Cooperative Load Scheduling for Multiple Aggregators Using Hierarchical ADMM

Xiangyu Zhang; Dave Biagioni; Peter Graf; Jennifer King
National Renewable Energy Laboratory, Golden, CO, 80401, U.S.A.

NREL NATIONAL RENEWABLE ENERGY LABORATORY

Introduction

In recent years, with advances in communication and smart device technologies, many aggregators have emerged to facilitate customer participation in Demand Response (DR). Aggregators, equipped with customized optimal control, can provide load scheduling services during DR events: following a load signal provided by the utility while minimizing overall customer discomfort. However, as the number of aggregators increases, it becomes challenging for utility companies to generate reference signals for each of them, especially considering that the aggregators' control algorithms are proprietary. The proposed work facilitates orchestrating multiple aggregator in a distributed way to fulfill load scheduling while privacy of aggregators is preserved.

Problem Formulation

Fig. 1 shows a utility company and N aggregators, each with N_i load clusters. Each cluster can optimally control all loads in the cluster (e.g., 1000 AC units) using a proprietary control algorithm. For a T-step control horizon, assuming $x_{ij} \in \mathcal{R}_+^T$ is the power allocated to Cluster j in Aggregator i, whose power consumption is $x_i = \sum_{j=1}^{N_i} x_{ij}$. The goal is to properly allocate x_{ij}, so that the total load $\sum_{i=1}^{N} x_i$ follows a signal β.

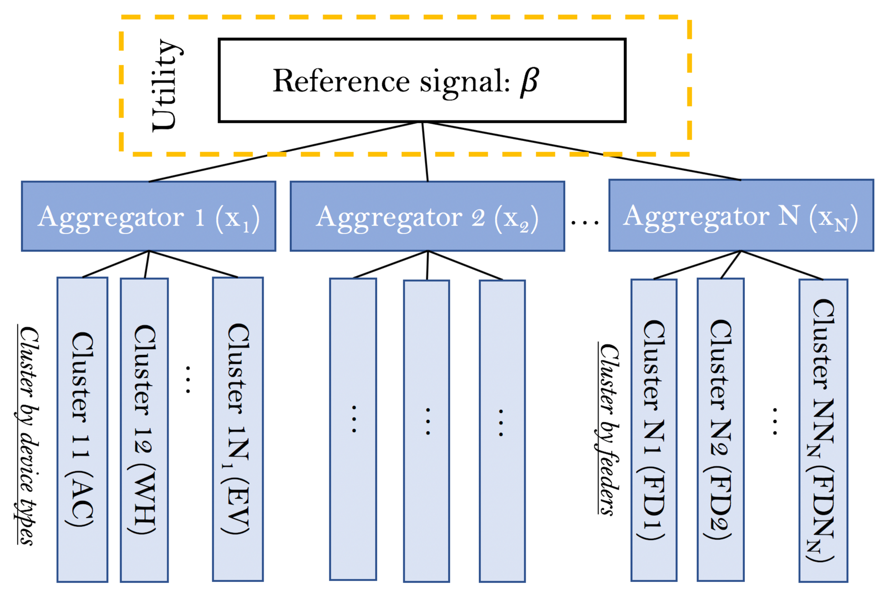

Fig. 1: Market participants for DR load scheduling.

Objective: Minimizing combined costs from all three types of participants (Utility, aggregators and clusters) for achieving the load following objective.

Total cost :

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

Aggregator and all its clusters' cost:

$$f_i(x_i) = \sum_{j=1}^{N_i} f_{ij}(x_{ij}) + \eta\Big(\sum_{j=1}^{N_i} x_{ij} - \tau_i\Big)^+ \qquad (1.b)$$

Cluster level discomfort cost:

$$f_{ij}(x_{ij}) = inf_*\mathcal{D}(x_{ij}) = inf_{\boldsymbol{s}}\boldsymbol{\mathcal{D}}(x_{ij}, \boldsymbol{s}) \qquad (1.c)$$

In (2), $(\zeta)^+ = \max\{0, \zeta\}$. $inf_{\boldsymbol{s}}\boldsymbol{\mathcal{D}}(x_{ij}, \boldsymbol{s})$ in (3) is the proprietary optimal control unknown to the utility, aiming at minimizing discomfort.

Hierarchical ADMM

Alternating direction method of multipliers (ADMM) is an optimization algorithm that can be used 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 specific problem with objective function $\sum_{i=1}^{N} f_i(x_i) + g(\sum_{i=1}^{N} x_i)$ is known as 'sharing problem' [1] is the building block of the proposed algorithm. When using ADMM to solve other ADMM subproblems, the multi-level hierarchy is introduced and that is hierarchical ADMM (H-ADMM).

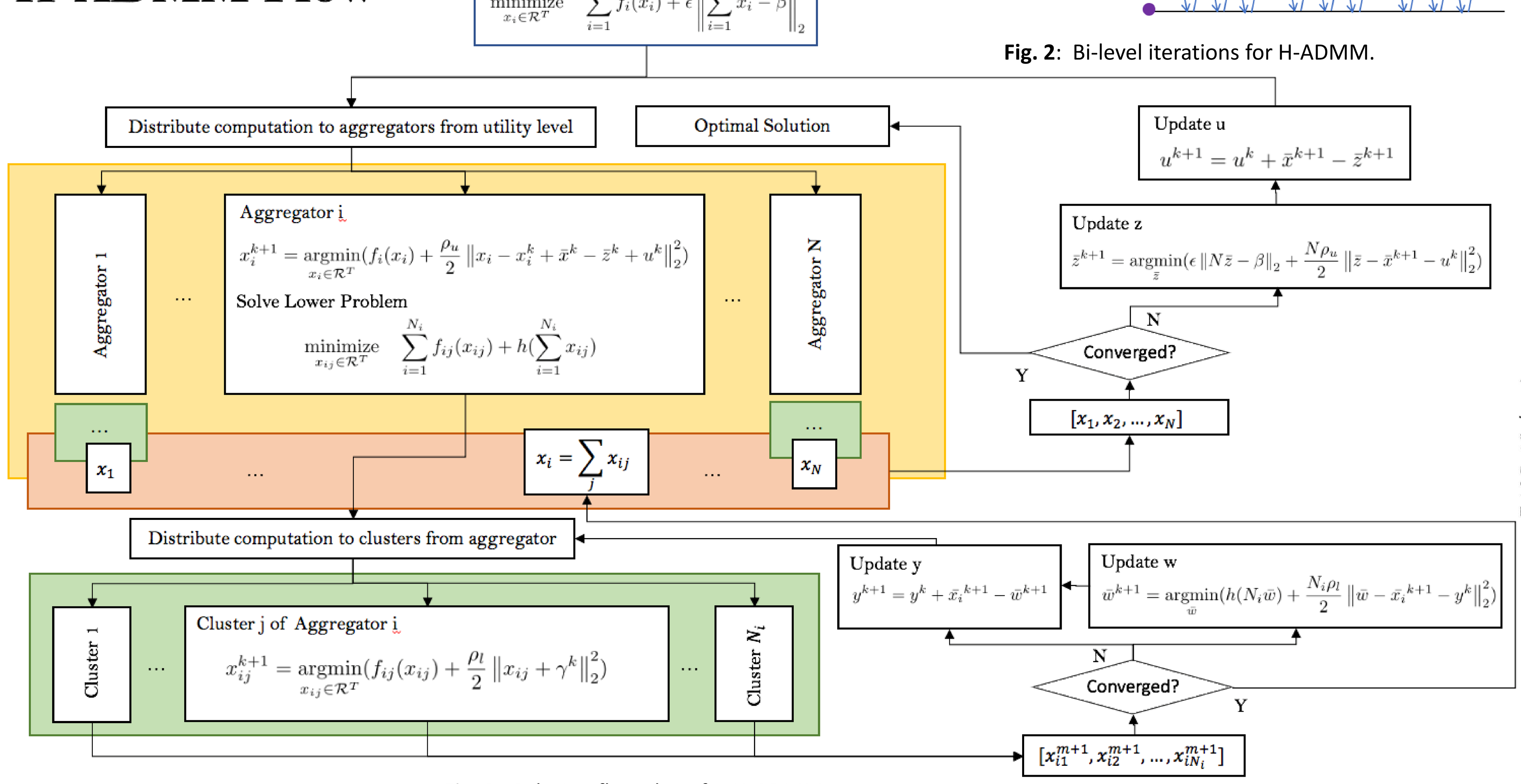

Fig. 3: Solution flow chart for H-ADMM.

According to [1], the first level problem (1.a) can be solved using the following iteration steps.

$$x_i^{k+1} = \underset{x_i \in \mathcal{X}_i}{\text{argmin}}\Big(f_i(x_i) + \frac{\rho_u}{2}\left\| x_i - x_i^k + \bar{x}^k - \bar{z}^k + u^k \right\|_2^2\Big) \qquad (2.a)$$

$$\bar{z}^{k+1} = \underset{\bar{z}}{\text{argmin}}\Big(\epsilon \left\| N\bar{z} - \beta \right\|_2^2 + \frac{N\rho_u}{2}\left\| \bar{z} - \bar{x}^{k+1} - u^k \right\|_2^2\Big) \qquad (2.b)$$

$$u^{k+1} = u^k + \bar{x}^{k+1} - \bar{z}^{k+1} \qquad (2.c)$$

Considering (1.b) and (2.a), the x-update above is in form of another ADMM sharing problem (see (3)), and thus can be again solved by the iterative solution (4.a-c).

$$\underset{x_{ij} \in \mathcal{X}}{\text{minimize}} \quad \sum_{j=1}^{N_i} f_{ij}(x_{ij}) + \eta\Big(\sum_{j=1}^{N_i} x_{ij} - \tau_i\Big)^+ + \frac{\rho_u}{2}\left\| \sum_{j=1}^{N_i} x_{ij} - x_i^k + \bar{x}^k - \bar{z}^k + u^k \right\|_2^2 \qquad (3)$$

$$x_{ij}^{k+1} = \underset{x_{ij} \in \mathcal{X}_{ij}}{\text{argmin}}\Big(f_{ij}(x_{ij}) + \frac{\rho_l}{2}\left\| x_{ij} - x_{ij}^k + \bar{x}_i^k - \bar{w}_i^k + y_i \right\|_2^2\Big) \qquad (4.a)$$

$$\bar{w}_i^{k+1} = \underset{\bar{w}_i}{\text{argmin}}\Big(h(N_i\bar{w}_i) + \frac{N_i\rho_l}{2}\left\| \bar{w}_i - \bar{x}_i^{k+1} - y_i^k \right\|_2^2\Big) \qquad (4.b)$$

$$y_i^{k+1} = y_i^k + \bar{x}_i^{k+1} - \bar{w}_i^{k+1} \qquad (4.c)$$

Fig. 2 and Fig. 3 show the sequential iteration relationship and flow chart of the two-level problem. In (2) and (4), where u and y are dual variables, $\bar{z}$ and $\bar{w}$ are intermediate optimization variables and ρ_u and ρ_l are upper-level/lower-level step size.

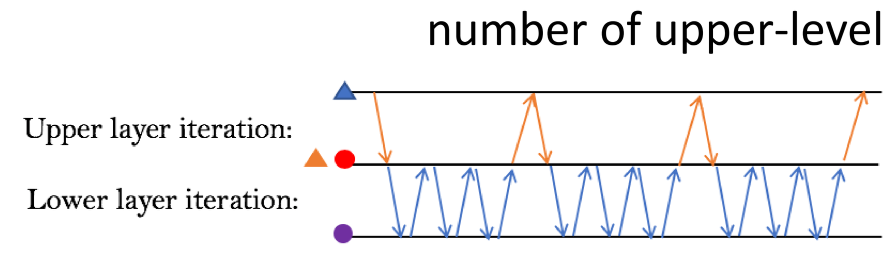

Fig. 2: Bi-level iterations for H-ADMM.

Case Study

Scenario setup: Considering three aggregators and each has two clusters to follow a load signal $\beta = [28,16,20,22,18,24,15,18]$ in an 8-step control horizon. Each aggregator has power limit of $\tau = [9,8,11]$. All unites above are MW. Two cluster models, which are unknown to utility, are considered:

Simple discomfort model: $f_{ij}(x_{ij}) = \sum_{t=1}^{T} \max(1/x_{ijt} - 1/a_{ij}, 0)$

Detailed discomfort model: $f_{ij}(x_{ij})$ is obtained from simulation, in which a certain optimal control is applied. In this case, $f_{ij}(x_{ij})$ does not have explicit form.

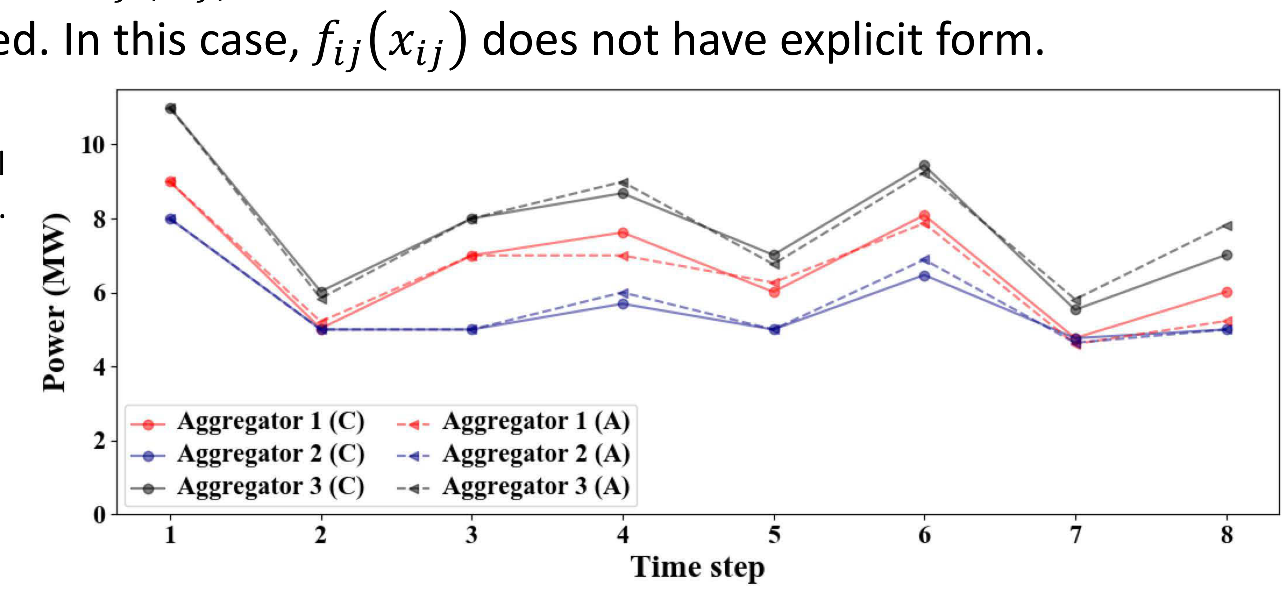

Fig. 4: Power consumption of all three aggregators in eight steps. Comparison between centralized (C) and H-ADMM (A). Results show the H-ADMM converges to the similar results yield by its centralized counterpart.

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 1: Cost comparison of three load allocation algorithms.

To prove H-ADMM efficacy, the equivalent centralized optimization (considering (1) and simple discomfort model, meaning utility knows cluster model) is solved, and the results are compared in Fig. 4 and Table 1. Fig. 5 (a) shows convergence with the number of upper-level iterations.

Both plots show the converging process of H-ADMM in two examples. However, due to detailed simulation, wall time convergence when using detailed model is slow. Future work can use a pre-trained surrogate discomfort model instead to accelerate the algorithm convergence.

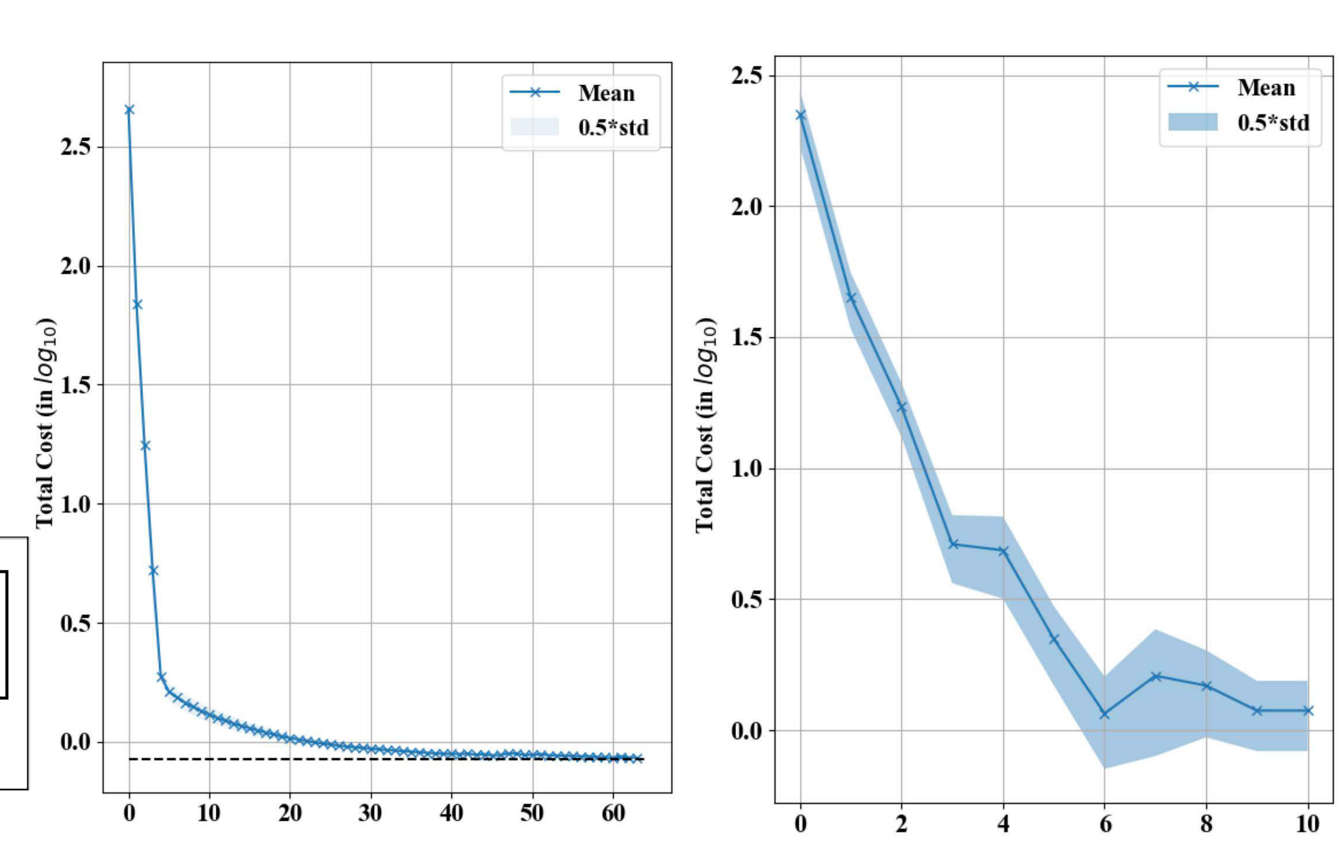

Fig. 5: Convergence curves for scenarios using two clusters model (simple (a. Left) and detailed (b.Right)).

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

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Innovative Smart Grid Technologies (ISGT 2020) North America
Washington, D.C.
February 17-20, 2020
NREL/PO-2C00-76083