LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-813257

Decentralized Failure-Tolerant Optimization of Electric Vehicle Charging

I. Aravena, S. Chapin, C. Ponce

August 3, 2020

IEEE Transactions on Smart Grid

Disclaimer

Decentralized Failure-Tolerant Optimization of Electric Vehicle Charging

Ignacio Aravena, *Member, IEEE,* Steve Chapin, and Colin Ponce

***Abstract*—We present a decentralized failure-tolerant algorithm for optimizing electric vehicle (EV) charging, using charging stations as computing agents. The algorithm is based on the alternating direction method of multipliers (ADMM) and it has the following features: (*i*) It handles capacity, peak demand, and ancillary services coupling constraints. (*ii*) It does not require a central agent collecting information and performing coordination (e.g. an aggregator), instead all agents exchange information and computations are carried out in a fully decentralized fashion. (*iii*) It can withstand the failure of any number of computing agents, as long as the remaining computing agents are in a connected communications network. We construct this algorithm by reformulating the optimal EV charging problem in a decomposable form, amenable to ADMM, and then developing efficient decentralized solution methods for the subproblems dealing with coupling constraints. We conduct numerical experiments on industry-scale synthetic EV charging datasets, with up to 1 152 charging stations, using a high performance computing cluster. The experiments demonstrate that the proposed algorithm can solve the optimal EV charging problem fast enough to permit the integration of EV charging with real-time electricity markets, even in the presence of failures.**

***Index Terms*—Electric vehicle charging, ADMM, decentralized algorithm, fail-proof algorithm**

I. Introduction

THE anticipated massive adoption of electric vehicles (EVs) will bring some of the biggest challenges and transformational opportunities for power systems operation. It will inevitably mean that current load patterns, for which transmission and distribution systems have been designed, will drastically change, requiring either new infrastructure or new operating paradigms to maintain the reliability of the power grid. For example, the average family, driving light-duty passenger EV, would see their electricity usage increase by almost 50%.[1] These changes in load are intensified by the progressive integration of intermittent resources in the generation mix.

EVs' characteristics distinguishing them from traditional loads. EVs can store energy in their batteries, bringing a form of distributed storage capacity to the power grid, and their instantaneous demand is very flexible, only requiring that net energy stored (the battery's State of Charge (SOC)) reaches a required level by a given time limit. This flexibility, if harnessed, could help maintain and improve the reliability of the grid, even in the presence of intermittent resources. However, harnessing EVs' flexibility is not straightforward because of the large number of devices that would need to be managed simultaneously in near real-time. Furthermore, in reality, imperfect communications infrastructure, disconnections, and other unexpected events increase the challenge of managing EV charging.

In this paper, we present a new algorithm for scheduling EV charging in near real-time, which simultaneously tackles the two aforementioned challenges in harnessing EV charging flexibility: scaling to real-world fleet and demand sizes, and tolerance to real-world failures.

A. Literature review

A significant amount of research in recent years has been devoted to the problem of coordinating EV chargers in a scalable fashion, from the perspectives of both scheduling and real-time control. Across these studies, the terms *distributed* and *decentralized* are often used interchangeably to characterize parallel algorithms. Following [3], we use these terms to refer to two different parallel algorithm classes:

Distributed algorithms have multiple *workers* communicate with a *central coordinator* to compute common quantities.

Decentralized algorithms have no *central coordinator*, and all computations are performed collectively by multiple *workers*.

Most existing approaches for EV charge coordination fall under the first category. In the following, we review a subset of them, representative of the most salient ideas in the area. Wen *et al.* [4], uses ADMM [5] to relax a mixed-integer optimal EV charging problem, where EVs communicate their intended consumption to an aggregator, which updates averages and dual multipliers, and broadcasts them back to the EVs. Joo and Ilić [6], proposes a distributed algorithm based on Lagrange relaxation to solve the more general optimal demand response problem. Gan *et al.* [7] develops a specialized trust-region distributed approach for finding the optimal valley-filling EV charging schedule. Vaya *et al.* [8] proposes a distributed ADMM approach for scheduling EV charging under uncertainty considering only physical EV constraints. Le Floch *et al.* [9] applies Nesterov's smoothing [10] to the optimal EV charging problem and uses gradient ascent and incremental stochastic gradient methods, implemented in a distributed fashion, to solve the dual problem. Ghavami *et al.* [11] proposes a distributed primal-dual subgradient method to

I. Aravena, S. Chapin and C. Ponce are with the Lawrence Livermore National Laboratory, Livermore, CA, 94550 USA. e-mail:{aravenasolis1, chapin8, ponce11}@llnl.gov.

[1]Based on 54 daily vehicle miles/household [1], a 4 mi/kWh energy usage in a typical modern EV, and an average household electricity consumption of 30 kWh [2].

find the optimal valley-filling EV charging schedule while considering thermal capacity limits of distribution lines within a transportation model. Grammatico [12] proposes a distributed game theoretic approach for EV charging control where a central controller gathers aggregate information of agents and broadcasts a pricing signal to all agents of each iteration. Wang *et al.* [13] presents a distributed implementation of the A* algorithm to optimally schedule renewable generation and EV charging, using a transportation model for the power grid. These approaches, as well as similar ones found in the literature, fail our scaling and resilience requirements because of the communications/computation bottleneck imposed by the central coordinator and the single point of failure (or attack) it introduces.

A handful of authors have proposed decentralized methods for optimal EV charging or smart grid optimization under different assumptions and diverse underlying ideas. Ma *et al.* [14] proposes a decentralized algorithm for scheduling EV charging based on non-cooperative games. The paper assumes the existence of an energy-only market, that prices are the same for all vehicles, and that they are a pre-specified function of the margin between demand and installed capacity. Adika and Wang [15] follows a similar approach, using a supplementary pricing strategy to influence the charging/discharging of batteries more directly. Zhang *et al.* [16] proposes a distributed Frank-Wolfe decomposition method for optimal valley-filling EV charging and a decentralized ADMM method for optimizing EV charging considering linearized and unbalanced distribution network constraints. The latter approach, which shares many aspects with the proximal message passing approach of Kraning *et al.* [17] for decentralized power grid scheduling, becomes decentralized as a consequence of the graph structure of power grid constraints, and therefore it cannot accomodate features that couple agents outside the power flow equations, such as ancillary services. Münsing *et al.* [18] uses the same underlying algorithm as [8] to optimize the operation of a microgrid, and makes the method decentralized by carrying out the dual variable update step, which was originally done by an aggregator, through a *blockchain* where each local update is recorded. Atallah *et al.* [19] formulates a mixed-integer optimal EV charging problem, where EVs are optimally assigned to neighboring stations instead of being fixed at a given station, and propose a decentralized heuristic based on non-cooperative games for finding high-quality solutions of the problem in actual operation.

Another significant aspect of the EV charging coordination problem is its integration with existing and envisioned electricity markets. In this regard, all the work reviewed so far considers exclusively energy-only markets, whereas optimal EV charge scheduling also has the potential to significantly contribute to ancillary services, particularly to regulation and spinning reserves [20]. With the aim of exploiting this potential, Lin *et al.* [21] proposes a formulation for optimal EV charging with frequency regulation as a vehicle-to-grid service. The paper takes advantage of the zero-energy nature of frequency regulation to obtain a compact formulation and proposes a distributed algorithm, similar to that of [7], for solving the resulting optimization problem in practice. Juul *et al.* [22] studies how to integrate EV storage while charging into CAISO's ancillary services markets and propose a convex model for optimal tracking of ancillary services activation signals while respecting EV charging constraints. Peng and Liang [23] proposes a decentralized control algorithm for active power compensation provided by EVs on distribution grids based on a task-swap mechanism among charging stations, where tasks are discrete active power compensation portions. The algorithm is experimentally shown to mitigate sudden (significant) load changes at the distribution system level.

Finally, it is worthwhile to mention that decentralized approaches based on ADMM and other methods are very common in large-scale optimization, particularly in *consensus* and *sharing* problems, among others [5]. For these classes of problems, decentralization is the consequence of either local coupling constraints (relating only pairs of neighboring agents) or simple global coupling constraints leading to trivial global updates such as averaging, which can be handled via decentralized reduction operations. Recent literature on the topic has focused on creating robust variants of the ADMM method. Li *et al.* [24] proposes a decentralized robust ADMM method for consensus problems, where misbehaving agents are detected based on statistical thresholds and temporally replaced in the computation with estimates from the misbehaving agents' neighbors. Münsing and Moura [25] analyzes different types of cyberattacks on distributed and decentralized ADMM methods, and proposes to use Hessian approximations – constructed from iterates – for detecting noise injection attacks via convexity verification. It is expected that future research in distributed and decentralized EV charging methods will incorporate these, and other, cybersecurity considerations, as they are a requisite for storage and EV management over public communication networks, such as the internet.

B. Contributions

Our main contribution is the design of a fully-decentralized algorithm for solving convex optimization problems with global coupling constraints. Specifically, our algorithm can solve problems of the form

$$\begin{aligned} \min_{x_i \in \mathcal{X}_i} \quad & \sum_{i=1}^{I} f_i(\boldsymbol{x}_i) + \sum_{k=1}^{K} \max_{j \in \mathcal{J}(k)} \left(\sum_{i \in \mathcal{I}^g(k)} g_{i,j}(\boldsymbol{x}_i) \right) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}^h(l)} h_{i,l}(\boldsymbol{x}_i) \leq B_l \quad l = 1, \ldots, L, \end{aligned}$$

where $\boldsymbol{x}_i$ is the variable block associated with agent i; $\mathcal{X}_i,\ i = 1, \ldots, I$ is a closed convex set; f_i, $g_{i,j}$ and $h_{i,l}$ are convex functions; $\mathcal{J}(k),\ k = 1, \ldots, K$ is a partition of $\{1, \ldots, I\}$; and $\mathcal{I}^g(k) \subseteq \{1, \ldots, I\},\ k = 1, \ldots, K$ and $\mathcal{I}^h(l) \subseteq \{1, \ldots, I\},\ l = 1, \ldots, L$. This problem structure is not particular to our formulation of the optimal EV charging problem. It arises in several variants of optimal EV charging, such as all the formulations in the references of the previous section, as well as other problems in smart grids, particularly those where ancillary services are considered [3], [18], [21].

The key features of the proposed algorithm as follows. (*i*) It avoids inefficiencies introduced when using a blockchain for

the aggregation step [18], large expansions of the constraint space (e.g. reformulating the constraints over a graph [24]), or re-formulating the problem as a consensus or exchange problem. The approach does not rely on the graph structure of the power flow constraints [16] to perform decentralized updates, making it able to handle ancillary services and enabling its applicability outside the power systems field. (*ii*) It is failure-tolerant, in the sense that if agent i fails, or it is removed from the system due to security/cybersecurity considerations, the algorithm will find the solution to the problem without agent i, as long as the underlying communication network between continuing agents remains connected. (*iii*) Our algorithm is based on ADMM and it is mathematically equivalent, in terms of iterations, to the distributed approach of [8]. Therefore it finds the exact solution of the optimal EV charging problem, as opposed to heuristics [19] or game-theoretic approaches converging in the limit with infinitely many agents [14], [15].

C. Notation and paper organization

Unless indicated otherwise, we use lowercase for optimization variables, uppercase for parameters and calligraphic symbols for sets. We denote vectors using boldface and use partial indexing, e.g. for $\boldsymbol{x} \in \mathbb{R}^{N\times M}$, with elements $x_{i,j}$, we denote $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,M})$.

The rest of the paper is organized as follows. Section II presents a stylized version of our optimal EV charging problem, focusing mostly on its coupling constraints. Section III presents a distributed ADMM approach for solving the optimal EV charging problem. Section IV derives our decentralized failure-tolerant ADMM approach. Section V presents numerical results of our decentralized algorithm on synthetic instances of the optimal EV charging problem. Finally, section VI concludes this paper and suggests directions for future research.

II. Optimal EV charging problem

We model the optimal EV charging problem from the perspective of charging station owners or operators, that is, we minimize the total electricity bill of all charging stations. We consider that each EV arrives at a certain (predefined) charging station, and it indicates how long it will be plugged-in and what should be its SOC by the time it unplugs. We consider only grid-to-vehicle power injections (V1G) for simplicity of our formulation. However, our algorithm does not make any assumptions based on V1G technology and it could already incorporate vehicle-to-grid injections (V2G).

Fig. 1 presents a schematic of the charging infrastructure considered in this work. Each *charging station* $s \in \mathcal{S}$ is behind a certain *meter* $m(s) \in \mathcal{M}$ (the point of connection with the distribution network), which in turn is connected to a certain *distribution feeder* $f \in \mathcal{F} : s \in \mathcal{S}(f)$. Charging station s buys electricity from the distribution network at the energy price at its corresponding meter, $\Pi^{\text{energy}}_{m(s)}$. The peak power drawn from the grid at meter m (by all stations in $\mathcal{S}(m)$) over the horizon $\mathcal{T} = \{1, \ldots, T\}$ is priced at $\Pi^{\text{peak}}_m > 0$ and billed to the charging stations in $\mathcal{S}(m)$. At the same time, each charging station s participates in a *reserve group* $g \in \mathcal{G} : s \in \mathcal{S}(g)$ which provides aggregated ancillary services to the grid. We focus only on spinning reserve, which is sold to the system by each reserve group at $\Pi^{\text{spin}}_{g,t}$ for intervals in $\{T^{\text{spin}} + 1, \ldots, T\}$, where T^{spin} is the number of intervals between the last spinning reserve market clearing and real-time operation. For intervals in $\{1, \ldots, T^{\text{spin}}\}$, the amount of spinning reserve to be provided has already been determined in previous runs of the scheduling process, and a penalty of $\Pi^{\text{fail}} > 0$ is billed to charging stations per unit of reserves they fail to provide.

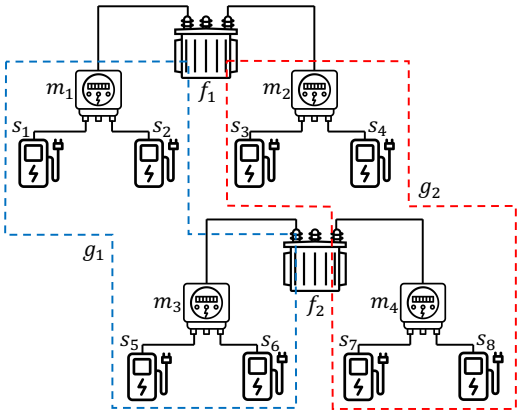

Fig. 1. Example topology of the charging infrastructure considered in this work[3]. Here $\mathcal{S}(m_1) = \{s_1, s_2\}$, $\mathcal{S}(f_1) = \{s_1, s_2, s_3, s_4\}$, $\mathcal{S}(g_1) = \{s_1, s_2, s_5, s_6\}$ (reserve group g_1 delimited with by dashed blue line), and so forth. Notably, stations in different feeders can be part of the same reserve group, extending coupling constraints beyond the limits of individual feeders.

Spinning reserves can be activated or not at each interval t of the horizon $\mathcal{T}$ and, when activated, charging operations in later intervals must be adjusted in order to meet the target SOC of each EV at the time it unplugs. We take a worst-case approach, ensuring that no matter the reserve activation signal, there exists, always, a feasible charging trajectory to meet the target SOC. We refer the reader to section EC.1 of the electronic companion [26] for the detailed formulation of constraints internal to charging stations, encompassing our EV models providing spinning reserves. In the following we focus on the coupling constraints between charging station, which is where our core contributions lie.

Let $p^{\mathbb{E}}_{s,t}, p^{\max}_{s,t}$ be the expected and maximum power draw, respectively, of station s at interval t, over all possible realizations of activation of spinning reserves; p^{peak}_m be the expected peak demand at meter m; $r^{\text{fail}}_{g,t}$ be promised but not delivered spinning reserve by group g at interval $t \leq T^{\text{spin}}$; and $r^{\text{spin}}_{s,t}$ be the spinning reserve provided by station s at interval t. Additionally, let $\boldsymbol{u} \in \mathbb{R}^{|\mathcal{S}|\times T}$, $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|\times T}$ and $\boldsymbol{w} \in \mathbb{R}^{|\mathcal{S}|\times T^{\text{spin}}}$ be clone variables of $\boldsymbol{p}^{\max}$, $\boldsymbol{p}^{\mathbb{E}}$ and $\boldsymbol{r}^{\text{spin}}$. Then, with all the previous considerations, the optimal EV charging problem can be formulated as (1) – (8).

[3]Station, meter and transformer illustrations sourced from *Noun Project*, https://thenounproject.com/.

$$\min_{\substack{p,r\\u,v,w\\\theta,\phi}} \sum_{s\in\mathcal{S}}\Big(\sum_{t\in\mathcal{T}}\Pi^{\text{energy}}_{m(s),t}\,p^{\mathbb{E}}_{s,t}-\sum_{t=T^{\text{spin}}+1}^{T}\Pi^{\text{spin}}_{g(s),t}\,r^{\text{spin}}_{s,t}\Big)+ \sum_{m\in\mathcal{M}}\Pi^{\text{peak}}_{m}\,\theta_m+\sum_{g\in\mathcal{G}}\sum_{t=1}^{T^{\text{spin}}}\Pi^{\text{fail}}\,\phi_{g,t} \tag{1}$$

$$\text{s.t.}\quad \sum_{s\in\mathcal{S}(f)} u_{s,t}\le P_{f,t}\quad \forall f\in\mathcal{F},t\in\mathcal{T} \tag{2}$$

$$\sum_{s\in\mathcal{S}(m)} v_{s,t}\le \theta_m\quad \forall m\in\mathcal{M},t\in\mathcal{T} \tag{3}$$

$$\sum_{s\in\mathcal{S}(g)} w_{s,t}\ge R^{\text{spin}}_{g,t}-\phi_{g,t}\quad \forall g\in\mathcal{G},t=1,\dots,T^{\text{spin}} \tag{4}$$

$$\boldsymbol{p}^{\max}_s=\boldsymbol{u}_s\quad \forall s\in\mathcal{S}\quad(\boldsymbol{\mu}_s) \tag{5}$$

$$\boldsymbol{p}^{\mathbb{E}}_s=\boldsymbol{v}_s\quad \forall s\in\mathcal{S}\quad(\boldsymbol{\nu}_s) \tag{6}$$

$$\boldsymbol{r}^{\text{spin}}_{s,\{1,\dots,T^{\text{spin}}\}}=\boldsymbol{w}_s\quad \forall s\in\mathcal{S}\quad(\boldsymbol{\xi}_s) \tag{7}$$

$$(\boldsymbol{p}^{\max}_s,\boldsymbol{p}^{\mathbb{E}}_s,\boldsymbol{r}^{\text{spin}}_s)\in\mathcal{D}_s\qquad \forall s\in\mathcal{S}, \tag{8}$$

The objective function (1) corresponds to the total cost of operating the charging stations, including energy charges, peak demand charges, spinning reserve penalties and revenues for spinning reserve provision. Constraint (2) ensures that total load from charging stations $s\in\mathcal{S}(f)$ does not surpass the net capacity (capacity minus inflexible demand) $P_{f,t}$ of the distribution feeder f, at all intervals $t\in\mathcal{T}$. Constraint (3) computes the expected peak demand at each meter m using an epigraph formulation for the max operator. Constraint (4) ensures that the spinning reserves promised by each reserve group are either provided or the corresponding penalty is charged to the group. Constraints (5) – (7), with dual variables $\boldsymbol{\mu}\in\mathbb{R}^{|\mathcal{S}|\times T}$, $\boldsymbol{\nu}\in\mathbb{R}^{|\mathcal{S}|\times T}$ and $\boldsymbol{\xi}\in\mathbb{R}^{|\mathcal{S}|\times T^{\text{spin}}}$, ensure that $\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}$ are clones of $\boldsymbol{p}^{\max},\boldsymbol{p}^{\mathbb{E}},\boldsymbol{r}^{\text{spin}}$. Constraint (8) enforces the internal constraints of charging stations, represented here by the polyhedral set $\mathcal{D}_s$ (defined formally in section EC.1 of [26]), on power draw and reserves.

III. Distributed decomposition algorithm

Following [5], [8], we can solve (1) – (8) with ADMM, using charging stations and one *aggregator* as computational units, as follows:

0) Let $k=0$, $\boldsymbol{p}^{\max,k}=\boldsymbol{p}^{\mathbb{E},k}=\boldsymbol{r}^{\text{spin},k}=\boldsymbol{u}^k=\boldsymbol{v}^k=\boldsymbol{\mu}^k=\boldsymbol{\nu}^k=\mathbf{0}$, and $\boldsymbol{w}^k=\boldsymbol{\xi}^k=\mathbf{0}$.
1) Each charging station $s\in\mathcal{S}$ updates its local variables (*x-update* in ADMM literature [5]):

$$\begin{aligned}(\boldsymbol{p}^{\max,k+1}_s,\boldsymbol{p}^{\mathbb{E},k+1}_s,\boldsymbol{r}^{\text{spin},k+1}_s)\;=\;&\\ \arg\min_{p,r}\;&\sum_{t\in\mathcal{T}}\Pi^{\text{energy}}_{m(s),t}\,p^{\mathbb{E}}_{s,t}-\sum_{t=T^{\text{spin}}+1}^{T}\Pi^{\text{spin}}_{g(s),t}\,r^{\text{spin}}_{s,t}+\\ &\langle\boldsymbol{\mu}^k_s-\rho\boldsymbol{u}^k_s,\boldsymbol{p}^{\max}_s\rangle+\frac{\rho}{2}\big\|\boldsymbol{p}^{\max}_s\big\|_2^2+\\ &\langle\boldsymbol{\nu}^k_s-\rho\boldsymbol{v}^k_s,\boldsymbol{p}^{\mathbb{E}}_s\rangle+\frac{\rho}{2}\big\|\boldsymbol{p}^{\mathbb{E}}_s\big\|_2^2+\\ &\langle\boldsymbol{\xi}^k_s-\rho\boldsymbol{w}^k_s,\boldsymbol{r}^{\text{spin}}_{s,\{1,\dots,T^{\text{spin}}\}}\rangle+\frac{\rho}{2}\big\|\boldsymbol{r}^{\text{spin}}_{s,\{1,\dots,T^{\text{spin}}\}}\big\|_2^2\\ \text{s.t.}\;\;&(\boldsymbol{p}^{\max}_s,\boldsymbol{p}^{\mathbb{E}}_s,\boldsymbol{r}^{\text{spin}}_s)\in\mathcal{D}_s\end{aligned} \tag{9}$$

2) Each charging station $s\in\mathcal{S}$ sends $\boldsymbol{p}^{\max,k+1}_s$, $\boldsymbol{p}^{\mathbb{E},k+1}_s$ and $\boldsymbol{r}^{\text{spin},k+1}_s$ to the aggregator.
3) The aggregator updates the clone variables (*z-update*):
   a) For each feeder $f\in\mathcal{F}$, interval $t\in\mathcal{T}$:

$$\begin{aligned}\boldsymbol{u}^{k+1}_{\mathcal{S}(f),t}=\;&\\ \arg\min_{u}\;&\sum_{s\in\mathcal{S}(f)}\Big(\big(-\mu^k_{s,t}-\rho\,p^{\max,k+1}_{s,t}\big)u_{s,t}+\frac{\rho}{2}u^2_{s,t}\Big)\\ \text{s.t.}\;\;&\sum_{s\in\mathcal{S}(f)}u_{s,t}\le P_{f,t}\qquad(\alpha_{f,t})\end{aligned} \tag{10}$$

   b) For each meter $m\in\mathcal{M}$:

$$\begin{aligned}(\boldsymbol{v}^{k+1}_{\mathcal{S}(m)},\cdot)=\;&\\ \arg\min_{v,\theta}\;&\Pi^{\text{peak}}_m\theta_m+\\ &\sum_{s\in\mathcal{S}(m)}\Big(\langle-\boldsymbol{\nu}^k_s-\rho\,\boldsymbol{p}^{\mathbb{E},k+1}_s,\boldsymbol{v}_s\rangle+\frac{\rho}{2}\|\boldsymbol{v}_s\|_2^2\Big)\\ \text{s.t.}\;\;&\sum_{s\in\mathcal{S}(m)}v_{s,t}\le\theta_m\quad\forall t\in\mathcal{T}\quad(\beta_{m,t})\end{aligned} \tag{11}$$

   c) For each reserve group $g\in\mathcal{G}$, interval $t\in\mathcal{T}$:

$$\begin{aligned}(\boldsymbol{w}^{k+1}_{\mathcal{S}(g),t},\cdot)=\;&\\ \arg\min_{w,\phi}\;&\Pi^{\text{fail}}\phi_{g,t}+\\ &\sum_{s\in\mathcal{S}(g)}\Big(\big(-\xi^k_{s,t}-\rho\,r^{\text{spin},k+1}_{s,t}\big)u_{s,t}+\frac{\rho}{2}w^2_{s,t}\Big)\\ \text{s.t.}\;\;&\sum_{s\in\mathcal{S}(g)}w_{s,t}\ge R^{\text{spin}}_{g,t}-\phi_{g,t}\qquad(\gamma_{g,t})\end{aligned} \tag{12}$$

4) Aggregator computes primal and dual residuals [5]:

$$\begin{aligned}\Delta^{\text{primal}}&=\Big(\|\boldsymbol{p}^{\max,k+1}-\boldsymbol{u}^{k+1}\|_2^2+\|\boldsymbol{p}^{\mathbb{E},k+1}-\boldsymbol{w}^{k+1}\|_2^2+\\ &\qquad\|\boldsymbol{r}^{\text{spin},k+1}_{1,\dots,T^{\text{spin}}}-\boldsymbol{w}^{k+1}\|_2^2\Big)^{1/2}\\ \Delta^{\text{dual}}&=\rho\Big(\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|_2^2+\|\boldsymbol{v}^{k+1}-\boldsymbol{v}^{k}\|_2^2+\\ &\qquad\|\boldsymbol{w}^{k+1}-\boldsymbol{w}^{k}\|_2^2\Big)^{1/2}\end{aligned} \tag{13}$$

   Terminate if target tolerances are achieved: $\Delta^{\text{primal}}\le\epsilon^{\text{primal}}$ and $\Delta^{\text{dual}}\le\epsilon^{\text{dual}}$.
5) Agregator sends updated clone variables $\boldsymbol{u}^{k+1}_s$, $\boldsymbol{v}^{k+1}_s$ and $\boldsymbol{w}^{k+1}_s$ to station s, for every $s\in\mathcal{S}$.
6) Each charging station $s\in\mathcal{S}$ update its dual variables, aggregator updates its copies of dual variables (*y-update*):

$$\begin{aligned}\boldsymbol{\mu}^{k+1}_s&=\boldsymbol{\mu}^k_s+\rho\big(\boldsymbol{p}^{\max,k+1}_s-\boldsymbol{u}^{k+1}_s\big)\\ \boldsymbol{\nu}^{k+1}_s&=\boldsymbol{\nu}^k_s+\rho\big(\boldsymbol{p}^{\mathbb{E},k+1}_s-\boldsymbol{v}^{k+1}_s\big)\\ \boldsymbol{\xi}^{k+1}_s&=\boldsymbol{\xi}^k_s+\rho\big(\boldsymbol{r}^{\text{spin},k+1}_s-\boldsymbol{w}^{k+1}_s\big)\end{aligned} \tag{14}$$

6) Let $k=k+1$ and return to 1.

In this implementation, a significant portion of the computation at each iteration, namely steps 0, 1 and 6 can be performed in parallel by the charging stations. However, steps 2–5 require a central aggregator, which needs updated information from each charging station in each iteration. This is especially true for step 3, which requires solving optimization problems using all the iterates computed in step 1.

The need for a central computation agent (*i*) creates a communication bottleneck (all charging stations require sending its iterates to the aggregator at each iteration), which poses scalability challenges for the approach, as well as (*ii*) introduces

a single point of failure for the entire computation process, which might be vulnerable to random errors or attacks. Both these effects are undesirable in practical implementations. We study how to remove these limitations in the next section.

IV. Decentralized failure-tolerant decomposition algorithm

Our approach to decentralize steps 2–5 of the algorithm in the previous section can be summarized as follows. We derive specialized solution methods for updating the clone variables (step 3) which depend only on aggregate quantities and local computation. These aggregates are sums of local quantities at each station and they are computed using decentralized (failure-tolerant) reduction operations. We use the same reduction operations to compute primal and dual residuals in a decentralized fashion, thereby making steps 2–5 fully decentralized. In this setting, the failure tolerance of our algorithm comes, in part, as a by-product of our use of reduction operations, as a failed station no longer contributes to the aggregates for updating clone variables, effectively disappearing from the optimization problem being solved.

In the following, first, we briefly describe the properties and implementation of decentralized reduction operations, then, we present our decentralized update methods for clone variables, and conclude with a full description of our decentralized scheduling algorithm for optimal EV charging.

A. Allreduce operations

An *allreduce* operation is a common communication collective in which every computing agent i in a group holds some data x_i, and the group collectively calculates

$$\overline{x} = \underset{i}{\otimes} x_i,$$

where $\otimes$ is any operation that is both commutative and associative, and all agents receive the answer. In our case, we use $\otimes = +$, as we need to sum decentralized pieces of data.

Allreduce has been well-studied in the past in the context of computations on reliable computer clusters, and a number of very efficient algorithms exist (see, e.g. [27]). Furthermore, many of these algorithms are decentralized; in the context of occasional failures, one of these methods combined with a simple check-and-retry procedure is likely ideal.

If greater resilience within the allreduce operation is required, more recent work has developed gossip protocol-based techniques for performing summation-allreduce with resilience against failures and data corruption [28], [29].

B. Decentralized solution of subproblems for updating clone variables z

We focus now on devising methods for solving the *capacity subproblem* (10), the *peak demand subproblem* (11) and the *ancillary services subproblem* (12), which can be implemented in a decentralized fashion using allreduce. Proofs for the propositions in this subsection are presented in section EC.2 of the electronic companion [26].

1) Decentralized solution of capacity subproblem: Consider a particular feeder $f \in \mathcal{F}$ and interval $t \in \mathcal{T}$. The capacity subproblem (10) associated with (f,t) can be interpreted as a biased projection of each individual station's maximum power draw $p_{s,t}^{\max,k+1}, s \in \mathcal{S}(f)$ onto the set admitted by the capacity of feeder $P_{f,t}$. Intuitively, if the stations' maximum power draw $\sum_{s\in\mathcal{S}(f)} p_{s,t}^{\max,k+1}$ exceeds the feeder's capacity $P_{f,t}$, then we should reduce the maximum draw of each station until the capacity limit is respected, which should manifest in the clone variables $u_{s,t}^{k+1}, s \in \mathcal{S}(f)$ since they always respect the capacity limit. This intuition is formalized in proposition 1.

Proposition 1. Let $\tilde{u}_{s,t} = p_{s,t}^{\max,k+1} + (1/\rho)\mu_{s,t}^k$ for all $s \in \mathcal{S}(f)$, $\chi_{f,t} = \sum_{s\in\mathcal{S}(f)} \tilde{u}_{s,t}$, and $\alpha_{f,t}^\dagger = (\rho/|\mathcal{S}(f)|)(\chi_{f,t} - P_{f,t})_+$. Then, $u_{s,t}^{k+1} = \tilde{u}_{s,t} - (1/\rho)\alpha_{f,t}^\dagger$ for all $s \in \mathcal{S}(f)$.

In words, this proposition tells us that the biased maximum power draw of each station is $\tilde{u}_{s,t}$. If the total biased maximum draw $\chi_{f,t}$ is smaller than the capacity $P_{f,t}$, then the clone variables take the value of the biased maximum power draw $\tilde{u}_{s,t}$. Otherwise, the clone variables will correspond to $\tilde{u}_{s,t}$ reduced by an adjustment factor that is equal for all stations under the feeder.

Note that the computation of $\tilde{u}_{s,t}$ is local at each station, so the computation of $\chi_{f,t}$ can be done via allreduce over the stations in $\mathcal{S}(f)$, and the computation of $u_{s,t}^{k+1}$ is local to each station (after $\chi_{f,t}$ has been computed and shared). Therefore, proposition 1 presents a decentralized approach for computing $\boldsymbol{u}^{k+1}$ at each iteration.

2) Decentralized solution of peak demand subproblem: Consider a particular meter $m \in \mathcal{M}$. The peak demand subproblem (11) aims at finding a slightly modified version of the expected power draw of each station $\boldsymbol{p}_s^{\mathbb{E},k+1}, s \in \mathcal{S}(m)$, adjusted by certain bias, that would decrease the peak demand penalization $\Pi_m^{\text{peak}}\theta_m$ of the meter. Intuition, in this case, indicates that that the result of solving this problem should result in a flattened version of the total expected draw $\sum_{s\in\mathcal{S}(m)} \boldsymbol{p}_s^{\mathbb{E},k+1}$, where intervals with the larger power draws will be decreased the most. This intuition can be verified using proposition 2.

Proposition 2. Let $\tilde{\boldsymbol{v}}_s = \boldsymbol{p}_s^{\mathbb{E},k+1} + (1/\rho)\boldsymbol{\nu}_s^k$ for all $s \in \mathcal{S}(m)$, $\boldsymbol{\psi}_m = \sum_{s\in\mathcal{S}(m)} \tilde{\boldsymbol{v}}_s$, and

$$\begin{aligned}(\boldsymbol{\beta}_m^\dagger, \cdot) = \ & \underset{\beta,\theta}{\arg\min}\ \theta_m \\ & \text{s.t. } \theta_m \mathbf{1}_{T\times 1} \geq \boldsymbol{\psi}_m - \frac{|\mathcal{S}(m)|}{\rho}\boldsymbol{\beta}_m \\ & \sum_{t\in\mathcal{T}} \beta_{m,t} = \Pi_m^{\text{peak}},\ \boldsymbol{\beta}_m \geq 0.\end{aligned} \tag{15}$$

Then, $\boldsymbol{v}_s^{k+1} = \tilde{\boldsymbol{v}}_s - (1/\rho)\boldsymbol{\beta}_m^\dagger$ for all $s \in \mathcal{S}(m)$.

Proposition 2 indicates that the biased expected peak demand of each station corresponds to $\tilde{\boldsymbol{v}}_s$, and it presents a linear program (LP) to compute the decrease vector $\boldsymbol{\beta}_m$ (which is guaranteed to be nonnegative). The proposition also presents a decentralized approach for computing $\boldsymbol{v}^{k+1}$: computing $\tilde{\boldsymbol{v}}_s$ locally, computing $\boldsymbol{\psi}_m$ via allreduce over the stations in $\mathcal{S}(m)$, and solving (15) and computing $\boldsymbol{v}_s^{k+1}$ locally at each station. While this decentralized approach requires

the repeated solution of problem (15) in all stations at each iteration, we note that this particular LP can be solved in super linear time, as indicated in proposition 3. We provide an algorithm achieving such performance in the proof of the proposition (section EC.2 of [26]).

Proposition 3. Problem (15) can be solved in, at most, $\mathcal{O}(T \log T)$ operations.

3) Decentralized solution of ancillary services subproblem: Consider a particular reserve group $g \in \mathcal{G}$ and interval $t \in \mathcal{T}$. The ancillary services subproblem (12) associated with (g,t) can be interpreted as a biased projection of individual station's reserve provision $r_{s,t}^{\text{spin},k+1}, s \in \mathcal{S}(g)$ onto the set admitted by the promised reserves of the group $R_{g,t}^{\text{spin}}$. In contrast to the capacity subproblem (10), the projection in this case is *soft*, in the sense that we are only projecting onto the set up to the point where it becomes more expensive than paying the penalty Π^{fail}. Proposition 4 presents an analogous result to that of proposition 1 for the capacity subproblem, allowing us to solve the ancillary services subproblem at each iteration in a decentralized fashion.

Proposition 4. Let $\tilde{w}_{s,t} = r_{s,t}^{\text{spin},k+1} + (1/\rho)\xi_{s,t}^k$ for all $s \in \mathcal{S}(g)$, $\omega_{g,t} = \sum_{s\in\mathcal{S}(g)} \tilde{w}_{s,t}$, and $\gamma_{g,t}^\dagger = \min\left\{(\rho/|\mathcal{S}(g)|)(R_{g,t}^{\text{spin}} - \omega_{g,t})_+, \Pi^{\text{fail}}\right\}$. Then, $w_{s,t}^{k+1} = \tilde{w}_{s,t} + (1/\rho)\gamma_{g,t}^\dagger$ for all $s \in \mathcal{S}(g)$.

C. Decentralized algorithm

Using the results of the previous subsection we propose a decentralized algorithm to solve problems (1) – (8) where each station $s \in \mathcal{S}$ performs the following operations:

0) Let $k = 0$, $\boldsymbol{p}_s^{\max,k} = \boldsymbol{p}_s^{\mathbb{E},k} = \boldsymbol{r}_s^{\text{spin},k} = \boldsymbol{u}_s^k = \boldsymbol{v}_s^k = \boldsymbol{\mu}_s^k = \boldsymbol{\nu}_s^k = \mathbf{0}$, and $\boldsymbol{w}_s^k = \boldsymbol{\xi}_s^k = \mathbf{0}$.
1) Compute $\boldsymbol{p}_s^{\max,k+1}, \boldsymbol{p}_s^{\mathbb{E},k+1}, \boldsymbol{r}_s^{\text{spin},k+1}$ by solving the *station subproblem* (9).
2) Compute $\boldsymbol{u}_s^{k+1}$:
   a) Compute $\tilde{\boldsymbol{u}}_s = \boldsymbol{p}_s^{\max,k+1} + (1/\rho)\boldsymbol{\mu}_s^k$.
   b) Compute $\boldsymbol{\chi}_f = \sum_{s\in\mathcal{S}(f)} \tilde{\boldsymbol{u}}_s$ and $|\mathcal{S}(f)|$ by allreduce over all stations in $\mathcal{S}(f)$.
   c) Compute $u_{s,t}^{k+1} = \tilde{u}_{s,t} - (1/|\mathcal{S}(f)|)(\chi_{f,t} - P_{f,t})_+$ for all $t \in \mathcal{T}$.
3) Compute $\boldsymbol{v}_s^{k+1}$:
   a) Compute $\tilde{\boldsymbol{v}}_s = \boldsymbol{p}_s^{\mathbb{E},k+1} + (1/\rho)\boldsymbol{\nu}_s^k$.
   b) Compute $\boldsymbol{\psi}_m = \sum_{s\in\mathcal{S}(m)} \tilde{\boldsymbol{v}}_s$ and $|\mathcal{S}(m)|$ by allreduce over all stations in $\mathcal{S}(m)$.
   c) Compute $\boldsymbol{\beta}_m^\dagger$ by solving problem (15).
   d) Compute $\boldsymbol{v}_s^{k+1} = \tilde{\boldsymbol{v}}_s - (1/\rho)\boldsymbol{\beta}_m^\dagger$.
4) Compute $\boldsymbol{w}_s^{k+1}$:
   a) Compute $\tilde{\boldsymbol{w}}_s = \boldsymbol{r}_{s,\{1,\ldots,T^{\text{spin}}\}}^{\text{spin},k+1} + (1/\rho)\boldsymbol{\xi}_s^k$.
   b) Compute $\boldsymbol{\omega}_g = \sum_{s\in\mathcal{S}(g)} \tilde{\boldsymbol{w}}_s$ and $|\mathcal{S}(g)|$ by allreduce over all station in $\mathcal{S}(g)$.
   c) Compute $w_{s,t}^{k+1} = \tilde{w}_{s,t} + \min\left\{(1/|\mathcal{S}(g)|)(R_{g,t}^{\text{spin}} - \omega_{g,t})_+, \Pi^{\text{fail}}/\rho\right\}$ for all $t \in \mathcal{T}$.
5) Compute primal and dual residuals:
   a) Compute $D_s^{\text{primal}} = \|\boldsymbol{p}_s^{\max,k+1} - \boldsymbol{u}_s^{k+1}\|_2^2 + \|\boldsymbol{p}_s^{\mathbb{E},k+1} - \boldsymbol{w}_s^{k+1}\|_2^2 + \|\boldsymbol{r}_{1,\ldots,T^{\text{spin}}}^{\text{spin},k+1} - \boldsymbol{w}_s^{k+1}\|_2^2$ and $D_s^{\text{dual}} = \|\boldsymbol{u}^{k+1} - \boldsymbol{u}^k\|_2^2 + \|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|_2^2 + \|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|_2^2$.
   b) Compute $D^{\text{primal}} = \sum_{s\in\mathcal{S}} D_s^{\text{primal}}$ and $D^{\text{dual}} = \sum_{s\in\mathcal{S}} D_s^{\text{dual}}$ by allreduce over all stations in $\mathcal{S}$.
   c) Compute $\Delta^{\text{primal}} = \sqrt{D^{\text{primal}}}$ and $\Delta^{\text{dual}} = \rho\sqrt{D^{\text{dual}}}$.
   d) If $\Delta^{\text{primal}} \le \epsilon^{\text{primal}}$ and $\Delta^{\text{dual}} \le \epsilon^{\text{dual}}$, terminate.
6) Compute $(\boldsymbol{\mu}_s^{k+1}, \boldsymbol{\nu}_s^{k+1}, \boldsymbol{\xi}_s^{k+1})$ as indicated in (14).
7) Let $k = k + 1$ and return to 1.

As anticipated, our algorithm does not require an aggregator collecting station information, and yet it is equivalent in terms of iterates to the algorithm of section III. Furthermore, our algorithm does not have any communication bottleneck, leading to better scalability in practical applications since reduction operations can be implemented over exponentially-growing communication trees (communication costs scale with the log of the number of stations).

Additionally, the proposed algorithm can tolerate any number of failures as long as the underlying communications network remains connected so that reductions can be performed. For example, we can consider step 2 and assume station $\tilde{s} \in \mathcal{S}(f)$ fails at iteration k. At iteration $k+1$, $\mathcal{S}(f)$ will no longer contain $\tilde{s}$. Step 2.a will be carried out as usual by all remaining stations in $\mathcal{S}(f)$. Step 2.b will not have the contribution of station $\tilde{s}$, which will be accounted for in the reduction and in the recomputation of $|\mathcal{S}(f)|$. Step 2.c will be carried out normally, and the iterates will move towards the optimal for the new problem, where $\mathcal{S}(f)$ does not contain $\tilde{s}$. The same reasoning applies to steps 3–5. Thus, this built-in failure-tolerance mechanism allows the solution process to continue in the event of station failures, whatever their nature, effectively hot-starting the algorithm for solving the modified problem.

V. Numerical experiments

We implemented the algorithm of section IV-C in C++, using Ipopt [30] compiled with HSL [31] for solving the station subproblem (9) and MPI [32] for handling communications in our tests. We conducted experiments on realistic-size synthetic instances using the Quartz cluster (2 × Intel Xeon E5-2695, 36 cores, 128 GB RAM per node), hosted at the Lawrence Livermore National Laboratory. In the following, we describe our process for generating instances, present experimental results on the performance of the proposed algorithm, and present experimental results on the algorithm's performance in the presence of failures.

A. Synthetic data generation

We generate synthetic instances with a symmetric topology: within every instance, each distribution feeder has the same number of meters under it, and each meter has the same number of charging stations under it. Every station has 4 EV plugs, with a maximum capacity of 7kW per plug, and a total capacity of 18kW. All instances have a 24-hour horizon, divided into 24 5-minute intervals (the next two hours), 24 15-minute intervals (the following 6 hours), and then 16 1-hour intervals to complete the 24-hour period. The remaining parameters of each instance are then randomly generated.

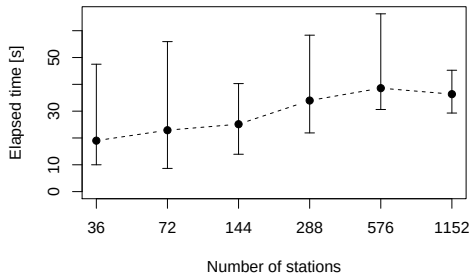

Fig. 2. Whiskers plot for solution times (wall clock) of our algorithm on all synthetic instances. Dots indicates median times over 10 instances for every charging station count.

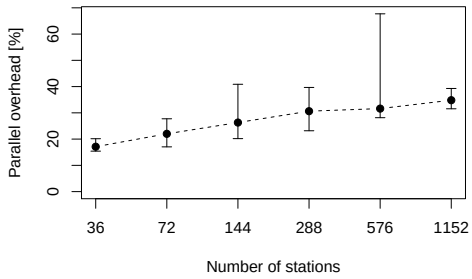

Fig. 3. Whiskers plot for parallel overhead of our algorithm on all synthetic instances. Dots indicates median overheads over 10 instances for every charging station count.

Stations are randomly assigned to reserve groups. EVs arrive randomly at each station, each with a random SOC between 15% and 45% of their total storage capacity, remain charging between 20 minutes and 8 hours, and must leave with a SOC randomly sampled between the arrival SOC and the storage capacity (capped at the maximum possible charge given by the plug capacity and the time that it remains plugged in). EVs' technical characteristics are uniformly sampled between 5kW and 7kW for input power, between 40kWh and 70kWh for storage capacity, and between 90% and 100% for charging efficiency. Prices for energy and reserve products are randomly generated within realistic bounds. Using these parameters, we create instances containing 36, 72, 144, 288, 576 and 1 152 charging stations, 10 instances per station count, for a total of 60 distinct instances.

B. Performance analysis

We solve all instances described in the previous sections with the following settings. Each charging station is assigned to a single core of LLNL's HPC cluster Quartz, emulating a practical deployment where each station is a computational unit. We set $\rho = 5$, $\epsilon^{\text{primal}} = \sqrt{|\mathcal{S}| \times (2T + T^{\text{spin}})} \times 50\text{mW}$ and $\epsilon^{\text{dual}} = \rho \times \epsilon^{\text{primal}}$.

Fig. 2 shows solution times for all our synthetic instances. It can be observed that solution times remain within the hard constraints imposed by real-time electricity market operations (5 minutes), even for instances with a large number of charging stations. The tendency for increasing solution times is explained, mainly, by two factors. (*i*) Longer solution times of certain charging station subproblems (9) due to the near singularity of the linear system to be solved at each interior point iteration (within Ipopt). This required us to hot-swap linear solvers (ma57 to ma27) to solve the subproblem and continue with the decentralized algorithm. These near singularity situations are more likely to occur with a larger number of subproblems to be solved at each step, hence the increasing time required to solve station subproblems as the number of charging stations increases. (*ii*) The algorithm has a logarithmically-increasing parallel overhead, as can be seen in Fig. 3, because of the use of communication structures similar to binary trees for allreduce operations.

The non-increasing number of iterations for convergence along with the major contributors to solution time increase indicate that—with a more robust solution approach for solving the charging station subproblem (9), such as using a specialized QP solver which could be built based on the OOQP package [33]—the total solution time of our algorithm should scale logarithmically with the number of charging stations, which would make our approach practical for deployment in real-world charging infrastructure.

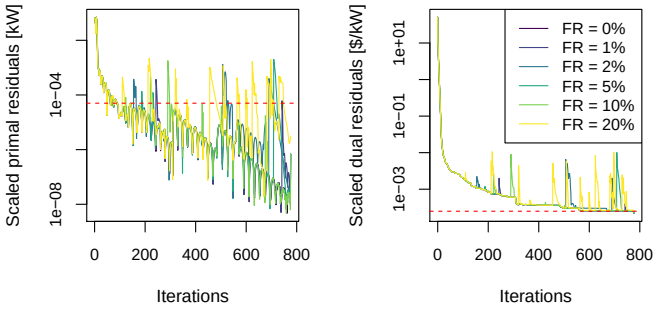

Fig. 4. Primal and dual residuals for synthetic a instance with 288 stations, while simulating random failures with different failure rates (FR). Residuals scaled by $\sqrt{|\mathcal{S}| \times (2T + T^{\text{spin}})}$. Tolerances indicated with a dashed red line.

C. Failure tolerance

We re-solve all 288-station instances while simulating stations going offline at different failure rates. We use the same solution parameters as in the previous section, except that we recompute the primal and dual tolerances after each failure, so that the optimal EV charging problem without the failed charging station is solved to the same relative accuracy as the original problem.

Fig. 4 presents the scaled primal and dual residuals for a representative subset of instances, all other present similar qualitative behaviour. Failures manifest as sharp increases in both primal and dual residuals, which our algorithms quickly reduces within the next couple of iterations after the failure. As a result we observe little to no increase in iteration count to solution, for failure rates of up to 20%. These results are also very encouraging, as they demonstrate that our algorithm can effectively withstand a large number of failures, taking advantage of the algorithm's partial solution before the failure to warm start the solution to the modified problem, indicating that our algorithm can work even under partial outages of the power grid.

VI. Conclusion

We have presented a fully decentralized algorithm for solving the optimal EV charging problem which can withstand any number of failures of charging stations. While we focus on the optimal EV charging problem, the ideas we presented can be applied to a larger class of problems, which may benefit from a decentralized approach to solving them.

We have numerically shown that the algorithm scales to industry-scale fleet sizes, respecting time limits for integration with real-time electricity markets, even with a high number of failures. These encouraging results can be further improved with more careful parameter setting (different ρ for different relaxed constraints), relaxed dual tolerances, and more robust methods for solving each charging station subproblem.

Further research will investigate methods for failure/corruption detection, efficient privacy-preserving extensions for optimizing system with potentially many charging station owners, efficient formulations for V2G technologies, and asynchronous decentralized extensions for faster convergence and communication delay/error tolerance.

Acknowledgment

The authors would like to thank Dr. Jonathan Donadee (Geli) and Dr. Amelia Musselman (LLNL) for formulating the initial version of the EV charging model, presented in section II and section EC.2 of [26], on which the current work is based. This work was funded by the U.S. Department of Energy (DOE) Vehicle Technologies Office under agreement 34828, and it was performed under the auspices of DOE by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] USDoT, "Summary of travel trends: 2017 national household travel survey," U.S. Department of Transportation, Federal Highway Administration, Jul. 2018.

[2] USEIA. (2019, Oct.) Frequently asked questions. U.S. Energy Information Administration. [Online]. Available: https://www.eia.gov/tools/faqs/faq.php?id=97&t=3

[3] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized dc optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2018.

[4] C. Wen, J. Chen, J. Teng, and P. Ting, "Decentralized plug-in electric vehicle charging selection algorithm in power systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1779–1789, 2012.

[5] 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*, vol. 3, no. 1, pp. 1–122, 2011.

[6] J. Joo and M. D. Ilić, "Multi-layered optimization of demand resources using lagrange dual decomposition," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2081–2088, 2013.

[7] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2013.

[8] M. G. Vayá, G. Andersson, and S. Boyd, "Decentralized control of plug-in electric vehicles under driving uncertainty," in *IEEE PES Innovative Smart Grid Technologies, Europe*, 2014, pp. 1–6.

[9] C. Le Floch, F. Belletti, S. Saxena, A. M. Bayen, and S. Moura, "Distributed optimal charging of electric vehicles for demand response and load shaping," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6570–6576.

[10] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[11] A. Ghavami, K. Kar, and A. Gupta, "Decentralized charging of plug-in electric vehicles with distribution feeder overload control," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3527–3532, 2016.

[12] S. Grammatico, "Exponentially convergent decentralized charging control for large populations of plug-in electric vehicles," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 5775–5780.

[13] L. Wang, S. Sharkh, and A. Chipperfield, "Optimal decentralized coordination of electric vehicles and renewable generators in a distribution network using a* search," *International Journal of Electrical Power & Energy Systems*, vol. 98, pp. 474 – 487, 2018.

[14] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 67–78, 2013.

[15] C. O. Adika and L. Wang, "Non-cooperative decentralized charging of homogeneous households' batteries in a smart grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1855–1863, 2014.

[16] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable electric vehicle charging protocols," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1451–1462, 2017.

[17] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Found. Trends Optim.*, vol. 1, no. 2, p. 73–126, Jan 2014.

[18] E. Münsing, J. Mather, and S. Moura, "Blockchains for decentralized optimization of energy resources in microgrid networks," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 2164–2171.

[19] R. F. Atallah, C. M. Assi, W. Fawaz, M. H. K. Tushar, and M. J. Khabbaz, "Optimal supercharge scheduling of electric vehicles: Centralized versus decentralized methods," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 7896–7909, 2018.

[20] P. Denholm, J. Eichman, T. Markel, and O. Ma, "Summary of market opportunities for electric vehicles and dispatchable load in electrolyzers," National Renewable Energy Laboratory (NREL), Tech. Report NREL/TP-6A20-64172, May 2015.

[21] J. Lin, K. Leung, and V. O. K. Li, "Optimal scheduling with vehicle-to-grid regulation service," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 556–569, 2014.

[22] F. Juul, M. Negrete-Pincetic, J. MacDonald, and D. Callaway, "Real-time scheduling of electric vehicles for ancillary services," in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–5.

[23] Z. Peng and L. Hao, "Decentralized coordination of electric vehicle charging stations for active power compensation," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.

[24] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust decentralized learning using admm with unreliable agents," 2017.

[25] E. Munsing and S. Moura, "Cybersecurity in distributed and fully-decentralized optimization: Distortions, noise injection, and admm," 2018.

[26] I. Aravena, S. Chapin, and C. Ponce, "Electronic Companion to: Decentralized Failure-Tolerant Optimization of Electric Vehicle Charging," 2020, available at:.

[27] R. Thakur, R. Rabenseifner, and W. Gropp, "Optimization of collective communication operations in mpich," *The International Journal of High Performance Computing Applications*, vol. 19, no. 1, pp. 49–66, 2005.

[28] G. Neiderbrucker and W. Gansterer, "Robust gossip-based aggregaton: A practical point of view," in *2013 Proceedings of the Meeting on Algorithm Engineering and Experiments (ALENEX)*, 2013, pp. 133–147.

[29] M. Casas, W. Gansterer, and E. Wimmer, "Resilient gossip-inspired all-reduce algorithms for high-performance computing: Potential, limitations, and open questions," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 366–383, 2018.

[30] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[31] HSL, "A collection of fortran codes for large scale scientific computation," 2015. [Online]. Available: http://www.hsl.rl.ac.uk/

[32] Message Passing Interface Forum, "MPI: a message-passing interface standard," June 2015, version 3.1. [Online]. Available: https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

[33] E. M. Gertz and S. J. Wright, "Object-oriented software for quadratic programming," *ACM Trans. Math. Softw.*, vol. 29, no. 1, p. 58–81, Mar. 2003. [Online]. Available: https://doi.org/10.1145/641876.641880

[34] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020. [Online]. Available: https://www.R-project.org/

Electronic Companion to: Decentralized Failure-Tolerant Optimization of Electric Vehicle Charging

Ignacio Aravena, *Member, IEEE,* Steve Chapin, and Colin Ponce

EC.1. EV CHARGING STATION CONSTRAINTS

This section presents the constraints defining the feasible operation domain of charging stations $\mathcal{D}_s, s \in \mathcal{S}$. Let $\mathcal{V}(s)$ be the set of EVs connected to station s at any interval in the horizon $\mathcal{T}$; $p_{v,t}, r^{\text{spin}}_{v,t}$ be, respectively, the power draw and spinning reserve provision of EV v at interval t; $r^{\text{recov}}_{s,t}$ be the power recovery for energy provided in activated spinning reserves of EV v at interval t (recovering energy of reserves activated at $\tau < t$) ; $\mathcal{D}_v$ be the feasible charge/discharge domain of EV $v \in \mathcal{V}$; and Λ be the probability of spinning reserve activation (identical for all stations and intervals). Then, we define domain $\mathcal{D}_s$ as:

$$\begin{aligned}\mathcal{D}_s = \Big\{ & (\boldsymbol{p}^{\text{max}}, \boldsymbol{p}^{\mathbb{E}}, \boldsymbol{r}^{\text{spin}}) \in \mathbb{R}^T \times \mathbb{R}^T \times \mathbb{R}^T \mid \\ & \exists (\boldsymbol{p}_v, \boldsymbol{r}^{\text{spin}}_v, \boldsymbol{r}^{\text{recov}}_v) \in \mathcal{D}_v \ \forall v \in \ \mathcal{V}(s) : \\ & p^{\text{max}}_t = \sum_{\substack{v \in \mathcal{V}(s): \\ \mathcal{T}(v) \ni t}} \big(p_{v,t} + r^{\text{recov}}_{v,t}\big), \\ & p^{\mathbb{E}}_t = \sum_{\substack{v \in \mathcal{V}(s): \\ \mathcal{T}(v) \ni t}} \big(p_{v,t} - \Lambda \cdot r^{\text{spin}}_{v,t} + \Lambda \cdot r^{\text{recov}}_{v,t}\big), \\ & r^{\text{spin}}_t = \sum_{\substack{v \in \mathcal{V}(s): \\ \mathcal{T}(v) \ni t}} r^{\text{spin}}_{v,t}, \ p^{\text{max}}_t \le P^{\text{max}}_s, \ \forall t \in \mathcal{T} \Big\}. \end{aligned} \tag{ec.1}$$

The first constraint in (ec.1) defines the maximum draw of station s as the sum of the draws and reserve recovery of all its connected EVs, which is indeed the worst case for each interval $t \in \mathcal{T}$: EVs draw their planned power and an additional amount to compensate for spinning reserves provided in previous intervals, without the reduction that would be caused by the activation of reserves in the current interval. The second constraint aggregates the expected draw of every EV at the station, considering the reserve activation probability Λ into the station expected draw, and the third constraint aggregates spinning reserve provisions in the same fashion. The forth constraint makes sure that the station's maximum power draw does not exceed the station's capacity.

We define the charge/discharge domain $\mathcal{D}_v$ of each EV $v \in \mathcal{V}$ as follows. Let $P^{\text{max}}_v, E^{\text{max}}_v, \eta_v$ be, respectively, the maximum power draw, energy storage capacity, and charging efficiency of v; $T^{\text{plug}}_v, T^{\text{unplug}}_v$ be the plug/unplug intervals of v and $\mathcal{T}(v) = \{T^{\text{plug}}_v, \dots, T^{\text{unplug}}_v\}$ be the set of intervals v is connected to its station; $E^{\text{plug}}_v, E^{\text{unplug}}_v$ be the SOC at T^{plug}_v and the target SOC at T^{unplug}_v; $q^{\text{recov}}_{t,\tau}$ be the recovery at interval t for reserve provided at interval $\tau < t$; and TR is the maximum allowed lag between reserve activation and recovery (e.g. if spinning reserve is activated at interval τ, all energy spent in the activation must be recovered by $\tau + TR$). Domain $\mathcal{D}_v$ is, then, defined as:

$$\begin{aligned}\mathcal{D}_v = \Big\{ & (\boldsymbol{p}, \boldsymbol{r}^{\text{spin}}, \boldsymbol{r}^{\text{recov}}) \in \mathbb{R}^{|\mathcal{T}(v)|} \times \mathbb{R}^{|\mathcal{T}(v)|}_+ \times \mathbb{R}^{|\mathcal{T}(v)|} \mid \\ & \exists \boldsymbol{q}^{\text{recov}}_t \in \mathbb{R}^{\min\{T^{\text{unplug}}_v - t, TR\}}_+ \ \ \forall t \in \mathcal{T}(v) \setminus T^{\text{unplug}}_v : \\ & 0 \le p_t - r^{\text{spin}}_t \ \ \forall t \in \mathcal{T}(v), \\ & p_t + \sum_{\tau = \max\{T^{\text{plug}}_v, t - TR\}}^{t-1} q^{\text{recov}}_{\tau,t} \le P^{\text{max}}_v \ \ \forall t \in \mathcal{T}(v), \\ & r^{\text{spin}}_t = \sum_{\tau = t+1}^{\min\{T^{\text{unplug}}_v, t + TR\}} q^{\text{recov}}_{t,\tau} \ \forall t \in \mathcal{T}(v), \\ & E^{\text{unplug}}_v \le E^{\text{plug}}_v + \sum_{t \in \mathcal{T}(v)} \eta_v \cdot p_t \le E^{\text{max}}_v \Big\}. \end{aligned} \tag{ec.2}$$

The first constraint in (ec.2) ensures that the power draw is always positive, even if spinning reserve is activated (we are only capturing existing V1G technology), while the second constraint ensures that the maximum power draw is not exceeded, even when recovering energy from previously activated reserves. The third constraint guarantees that if reserve is activated at a certain interval, the energy lost to reserves can be recovered in the coming TR intervals. The fourth constraint ensures that the target SOC is met and that the storage capacity is respected.

EC.2. PROOFS

Proof of Prop. 1. We will show that the pair $(\boldsymbol{u}^{k+1}_{\mathcal{S}(f),t}, \alpha^{\dagger}_{f,t})$, as defined in Prop. 1, is a primal-dual solution to the Karush-Kuhn-Tucker [1] (KKT) conditions of problem (10), which can be written as:

$$0 \le \alpha_{f,t} \perp \sum_{s \in \mathcal{S}(f)} u_{s,t} \le P_{f,t},$$

$$u_{s,t} = p^{\text{max},k+1}_{s,t} + \frac{1}{\rho}\mu^k_{s,t} - \frac{1}{\rho}\alpha_{f,t} \ \ \forall s \in \mathcal{S}(f),$$

I. Aravena, S. Chapin and C. Ponce are with the Lawrence Livermore National Laboratory, Livermore, CA, 94550 USA. e-mail:{aravenasolis1, chapin8, ponce11}@llnl.gov.

[1] Recall that Karush-Kuhn-Tucker conditions are necessary and sufficient conditions for optimality of convex programs with linear constraints.

and, using the definitions of Prop. 1, reduce to:

$$0 \leq \alpha_{f,t} \perp \frac{|\mathcal{S}(f)|}{\rho}\alpha_{f,t} \geq \chi_{f,t} - P_{f,t}.$$

There are two possible cases depending on the values of $\chi_{f,t}$:

- $\chi_{f,t} \leq P_{f,t}$. Then, $\alpha_{f,t} = 0, u_{s,t} = \tilde{u}_{s,t}$, for all $s \in \mathcal{S}(f)$, satisfies the KKT system.
- $\chi_{f,t} > P_{f,t}$. Then $\alpha_{f,t} = (\rho/|\mathcal{S}(f)|)(\chi_{f,t} - P_{f,t}), u_{s,t} = \tilde{u}_{s,t} - (1/N)(\chi_{f,t} - P_{f,t})$, for all $s \in \mathcal{S}(f)$, satisfies the KKT system.

Hence, $\alpha_{f,t} = \alpha^{\dagger}_{f,t} = (\rho/|\mathcal{S}(f)|)(\chi_{f,t} - P_{f,t})_+, u_{s,t} = u^{k+1}_{s,t} = \tilde{u}_{s,t} - (1/\rho)\alpha^{\dagger}_{f,t}$, for all $s \in \mathcal{S}(f)$, always satisfies the KKT system of problem (10). □

Proof of Prop. 2. We will show that problems (11) and (15) are equivalent. More precisely, we show that the space of optimal solutions on $(\theta_m, \boldsymbol{\beta}_m)$ of both problems coincide, and further, that $\boldsymbol{v}_{\mathcal{S}(m)}$ is implied by $\boldsymbol{\beta}_m$. The equivalence proof is based upon the KKT conditions of both problems, which we present in the following. The KKT conditions of problem (11) can be written as:

$$\begin{aligned} 0 \leq \boldsymbol{\beta}_m \perp \sum_{s\in\mathcal{S}(m)} \boldsymbol{v}_s \leq \theta_m,\ \mathbf{1}_{T\times 1}, \\ \boldsymbol{v}_s = \boldsymbol{p}^{\mathbb{E},k+1}_s + \frac{1}{\rho}\boldsymbol{\nu}^k_s - \frac{1}{\rho}\boldsymbol{\beta}_m \quad \forall s \in \mathcal{S}(m), \\ \sum_{t\in\mathcal{T}} \beta_{m,t} = \Pi^{\text{peak}}_m. \end{aligned}$$

It is clear, from the second condition, that the value of $\boldsymbol{v}_{\mathcal{S}(m)}$ is implied by $\boldsymbol{\beta}_m$. Substituting $\boldsymbol{v}_s$ from the second condition onto the first, we obtain the following reduced optimality conditions for problem (11):

$$0 \leq \boldsymbol{\beta}_m \perp \boldsymbol{\psi}_m - \frac{|\mathcal{S}(m)|}{\rho}\boldsymbol{\beta}_m \leq \theta_m\,\mathbf{1}_{T\times 1},\ \sum_{t\in\mathcal{T}} \beta_{m,t} = \Pi^{\text{peak}}_m. \tag{ec.3}$$

The KKT conditions of problem (15), on the other hand, can be written as:

$$\begin{aligned} 0 \leq \boldsymbol{\beta}_m \perp \eta\,\mathbf{1}_{T\times 1} \geq \frac{|\mathcal{S}(m)|}{\rho}\boldsymbol{\zeta}, \\ 0 \leq \boldsymbol{\zeta} \perp \boldsymbol{\psi}_m - \frac{|\mathcal{S}(m)|}{\rho}\boldsymbol{\beta}_m \leq \theta_m\,\mathbf{1}_{T\times 1}, \\ \sum_{t\in\mathcal{T}} \zeta_t = 1,\ \sum_{t\in\mathcal{T}} \beta_{m,t} = \Pi^{\text{peak}}_m, \end{aligned} \tag{ec.4}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^T$ and $\eta \in \mathbb{R}$ are dual variables of the linear constraints of (15).

Let $A = \{(\theta_m, \boldsymbol{\beta}_m) \in \mathbb{R}^{1+T} \mid \text{(ec.3)}\}$ and $B = \{(\theta_m, \boldsymbol{\beta}_m) \in \mathbb{R}^{1+T} \mid \exists(\eta, \boldsymbol{\zeta}) \in \mathbb{R}^{1+T} : \text{(ec.4)}\}$. We can show the following:

- $A \subseteq B$. To see this, let $(\bar{\theta}_m, \bar{\boldsymbol{\beta}}_m) \in A$, $\mathcal{T}^0 = \{t|\bar{\beta}_{m,t} = 0\}$, and $\mathcal{T}^1 = \{t|\bar{\beta}_{m,t} > 0\}$. Note that $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m) \in A \implies |\mathcal{T}^1| \geq 1$. Then, we can construct $\bar{\eta}$ and $\bar{\boldsymbol{\zeta}}$ such that $(\bar{\theta}_m, \bar{\boldsymbol{\beta}}_m, \bar{\eta}, \bar{\boldsymbol{\zeta}})$ solves (ec.4) as: $\bar{\eta} = |\mathcal{S}(m)|/(\rho|\mathcal{T}^1|)$, $\bar{\zeta}_t = 0\ \forall t \in \mathcal{T}^0$, $\bar{\zeta}_t = 1/|\mathcal{T}^1|\ \forall t \in \mathcal{T}^1$. Thus, $(\bar{\theta}_m, \bar{\boldsymbol{\beta}}_m) \in B$.
- $B \subseteq A$. Let $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m) \in B$. It is clear that $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m)$ satisfies all linear equalities and inequalities in (ec.3), as those are also present in (ec.4). It remains to show that $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m)$ satisfies the complementarity condition $\boldsymbol{\beta}_m \perp \boldsymbol{\psi}_m - (|\mathcal{S}(m)|/\rho)\boldsymbol{\beta}_m - \theta_m\,\mathbf{1}_{T\times 1}$. We will proceed by contradiction. Assume $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m)$ does not satisfy the aforementioned complementarity condition, thus there exists a $t \in \mathcal{T}$ for which $\bar{\beta}_{m,t} > 0$ and $\psi_{m,t} - (|\mathcal{S}(m)|/\rho)\bar{\beta}_{m,t} < \bar{\theta}_m$. However, $\eta \geq 1/\rho$, therefore, $\bar{\beta}_{m,t} > 0 \implies \eta = (|\mathcal{S}(m)|/\rho)\zeta_{m,t} \implies \zeta_{m,t} \geq 1/|\mathcal{S}(m)| \implies \psi_{m,t} - (|\mathcal{S}(m)|/\rho)\bar{\beta}_{m,t} = \bar{\theta}_m$, a contradiction. Thus, $(\bar{z}_m, \bar{\boldsymbol{\beta}}_m) \in A$.

Therefore, $A = B$, which concludes our proof. □

Proof of Prop. 3. The proof proceeds by presenting a greedy heuristic for problem (15) and, then, showing that it always produces an optimal solution. Let $z = \rho/(|\mathcal{S}(m)| \cdot \Pi^{\text{peak}}_m)\,\theta_m$, $\boldsymbol{D} = \rho/(|\mathcal{S}(m)| \cdot \Pi^{\text{peak}}_m)\,\boldsymbol{\psi}_m$, and $\boldsymbol{x} = 1/\Pi^{\text{peak}}_m\boldsymbol{\beta}_m$, then problem (15) can be rewritten as

$$\begin{aligned} \min_{z,x}\ & z \\ \text{s.t.}\ & z \geq D_t - x_t \quad \forall t \in \mathcal{T} \\ & \sum_{t\in\mathcal{T}} x_t = 1,\ x \geq 0, \end{aligned} \tag{ec.5}$$

for which any optimal solution must satisfy

$$0 \leq x_t \perp z + x_t \geq D_t\ \forall t \in \mathcal{T},\ \sum_{t\in\mathcal{T}} x_t = 1, \tag{ec.6}$$

due to Prop. 2. We construct a candidate solution $(z^H, \boldsymbol{x}^H)$ to (ec.5) through the following greedy heuristic:

1) Compute a decreasing order permutation $i(\cdot)$ for $\boldsymbol{D}$ ($D_{i(s)} \geq D_{i(t)}$ iff $i(s) \leq i(t)$, for $s, t \in \mathcal{T}$).
2) Let $k = 2$, $r = 1$ and $z^H = D_{i(1)}$.
3) If $k \leq T$ and $(k-1)(D_{i(k)} - D_{i(k-1)}) \leq r$ then:
   - Let $z^H = D_{i(k)}$, $r = r - (k-1)(D_{i(k-1)} - D_{i(k)})$.

   else:
   - Let $z^H = z^H - r/(k-1)$, $r = 0$.
4) If $r > 0$ then: let $k = k + 1$ and return to 3.
5) Let $x^H_t = \max\{0, D_t - z^H\}\ \forall t \in \mathcal{T}$.

We claim that $(z^H, \boldsymbol{x}^H)$ satisfies (ec.6) and, therefore, it is optimal for (ec.5). To see this, observe that inequality and complementarity constraints of (ec.6) are trivially satisfied by $(z^H, \boldsymbol{x}^H)$. It remains to show that $\sum_{t\in\mathcal{T}} x^H_t = 1$. Let k^H be the last k on which the condition of step 3 holds, then

$$z^H = D_{i(k^H)} - \frac{1}{k^H}\left(1 - \sum_{k=2}^{k^H}(k-1)(D_{i(k)} - D_{i(k-1)})\right),$$

thus

$$\begin{aligned} \sum_{t\in\mathcal{T}} x_t &= \sum_{k=1}^{k^H} D_{i(k)} - k^H z^H \\ &= \sum_{k=1}^{k^H}\big(D_{i(k)} - D_{i(k^H)}\big) - \\ &\qquad \sum_{k=2}^{k^H}(k-1)\big(D_{i(k-1)} - D_{i(k)}\big) + 1 = 1. \end{aligned}$$

Therefore, $(z^H, \boldsymbol{x}^H)$ is optimal for (ec.5). Optimal solutions for (15) can be found by simply scaling $(z^H, \boldsymbol{x}^H)$ as indicated in the beginning of this proof.

Finally, counting the worst-case number of operations of our optimal heuristic we have: $T \log T$ comparisons for sorting in step 1; $2T$ assignments in steps 3 and 4; and T assignments in step 5. Dropping linear terms, problem (15) worst-case time complexity is, at most, $\mathcal{O}(T \log T)$. □

Proof of Prop. 4. Omitted for brevity. Mild extension to proof of Prop. 1. □

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.