided in a future work.

V. Discussion and Future Work

In this study, we proposed a load scheduling framework that has the following merits: 1) the problem formulation allows cooperation and negotiation among aggregators/clusters to collectively and optimally determine the power allocation; 2) the framework is built on existing hierarchical aggregator infrastructure in which clusters are delegates to reflect lower level demand in the form of discomfort and to implement optimal control at device level; 3) similar to real life scenarios,



(a) Simple cluster model

(b) Detailed cluster model

Fig. 7: Convergence rates for both simple model (a) and detailed model (b): log of total cost vs. upper layer iterations. Standard deviation is calculated using results from 10 trials.

autonomy is provided to aggregators on how to implement cluster control. Furthermore, H-ADMM does not require aggregators to share control and model details to a centralized solver and thus guarantees user privacy. In addition, when $f_{ij}(x_{ij})$ is not differentiable, distributed H-ADMM might be more computationally tractable than a centralized approach. Future works will 1) investigate the empirical and theoretical convergence of the proposed algorithm when using detailed models of $f_{ij}(x_{ij})$; 2) properly design prices and incentives so that multiple aggregators are willing to cooperate as described by this framework; 3) investigate techniques to increase the overall computational efficiency of H-ADMM.

References

[1] Mohamed H Albadi and Ehab F El-Saadany. A summary of demand response in electricity markets. *Electric power systems research*, 78(11):1989–1996, 2008.

[2] Farshid Shariatzadeh, Paras Mandal, and Anurag K Srivastava. Demand response for sustainable energy systems: A review, application and implementation strategy. *Renewable and Sustainable Energy Reviews*, 45:343–350, 2015.

[3] B Durga Hari Kiran and M Sailaja Kumari. Demand response and pumped hydro storage scheduling for balancing wind power uncertainties: A probabilistic unit commitment approach. *International Journal of Electrical Power & Energy Systems*, 81:114–122, 2016.

[4] Nikolaos G Paterakis, Ozan Erdinc, Anastasios G Bakirtzis, and João PS Catalão. Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies. *IEEE Transactions on Industrial Informatics*, 11(6):1509–1519, 2015.

[5] Phani Chavali, Peng Yang, and Arye Nehorai. A distributed algorithm of appliance scheduling for home energy management system. *IEEE Transactions on Smart Grid*, 5(1):282–290, 2014.

[6] Chen Chen, Jianhui Wang, and Shalinee Kishore. A distributed direct load control approach for large-scale residential demand response. *IEEE Transactions on Power Systems*, 29(5):2219–2228, 2014.

[7] Zhiwei Xu, Zechun Hu, Yonghua Song, Wei Zhao, and Yongwang Zhang. Coordination of pevs charging across multiple aggregators. *Applied energy*, 136:582–589, 2014.

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.