A Generalized Frank-Wolfe Approach to Decentralized Electric Vehicle Charging

Liang Zhang, Vassilis Kekatos, and Georgios B. Giannakis

Abstract— **The uncoordinated charging of large electric vehicle (EV) fleets could have an adverse influence on power network operation. To guarantee the secure and economic operation of power grids, vehicle charging needs to be coordinated to minimize the power supply cost, while catering vehicle charging requests. Provisioning large-scale fleets scheduled at fine timescales, the task of EV scheduling is tackled here by properly adopting the Frank-Wolfe method. Upon devising an optimal step size rule, the novel scheme is shown to enjoy fast convergence rate, especially during the first iterations. The derived charging protocol features affordable computational requirements from vehicle controllers and minimal information exchange between vehicles and their aggregator. To cope with random cyber delays in the communication links between vehicles and the aggregator, an asynchronous version of the charging scheme is also studied. Interpreted as a block stochastic Frank Wolfe algorithm, the latter ensures feasibility across iterations, converges in the mean, and enjoys the same order of convergence rate attained by its synchronous counterpart. Numerical tests demonstrate the advantage of our deterministic scheme over a state-of-the-art projected gradient descent alternative, as well as the robustness of its stochastic counterpart to asynchronous updates.**

I. Introduction

Electric vehicles (EVs) receive increasing attention as an effective means to reduce greenhouse gas emission and mitigate oil dependency. Nonetheless, charging large numbers of EVs will greatly affect the overall load profile. Without proper coordination scheme, charging of even a penetration 10% of EV loads will cause voltage magnitude drop and unacceptable load peaks [1]. On the other hand, vehicle loads can be controlled to minimize charging costs or provide auxillary sevices leveraging power electronics.

Charging protocols for electric vehicles has been an active area of current research; see [2] for a review. Compared to centralized control, decentralized control strategies enjoy computational efficiency and enhance user privacy. A heuristic decentralized EV scheduling mechanism based on congestion pricing used in Internet Protocol (IP) networks is proposed in [3]. A game theoretic approach is devised in [4], where a Nash equilibrium point is proved to exist presuming the unrealistic scenario that all vehicles have identical charging requests and plug-in/-out times. Leveraging the Lagrange relaxation method, iterative optimal decentralized schemes are proposed in [5] and [6]. Coordination of vehicles is achieved through distribution locational marginal prices in [7]. Reference [8] shows that a feasible valley-filling charging profile is optimal for any convex charging cost, and it further proposed a decentralized protocol that can be interpreted as a projected gradient descent (PGD). An ant-based swarm optimization algorithm is used in [9], while [10] suggests a multi-agent system. The vehicle charging problem is solved in a decentralized manner via the alternating direction method of multipliers (ADMM) in [11]. To tackle the spatial coupling introduced by transformer capacity limitations, a combination of ADMM and PGD has been reported in [12]. A real-time decentralized charging method based on dual decomposition and projected subgradient is developed in [13]; nevertheless, vehicle charging requirements are neglected. Considering unpredictable load and vehicle plug-in times, an online decentralized charging scheme is devised in [14]; its asymptotic performance is analyzed under the presumption that the EV charging requirements can be automatically satisfied. Resorting to the water filling scheme, a joint optimal power flow and EV management problem is solved in [15].

The contribution of this work is two-fold. First, a decentralized charging protocol is developed based on the Frank-Wolf method. The devised scheme enjoys fast convergence, especially during the first few iterations. Its closed-form updates pose minimal computational requirements for the vehicle controllers, and the overall computational time is significantly reduced. In particular, our numerical tests demonstrate a 100-times speed-up advantage over existing alternatives. Second, to cope with cyber failures in the communication link between the aggregator and the vehicle controllers, an asynchronous variant of the charging scheme is studied. By judiciously modifying its step size, the asynchronous scheme is shown to converge in the mean at the same $\mathcal{O}(\frac{1}{k})$ rate attained by its synchronous counterpart. This work complements [16], where synchronous charging protocols complying with distribution grid constraints were built on the plain Frank-Wolfe scheme.

The remainder of this paper is organized as follows: Section II introduces the Frank-Wolfe method for solving general convex quadratic programs. Section III formulates the optimal EV charging control problem. The optimal charging scheme and its asynchronous counterpart are detailed in Section IV. Section V shows simulation results and the paper in concluded in Section VI.

Notation. Lower- (upper-) case boldface letters are reserved for column vectors (matrices). Calligraphic symbols

Work in this paper was supported by NSF grants 1423316, 1442686, 1508993, and 1509040.

L. Zhang and G. B. Giannakis are with the Digital Technology Center and the ECE Dept., University of Minnesota, Minneapolis, MN 55455, USA. Emails: {zhan3523,georgios}@umn.edu

V. Kekatos is with the Bradley Dept. of ECE, Virginia Tech, Blacksburg, VA 24061, USA. Email: kekatos@vt.edu

denote sets. Symbol $^\top$ is used to represent vector and matrix transposition. Vectors $\mathbf{0}$ and $\mathbf{1}$ stand for the all-zeros and all-ones vectors, respectively.

II. The Frank-Wolfe Algorithm

We first apply the Frank-Wolfe algorithm for solving a general convex quadratic optimization problem. Define the function $f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top\mathbf{A}\mathbf{x} + \mathbf{b}^\top\mathbf{x}$ for a symmetric positive semi-definite matrix $\mathbf{A}$. Consider then the quadratic minimization problem

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} \quad f(\mathbf{x}) \quad \text{s. to} \quad \mathbf{x} \in \mathcal{X} \tag{1}$$

over the convex and compact feasible set $\mathcal{X}$. The Frank-Wolfe method, also known as conditional gradient algorithm, selects an initial feasible vector $\mathbf{x}^0 \in \mathcal{X}$, and iterates over k between the next two updates [17]

$$\mathbf{s}^k \in \arg\min_{\mathbf{s}\in\mathcal{X}} \ \mathbf{s}^\top \nabla f(\mathbf{x}^k) \tag{2a}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_k(\mathbf{s}^k - \mathbf{x}^k) \tag{2b}$$

for a step size $\gamma_k \in (0,1]$. By selecting $\gamma_k = \frac{2}{k+2}$, a convergence rate of $\mathcal{O}\left(\frac{1}{k}\right)$ can be achieved; see e.g., [17]. Nevertheless, if the γ_k's are selected optimally, a faster convergence rate can be attained, especially in the first few iterates. To facilitate the analysis, let us define the surrogate duality gap as [17]

$$g(\mathbf{x}) := \max_{\mathbf{s}\in\mathcal{X}} \ (\mathbf{x} - \mathbf{s})^\top \nabla f(\mathbf{x}). \tag{3}$$

Evaluating the latter at $\mathbf{x}^k$ and due to (2a), yields

$$g(\mathbf{x}^k) = -(\mathbf{v}^k)^\top \nabla f(\mathbf{x}^k) \tag{4}$$

where $\mathbf{v}^k := \mathbf{s}^k - \mathbf{x}^k$. From the convexity of $f(\mathbf{x})$, it holds that $g(\mathbf{x}) \geq f(\mathbf{x}) - f(\mathbf{x}^*) \geq 0$; see [17] for details. For the problem in (1), the per-iteration optimal step size γ_k can be found in closed form as asserted next.

Lemma 1. *The optimal step size γ_k in* (2b) *for solving* (1) *is*

$$\gamma_k = \min\left\{\frac{g(\mathbf{x}^k)}{(\mathbf{v}^k)^\top\mathbf{A}\mathbf{v}^k}, 1\right\}. \tag{5}$$

Proof. Performing line-search for γ_k entails tackling the univariate quadratic problem

$$\gamma_k := \arg\min_{\gamma\in[0,1]} \tfrac{1}{2}(\mathbf{x}^k+\gamma\mathbf{v}^k)^\top\mathbf{A}(\mathbf{x}^k+\gamma\mathbf{v}^k)+\mathbf{b}^\top(\mathbf{x}^k+\gamma\mathbf{v}^k)$$

whose solution is obtained by projecting the unconstrained solution $\frac{-\mathbf{v}_k^\top(\mathbf{A}\mathbf{x}^k+\mathbf{b})}{(\mathbf{v}^k)^\top\mathbf{A}\mathbf{v}^k} = \frac{g(\mathbf{x}^k)}{(\mathbf{v}^k)^\top\mathbf{A}\mathbf{v}^k}$ onto $[0,1]$. The solution in (5) is obtained upon noticing that $\frac{g(\mathbf{x}^k)}{(\mathbf{v}^k)^\top\mathbf{A}\mathbf{v}^k} \geq 0$. □

If γ_k is selected according to (5), the optimality gap is characterized as follows.

Proposition 1. *Let $h(\mathbf{x}^k) := f(\mathbf{x}^k) - f(\mathbf{x}^*)$. If γ_k is set according to* (5)*, it holds that*

$$h(\mathbf{x}^{k+1}) \leq \left(1 - \frac{\gamma_k}{2}\right) h(\mathbf{x}^k). \tag{6}$$

Proof. To simplify notation, denote $f(\mathbf{x}^{k+1})$, $h(\mathbf{x}^{k+1})$, $\mathbf{x}^{k+1}$ as f^+, h^+, $\mathbf{x}^+$; and $f(\mathbf{x}^k)$, $h(\mathbf{x}^k)$, $\mathbf{x}^k$, $\mathbf{v}^k$, γ_k as $f(\mathbf{x})$, $h(\mathbf{x})$, $\mathbf{x}$, $\mathbf{v}$, γ, respectively. It holds that

$$\begin{aligned} f^+ &= f + \frac{1}{2}\gamma^2\mathbf{v}^\top\mathbf{A}\mathbf{v} + \gamma\mathbf{v}^\top(\mathbf{A}\mathbf{x}+\mathbf{b}) \\ &= f + \frac{1}{2}\gamma\mathbf{v}^\top(\mathbf{A}\mathbf{x}+\mathbf{b}) + \frac{1}{2}\gamma\left[\gamma\mathbf{v}^\top\mathbf{A}\mathbf{v} + \mathbf{v}^\top(\mathbf{A}\mathbf{x}+\mathbf{b})\right] \\ &= f - \frac{1}{2}\gamma g(\mathbf{x}) + \frac{1}{2}\gamma\left[\gamma\mathbf{v}^\top\mathbf{A}\mathbf{v} - g(\mathbf{x})\right] \\ &\leq f - \frac{\gamma}{2}g(\mathbf{x}) \end{aligned} \tag{7}$$

since $\gamma \leq \frac{g(\mathbf{x})}{\mathbf{v}^\top\mathbf{A}\mathbf{v}}$ from (5). Given that $-g(\mathbf{x}) \leq -h(\mathbf{x})$, subtracting f^* from both sides of the inequality in (7) yields

$$h^+ \leq \left(1 - \frac{\gamma}{2}\right) h$$

which proves the claim. □

Proposition 1 not only implies the convergence of (2); it further shows that the optimality gap is non-increasing. For large γ_k, an exponential decay can be expected. During the first iterations, $h(\mathbf{x}^k)$ are $g(\mathbf{x}^k)$ are expected to be large, thus resulting in large γ_k's [cf. (5)]. This justifies our experimental observations that the Frank-Wolfe scheme based on line search converges faster during the initial iterates. A related analysis but for the so termed generalized Lagrangian is provided in [18].

The Frank-Wolfe algorithm features a disciplined stopping criterion: Because $g(\mathbf{x}^k) \geq f(\mathbf{x}^k) - f(\mathbf{x}^*)$, a valid stopping criterion could be selected as

$$\frac{g(\mathbf{x}^k)}{|f(\mathbf{x}^k)|} \leq \epsilon \tag{8}$$

where ϵ is the desired accuracy. Upon obtaining $\mathbf{s}^k$ from (2a), the calculation of $g(\mathbf{x}^k)$ via (4) is straightforward.

III. Vehicle Charging Control Problem

Suppose an aggregator wants to charge a fleet of N EVs over a period of T consecutive time slots comprising the set $\mathcal{T} := \{t : t = 1, \ldots, T\}$. The charging rate for vehicle n at slot t is denoted by $p_n(t)$, and it can lie within $[0, \bar{p}_n(t)]$. Because a vehicle can be charged only when it is connected to the grid; if $\mathcal{T}_n \subseteq \mathcal{T}$ is the connection interval for vehicle n, then $\bar{p}_n(t) = 0$ for $t \notin \mathcal{T}_n$, or

$$\bar{p}_n(t) = \begin{cases} \bar{p}_n & , \ t \in \mathcal{T}_n \\ 0 & , \ \text{otherwise} \end{cases} \quad \forall t \in \mathcal{T}$$

where $\bar{p}_n$ is its maximum charging rate determined by the battery specifications. By the end of the horizon $\mathcal{T}$, the total energy needed by EV n is represented by B_n. The latter depends on the initial and desired state of charge as well as the battery efficiency. The charging profile for vehicle n denoted by $\mathbf{p}_n := [p_n(1) \ \cdots \ p_n(T)]^\top$ should lie in the set

$$\mathcal{P}_n := \left\{\mathbf{p}_n : \mathbf{p}_n^\top\mathbf{1} = B_n, \ 0 \leq p_n(t) \leq \bar{p}_n(t) \ \forall t \in \mathcal{T}\right\} \tag{9}$$

which is convex and compact.

To minimize its electricity cost, the aggregator aims at solving the optimal EV charging problem [8]

$$\begin{aligned} \mathbf{p}^* \in \arg\min_{\mathbf{p}} \quad & F(\mathbf{p}) \\ \text{s. to} \quad & \mathbf{p}_n \in \mathcal{P}_n, \ \forall\, n \in \mathcal{N} \end{aligned} \tag{10}$$

where $\mathbf{p} := [\mathbf{p}_1^\top \ \cdots \ \mathbf{p}_N^\top]^\top$, and the total cost is defined as

$$F(\mathbf{p}) := \sum_{t=1}^{T} C\left(d(t) + \sum_{n=1}^{N} p_n(t)\right) \tag{11}$$

where $\{d(t)\}_{t=1}^T$ models any inelastic base load assumed to be known in advance, and $C(\cdot)$ is the electricity cost for the aggregator that is assumed fixed across time t. Observe that due to the summation $\sum_{n=1}^N p_n(t)$ appearing in the argument of $C(\cdot)$, the objective function $F(\mathbf{p})$ is not strongly convex in $\mathbf{p}$, even when $C(\cdot)$ is quadratic. The feasible set for problem (10) is the Cartesian product $\mathcal{P} := \mathcal{P}_1 \times \ldots \times \mathcal{P}_N$, and as such it is convex and compact too. Therefore, problem (10) is convex.

Assuming the electricity cost to be a quadratic function $H(\mathbf{p}) := \sum_{t=1}^T \left(d(t) + \sum_{n=1}^N p_n(t)\right)^2$, the ensuing instance of problem (10) is obtained

$$\begin{aligned} \min_{\mathbf{p}} \quad & H(\mathbf{p}) \\ \text{s. to} \quad & \mathbf{p}_n \in \mathcal{P}_n, \ \forall\, n \in \mathcal{N} \end{aligned} \tag{12}$$

Interestingly, the minimizers of (12) are also minimizers of (10). The ensuing proposition generalizes the result in [8] which confined $C(\cdot)$ to be strictly convex; thus excluding linear costs.

Proposition 2. *If $\mathbf{p}^*$ is a minimizer of (12), then $\mathbf{p}^*$ is a minimizer of (10) with respect to any convex differentiable function $C(\cdot)$.*

Proof. If $\mathbf{p}^*$ is a minimizer of (12), then the first order optimality condition imply that

$$\sum_{n=1}^{N} \langle \mathbf{d} + \sum_{n=1}^{N} \mathbf{p}_n^*,\ \mathbf{p}_n - \mathbf{p}_n^* \rangle \geq 0,\ \forall \mathbf{p}_n \in \mathcal{P}_n,\ \forall n \in \mathcal{N}$$

where $\mathbf{d} := [d(1) \ \cdots \ d(T)]^\top$. Equivalently, it holds that

$$\sum_{n=1}^{N} \langle \mathbf{d} + \sum_{n=1}^{N} \mathbf{p}_n^*,\ \mathbf{p}_n \rangle \geq \sum_{n=1}^{N} \langle \mathbf{d} + \sum_{n=1}^{N} \mathbf{p}_n^*,\ \mathbf{p}_n^* \rangle.$$

The latter implies that $\mathbf{p}^*$ is also the minimizer of

$$\begin{aligned} \mathbf{p}^* \in \arg\min_{\mathbf{p}} \quad & \sum_{n=1}^{N} \langle \mathbf{d} + \sum_{n=1}^{N} \mathbf{p}_n^*,\ \mathbf{p}_n \rangle \\ \text{s. to} \quad & \mathbf{p}_n \in \mathcal{P}_n, \ \forall\, n \in \mathcal{N} \end{aligned} \tag{13}$$

Given that the linear program in (13) is separable across vehicles, the $\mathbf{p}_n^*$'s are equivalently the minimizers of

$$\begin{aligned} \mathbf{p}_n^* \in \arg\min_{\mathbf{p}_n} \quad & \langle \mathbf{d} + \sum_{n=1}^{N} \mathbf{p}_n^*,\ \mathbf{p}_n \rangle \\ \text{s. to} \quad & \mathbf{p}_n \in \mathcal{P}_n \end{aligned} \tag{14}$$

for all $n \in \mathcal{N}$. Define the vector of the cost gradients across all time slots as

$$\mathbf{g}_C(\mathbf{p}) := [\dot{C}\big(d(1) + \sum_{n=1}^{N} p_n(1)\big) \ \cdots \ \dot{C}\big(d(T) + \sum_{n=1}^{N} p_n(T)\big)]^\top.$$

Recall that function $\dot{C}(\cdot)$ is increasing for any convex $C(\cdot)$. Therefore, the ordering of the entries in $\mathbf{g}_C(\mathbf{p}^*)$ coincides with the ordering of the entries in $\mathbf{d} + \sum_{n=1}^N \mathbf{p}_n^*$. The latter implies that the $\mathbf{p}_n^*$'s minimizing (14) are also minimizers of

$$\mathbf{p}_n^* \in \arg\min_{\mathbf{p}_n \in \mathcal{P}_n} \langle \mathbf{g}_C(\mathbf{p}^*), \mathbf{p}_n \rangle$$

for all n. The latter is equivalent to the inequalities:

$$\langle \mathbf{g}_C(\mathbf{p}^*),\ \mathbf{p}_n - \mathbf{p}_n^* \rangle \geq 0 \quad \forall\, n \in \mathcal{N}.$$

Since all $\mathbf{p}_n$'s are feasible and satisfy the latter inequality, they are also minimizers for problem (10) for any convex differentiable objective $C(\cdot)$. $\square$

Proposition 2 enables us to tackle a general class of vehicle charging problems: Solving (10) is rendered equivalent to solving its quadratic counterpart in (12). The latter can be efficiently solved using the Frank-Wolfe iterates of (2) as delineated next. Note that vehicle charging problems with a more general time-varying cost can still be talcked by the plain Frank-Wolfe scheme [16]; nonetheless, the superior convergence property of (6) no longer holds.

IV. Optimal Charging Schedulers

Since $H(\mathbf{p})$ is convex differentiable and set $\mathcal{P}$ is convex and compact, the Frank-Wolfe scheme of Section II is adopted here for tackling the charging problem in (12).

A. Optimal Decentralized Charging Control

At iteration k, the gradient $\mathbf{g}^k := \nabla H(\mathbf{p}^k)$ is evaluated, and the optimization problem in (2a) is subsequently solved. The latter corresponds to the linear program

$$\begin{aligned} \{\mathbf{s}_n^k\} \in \arg\min_{\{\mathbf{s}_n\}} \quad & \sum_{n=1}^{N} \mathbf{s}_n^\top \mathbf{g}_n^k \\ \text{s. to} \quad & \mathbf{s}_n \in \mathcal{P}_n \ \forall n \in \mathcal{N} \end{aligned} \tag{15}$$

where the gradient vector has been partitioned as $\mathbf{g}^k := [(\mathbf{g}_1^k)^\top \ \ldots \ (\mathbf{g}_N^k)^\top]^\top$. Two important observations are in order. Heed first that due to the form of $H(\mathbf{p})$, the per-vehicle partial gradients coincide, that is $\mathbf{g}_n^k = \mathbf{c}^k$ for all $n \in \mathcal{N}$; the t-th entry of vector $\mathbf{c}^k$ is

$$c^k(t) := 2(d(t) + \sum_{n=1}^{N} p_n^k(t)) \tag{16}$$

for $t \in \mathcal{T}$. The second observation is that because of the separable structure of its feasible set $\mathcal{P}$, the linear program in (15) decouples across vehicles as

$$\begin{aligned} \mathbf{s}_n^k \in \arg\min_{\mathbf{s}_n} \quad & \mathbf{s}_n^\top \mathbf{c}^k \\ \text{s. to} \quad & \mathbf{s}_n \in \mathcal{P}_n \end{aligned} \tag{17}$$

for all n. Problem (17) entails minimizing a linear function over the polytope $\mathcal{P}_n$. The latter problem can be solved using a simple sorting algorithm [19]: First, the entries of $\mathbf{c}^k$ are sorted in increasing order as $c^k(t_1^k) \leq c^k(t_2^k) \leq \ldots \leq c^k(t_T^k)$. Recognizing that the subproblems (17) share vector $\mathbf{c}^k$ for all n, this sorting operation is performed only once using for example the Merge-Sort algorithm with complexity $\mathcal{O}(T \log T)$ [20]. Subsequently, for every vehicle n, find the index m_n^k for which

$$\sum_{i=1}^{m_n^k} \bar{p}_n(t_i^k) \leq R_n \text{ and } \sum_{i=1}^{m_n^k+1} \bar{p}_n(t_i^k) > R_n. \quad (18)$$

Then, the entries of $\mathbf{s}_n^k$ in (17) are neatly decided as

$$s_n^k(t_i^k) = \begin{cases} \bar{p}_n(t_i^k) & , i = 1, \ldots, m_n^k - 1 \\ R_n - \sum_{j=1}^{m_n^k-1} \bar{p}_n(t_j^k) & , i = m_n^k \\ 0 & , i = m_n^k + 1, \ldots, T \end{cases} \quad (19)$$

per vehicle n. The aforementioned solution reveals that vehicles charge during periods of lowest load. Critically, making the decisions in (19) requires knowing only the *ordering* of the time slots $\{t_1^k, t_2^k, \ldots, t_T^k\}$, and not the actual values of the common gradients $\{c^k(t_1^k), c^k(t_2^k), \ldots, c^k(t_T^k)\}$.

The second step of the Frank-Wolfe algorithm [cf. (2b)] simply performs convex combinations to update the charging profiles for all n as

$$\mathbf{p}_n^{k+1} = (1 - \gamma_k)\mathbf{p}_n^k + \gamma_k \mathbf{s}_n^k. \quad (20)$$

The step size γ_k can be set to $\gamma_k = \frac{2}{k+2}$, or chosen optimally according to (5), which for the problem at hand yields

$$\gamma_k = -\frac{\sum_{t=1}^T a(t)w(t)}{\sum_{t=1}^T (w(t))^2} \quad (21)$$

with $a(t) := d(t) + \sum_{n=1}^N p_n^k(t)$ and $w(t) := \sum_{n=1}^N s_n^k(t) - \sum_{n=1}^N p_n^k(t)$.

A decentralized charging protocol is proposed next to implement the aforementioned Frank-Wolfe control scheme. Each EV is presumed to be equipped with a smart controller that communicates with the aggregator and is able to perform simple computing tasks, such as finding the decisions in (18) and (19). Smart controllers are treated as the nodes of a tree graph whose root is the aggregator server. This cyber architecture matches well the physical system structure of a radial information router system.

To optimally select γ_k from (21), the aggregator needs to acquire the summation vector $\sum_{n=1}^N \mathbf{s}_n^k$ from the vehicle controllers. To achieve that, either each vehicle communicates its own $\mathbf{s}_n^k$ to the aggregator, or the summation $\sum_{n=1}^N \mathbf{s}_n^k$ is successively calculated over the nodes of a tree. The information exchange for the latter scheme is as follows. Each vehicle controller first initializes a $\mathcal{P}_n$–feasible charging profile $\{\mathbf{p}_n^0\}$ with respect to its charging demands $\{(\mathcal{T}_n, R_n)\}$. It then collects the aggregate charging profile from downstream vehicles and adds it up to its own charging profile. The calculated partial sum is forwarded to the next vehicle up the tree. Upon getting $\sum_{n=1}^N \mathbf{p}_n^0$, the aggregator calculates the marginal costs $\{c^0(t)\}_{t \in \mathcal{T}}$ according to (16), and sorts them by their values to obtain the time slot ordering $\{t_1^0, t_2^0, \ldots, t_T^0\}$. The ordering is then broadcast to all vehicles, where controllers implement in parallel the simple decision rule of (19) to acquire $\{\mathbf{s}_n^0\}$. The summation of $\{\mathbf{s}_n^0\}$ is transmitted back to the aggregator as depicted in the right panel of Fig. 1.

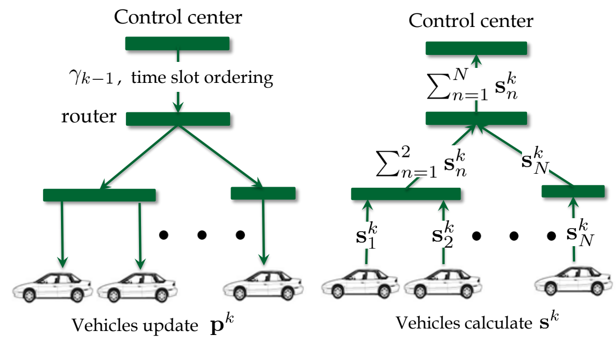

Fig. 1: Information flows for Algorithm 1 at iteration $k \geq 1$. *Left:* Aggregator broadcasts γ_{k-1} and the ordering of $\{\mathbf{c}^k(t)\}$ (smallest to largest) to EVs. *Right:* Summations of intermediate charging profiles $\{\mathbf{s}_n^k\}$ are forwarded to aggregator.

For iterations $k \geq 1$, the aggregator calculates γ_{k-1} from (21). The summation of the charging loads is updated as

$$\sum_{n=1}^N \mathbf{p}_n^k = \sum_{n=1}^N \mathbf{p}_n^{k-1} + \gamma_{k-1} \left(\sum_{n=1}^N \mathbf{s}_n^{k-1} - \sum_{n=1}^N \mathbf{p}_n^{k-1} \right). \quad (22)$$

The gradient $\mathbf{c}^k$ is calculated from (16). Then, γ_{k-1} and the ordering of the entries of $\mathbf{c}^k$ are broadcast to all EVs as demonstrated in the left panel of Fig. 1. Upon receiving γ_{k-1} and the time slot ordering, each vehicle controller calculates $\mathbf{p}_n^k$ via (20), and updates $\mathbf{s}_n^k$ based on (18) and (19). Note that the obtained $\mathbf{p}_n^k$ is stored locally at vehicle n.

According to this architecture, the charging controllers do not submit their charging profiles to the aggregator, thus preserving the privacy of the EV users. The aggregator controller on the other hand does not announce per-slot marginal costs, but instead broadcasts time slot rankings. The overall decentralized charging scheme is summarized in Algorithm 1.

A PGD based scheme for solving (10) has been suggested in [8]. According to that scheme, at iteration k every vehicle updates its charging profile as

$$\mathbf{p}_n^{k+1} := \arg \min_{\mathbf{p}_n \in \mathcal{P}_n} \|\mathbf{p}_n - (\mathbf{p}_n^k - \mu \mathbf{c}^k)\|_2^2 \quad (23)$$

for a step size $\mu > 0$. Problem (23) projects vector $(\mathbf{p}_n^k - \mu \mathbf{c}^k)$ onto $\mathcal{P}_n$, which is a non-trivial computational task. Granted that (11) entails a non-strongly convex objective function, the convergence rate of this PGD scheme is at most $\mathcal{O}\left(\frac{1}{k}\right)$; see [21].

Algorithm 1 Optimal decentralized charging control

Input: Stopping criterion ϵ.

1: Initialize $\{\mathbf{p}_n^0\}$, $\mathbf{c}^0$, and $\{\mathbf{s}_n^0\}$.
2: **for** $k = 1, 2, \ldots$ **do**
3: Aggregator obtains γ_k from (21);
4: (or it sets $\gamma_k = \frac{2}{k+2}$).
5: Aggregator updates $\sum_{n=1}^N \mathbf{p}_n^k$ according to (22).
6: Aggregator evaluates $\mathbf{c}^k$ using (16).
7: Aggregator sends sorted $\{c^k(t)\}_{t=1}^T$ to all EVs.
8: Vehicle n calculates $\{\mathbf{p}_n^k\}$ via (20).
9: Vehicle n updates $\{\mathbf{s}_n^k\}$ from (18) and (19).
10: **end for**

B. Asynchronous Updates

At each iteration, Alg. 1 requires all vehicles to update their charging profiles according to the current control signal. In practical charging scenarios, vehicle controllers may not be able to update their charging profiles synchronously. That could be the result of failures in the communication links between the aggregator and the vehicle controllers, or due to processing delays in vehicle controllers. In such scenarios, the step size γ_k has to be modified to guarantee the convergence of Alg. 1. Let us assume that lost updates occur independently at random across iterates and vehicles. If the probability of a successful update is larger than $\alpha := \bar{N}/N$, the step size γ_k can be modified as

$$\gamma_k = \frac{2}{\alpha k + 2}. \quad (24)$$

To proceed with the iteration complexity analysis, define $\tilde{h}(\mathbf{p}) := H(\mathbf{p}) - H(\mathbf{p}^*)$, and denote the diameter of the feasible set $\mathcal{P}$ as

$$D_{\mathcal{P}} := \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{P}} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2.$$

Proposition 3. *If the probability of successful updates is larger than α and the step size is set as $\gamma_k = \frac{2}{\alpha k+2}$, then Algorithm 1 achieves*

$$\mathbb{E}[\tilde{h}(\mathbf{x}^k)] \leq \frac{4(1-\alpha)\tilde{h}(\mathbf{x}^0) + 2kD_{\mathcal{P}}}{(\alpha k + 2 - \alpha)^2} \quad (25)$$

for $k \geq 1$.

Proof. In view of [22, Lemma 2], the next inequality holds

$$\mathbb{E}[\tilde{h}(\mathbf{x}^{k+1})] \leq (1 - \alpha\gamma_k)\mathbb{E}[\tilde{h}(\mathbf{x}^k)] + \frac{\gamma_k^2}{2} D_{\mathcal{P}} \quad (26)$$

where the expectation operator $\mathbb{E}[\cdot]$ is applied over communication failures. For $k = 1$, inequality (25) holds, since $\mathbb{E}[\tilde{h}(\mathbf{x}^1)] \leq (1-\alpha)\tilde{h}(\mathbf{x}^0) + \frac{1}{2}D_{\mathcal{P}}$ by (26).

Proving by induction, assume that (25) holds true for k. To prove (25) for $k+1$, note that

$$\begin{aligned}(\alpha k + 2 - 2\alpha)(\alpha k + 2) &= (\alpha k + 2 - \alpha)^2 - \alpha^2 \\ &\leq (\alpha k + 2 - \alpha)^2\end{aligned}$$

thus yielding

$$\frac{\alpha k + 2 - 2\alpha}{(\alpha k + 2 - \alpha)^2} \leq \frac{1}{\alpha k + 2}. \quad (27)$$

From $1 - \alpha\gamma_k = \frac{\alpha k + 2 - 2\alpha}{\alpha k + 2}$ and (27), it holds that

$$\frac{1 - \alpha\gamma_k}{(\alpha k + 2 - \alpha)^2} \leq \frac{1}{(\alpha k + 2)^2}. \quad (28)$$

Plugging (28) into (26) provides

$$\begin{aligned}\mathbb{E}[\tilde{h}(\mathbf{x}^{k+1})] &\leq \frac{4(1-\alpha)\tilde{h}(\mathbf{x}^0)}{(\alpha k+2)^2} + \frac{2kD_{\mathcal{P}}}{(\alpha k+2)^2} + \frac{2D_{\mathcal{P}}}{(\alpha k+2)^2} \\ &= \frac{4(1-\alpha)\tilde{h}(\mathbf{x}^0) + 2(k+1)D_{\mathcal{P}}}{(\alpha k+2)^2}\end{aligned}$$

thus proving the claim. □

Stochastic Frank-Wolfe updates have also been been considered in [22], where the step size was modified as $\gamma_k = \frac{2\alpha}{\alpha^2 k + 2/N}$ which can yield $\gamma_k > 1$ (check e.g., γ_1 and γ_2 for $N = 100$, and $\alpha = 0.9$), thus introducing infeasibility issues in the Frank-Wolfe iterates.

To summarize, considering asynchronous updates of EVs, the step size of Alg. 1 has to be modified. Proposition 3 asserts that for the step size rule in (24), the updates of Alg. 1 remain feasible at all times, while the objective value is guaranteed to converge in expectation with rate $\mathcal{O}\left(\frac{1}{k}\right)$.

V. Numerical Tests

The efficacy of the devised charging scheme was verified by simulating the charging of 52 EVs. The battery capacity of all vehicles was assumed to be 24 kWh. The maximum charging power was fixed to 3.45 kW. According to actual travel survey data [23], the probability density function (pdf) for EV plug-in times in hours is

$$f_{\text{in}}(\tau) = \begin{cases} \mathcal{N}(\mu_{\text{in}} - 24, \sigma_{\text{in}}), & 0 < \tau \leq \mu_{\text{in}} - 12 \\ \mathcal{N}(\mu_{\text{in}}, \sigma_{\text{in}}), & \mu_{\text{in}} - 12 < \tau \leq 24 \end{cases} \quad (29)$$

where $\mu_{\text{in}} = 17.47$, $\sigma_{\text{in}} = 3.41$, and $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian pdf with mean μ and variance σ^2. The pdf for plug-out times in hours is

$$f_{\text{out}}(\tau) = \begin{cases} \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}}), & 0 < \tau \leq \mu_{\text{out}} + 12 \\ \mathcal{N}(\mu_{\text{out}} + 24, \sigma_{\text{out}}), & \mu_{\text{out}} + 12 < \tau \leq 24 \end{cases} \quad (30)$$

where $\mu_{\text{out}} = 8.92$ and $\sigma_{\text{out}} = 3.24$. Moreover, daily travel miles are distributed according to

$$f_{\text{miles}}(y) = \frac{1}{\sqrt{2\pi}\sigma_{\text{miles}} y} \exp\left(-\frac{(\log(y) - \mu_m)^2}{2\sigma_m^2}\right) \quad (31)$$

where $\mu_{\text{miles}} = 2.98$ and $\sigma_{\text{miles}} = 1.14$. For each vehicle, the expected state of charge (SOC) was set to 90%. The energy needed per 100 km was $E_{100} = 15$ kWh, and the initial SOC was obtained as $S_n^0 = 0.9 - M_n E_{100}/(100 B_n)$, where M_n denotes daily travel miles for vehicle n, and B_n is the battery capacity of vehicle n. Normalized base load curves with base unit 1000 kW were obtained by averaging the 2014 residential load data from Southern California Edison [24]. The simulation horizon, set from 12:00 pm to 12:00 pm the

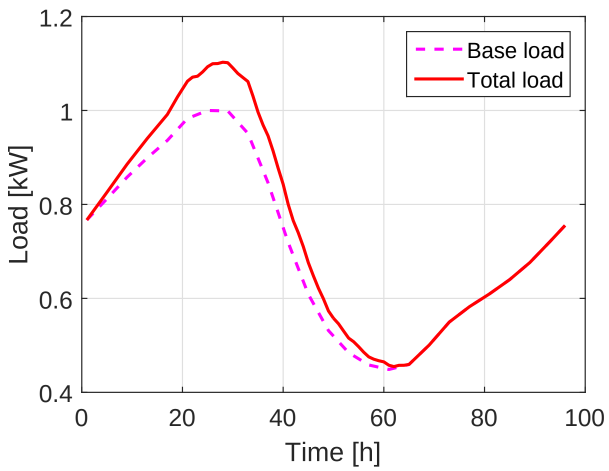

Fig. 2: Load curves after uncoordinated charging of 52 EVs.

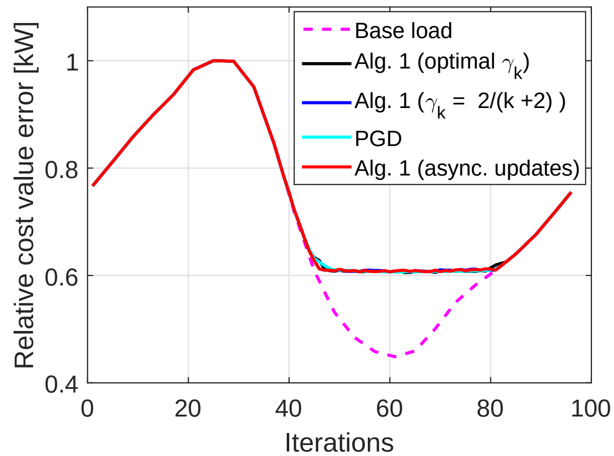

Fig. 3: Load curves after optimal charging of 52 EVs.

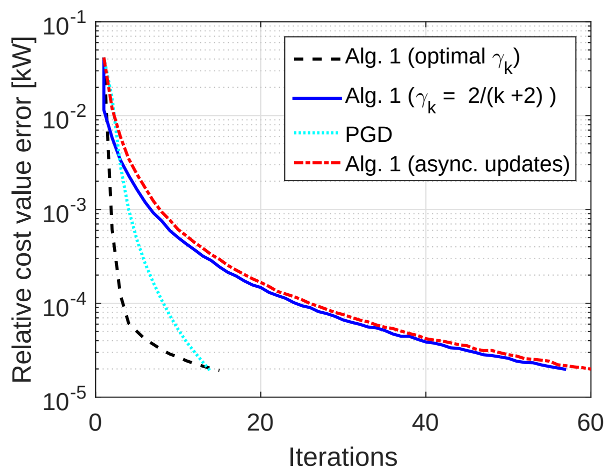

Fig. 4: Convergence performance of Alg. 1 and PGD.

next day, comprised $T = 96$ time slots. Numerical tests were run using Matlab on an Intel CPU @ 3.6 GHz (32 GB RAM) computer.

Uncoordinated charging: In this case, all EVs were assumed to begin charging as soon as they plug-in, and stop charging when the desired SOC level was reached. The resultant total load curves were demonstrated in Fig. 2, which clearly indicates that uncoordinated charging for even 52 EVs increases the load peak tremendously.

Optimal decentralized scheduling: Algorithm 1 with optimal γ_k or with $\gamma_k = \frac{2}{k+2}$, was compared to the PGD method of [8]. In PGD, the subproblems (23) are tackled via the default solver provided by YALMIP [25]. Asynchronous updates where 51 out of 52 EVs were randomly selected to update their charging profiles at each iteration, was also tested. To serve as a benchmark, the quadratic problem (12) was solved using the off-the-shelf solver SeDuMi. All algorithms were initialized using the uncoordinated charging profiles and run until the relative cost error ϵ became smaller than 2×10^{-5}. As can be seen from Fig. 3, all the obtained load curves feature a flat load valley without increasing the peak load, thus verifying the efficacy of Alg. 1 and PGD. However, running both algorithms sequentially, the required computational time differs significantly: 41.1 sec for Alg. 1 with optimal γ_k; 174.9 μsec for Alg. 1 with $\gamma_k = \frac{2}{k+2}$; 187.7 μsec for Alg. 1 with asynchronous updates; and 94.84 sec for PGD. The numerical tests confirm that the proposed methods enjoy a notable speedup advantage over existing alternatives.

The convergence curves of our novel scheme and PGD are shown in Fig. 4: The decreasing rate of Alg. 1 with optimal step size is large at first and gradually decays. The iterations required by Alg. 1 with the optimal step size and PGD to reach an ϵ smaller than 2×10^{-5} are almost identical. Even with asynchronous updates, Alg. 1 converges at a rate similar to the synchronous Alg. 1 for $\gamma_k = \frac{2}{k+2}$. Figure 5 shows how the running time per update (averaged over vehicles and iterations) scales with the charging slots T: Although the update time increases linearly with T for both schemes, our method requires times in the order of 10^{-5} sec, while PGD times are in the order of 0.1 sec. This superior behavior is attributed to the closed-form updates in (19).

VI. Conclusions

Optimal charging of EVs introduces time-coupling constraints, while the problem dimension increases linearly with the number of vehicles and time slots. To address the computational issues involved in scheduling large fleets over fine timescales, a decentralized charging scheme was developed based on the Frank-Wolfe method. Based on numerical tests, the novel charging protocol converged 100 times faster than a competing alternative, while its closed-form updates pose minimal computing requirements to vehicle controllers. To account for random cyber delays, an asynchronous variant was also devised. This stochastic block-coordinate protocol was shown to converge at the rate of $\mathcal{O}\left(\frac{1}{k}\right)$ upon properly controlling its step size. Extending our schemes to real-time vehicle scheduling constitutes a challenging research direction.

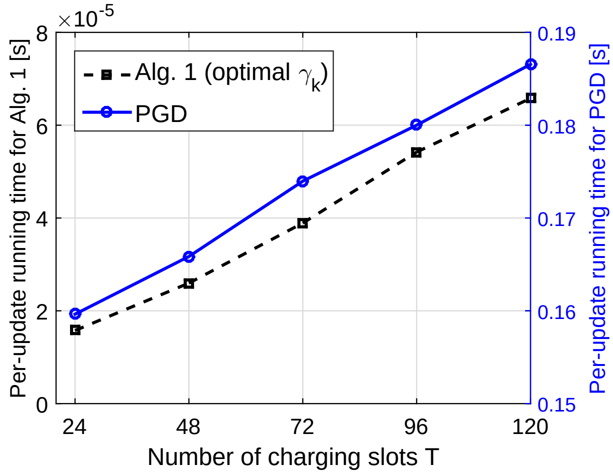

Fig. 5: Computational time as a function of charging periods T for Alg. 1 and PGD. Note the scale difference between the two plots.

REFERENCES

[1] P. Richardson, D. Flynn, and A. Keane, "Optimal charging of electric vehicles in low-voltage distribution systems," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 268–279, Feb. 2012.

[2] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Processing Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[3] Z. Fan, "A distributed demand response algorithm and its application to PHEV charging in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1280–1290, Sep. 2012.

[4] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control for large populations of plug-in electric vehicles," in *Proc. Conf. on Decision and Control*, Atlanta, GA, Dec. 2010.

[5] Y. Zhang, N. Gatsis, and G. B. Giannakis, "Robust energy management for microgrids with high-penetration renewables," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 944–953, Oct. 2013.

[6] D. Papadaskalopoulos, G. Strbac, P. Mancarella, M. Aunedi, and V. Stanojevic, "Decentralized participation of flexible demand in electricity markets; Part II: Application with electric vehicles and heat pump systems," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3667–3674, Nov. 2013.

[7] R. Li, Q. Wu, and S. Oren, "Distribution locational marginal pricing for optimal electric vehicle charging management," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 203–211, Jan. 2014.

[8] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, May 2013.

[9] S. Xu, D. Feng, Z. Yan, L. Zhang, N. Li, L. Jing, and J. Wang, "Ant-based swarm algorithm for charging coordination of electric vehicles," *International Journal of Distributed Sensor Networks*, 2013.

[10] E. L. Karfopoulos and N. D. Hatziargyriou, "A multi-agent system for controlled charging of a large population of electric vehicles," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1196–1204, May 2013.

[11] J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen, "Alternating direction method of multipliers for decentralized electric vehicle charging control," in *Proc. Conf. on Decision and Control*, Florence, Italy, Dec. 2013.

[12] W.-J. Ma, V. Gupta, and U. Topcu, "On distributed charging control of electric vehicles with power network capacity constraints," in *Proc. American Control Conference*, Portland, OR, Jun. 2014.

[13] O. Ardakanian, C. Rosenberg, and S. Keshav, "Distributed control of electric vehicle charging," in *Proc. Intl. Conf. on Future Energy Systems*, Berkeley, CA, May 2013.

[14] Q. Li, T. Cui, R. Negi, F. Franchetti, and M. D. Ilic, "On-line decentralized charging of plug-in electric vehicles in power systems," 2011. [Online]. Available: http://arxiv.org/pdf/1106.5063v2.pdf

[15] N. Chen, C. W. Tan, and T. Quek, "Electric vehicle charging in smart grid: Optimality and valley-filling algorithms," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 6, pp. 1073–1083, Dec. 2014.

[16] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable electric vehicle charging protocols," *IEEE Trans. Power Syst.*, 2016, (early access).

[17] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proc. Intl. Conf. on Machine Learning*, Atlanta, GA, Jun. 2013.

[18] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1–2, pp. 95–110, 1956.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[20] D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*. Boston, MA: Pearson Education, 1998, vol. 3.

[21] D. P. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA: Athena Scientific, 2015.

[22] Y. X. Wang, V. Sadhanala, W. Dai, W. Neiswanger, S. Sra, and E. P. Xing, "Parallel and distributed block-coordinate Frank-Wolfe algorithms," in *Proc. Intl. Conf. on Machine Learning*, New York City, NY, Jun. 2016.

[23] Federal highway administration. US Dept. of Transportation. [Online]. Available: http://nhts.ornl.gov/2009/pub/stt.pdf

[24] Southern california edison dynamic load profiles. Southern California Edison. [Online]. Available: https://www.sce.com/wps/portal/home/regulatory/load-profiles/

[25] J. Lofberg, "A toolbox for modeling and optimization in MATLAB," in *Proc. of the CACSD Conf.*, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip/