Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

A New Decentralized Algorithm for Optimal Load Shifting via Electric Vehicles

Hao Xing[1], Zhiyun Lin[1], Minyue Fu[2]

1. School of Automation, Hangzhou Dianzi University, Hangzhou 310018, P.R. China
E-mail: xing.h@hdu.edu.cn; linz@hdu.edu.cn

2. School of Electrical Engineering and Computer Science, University of Newcastle, NSW 2308 Australia
E-mail: minyue.fu@newcastle.edu.au

Abstract: In this paper we investigate the optimal load shifting problem via electric vehicles (EVs) in a smart grid scenario which aims at flattening the total demand curve as much as possible while each EV's local constraints are satisfied. We assume bidirectional energy exchange between EVs and the power grid and formulate the problem as a mixed integer quadratic programming problem. To solve this problem in a decentralized fashion, we propose a decentralized optimal algorithm where EVs and the aggregator cooperatively find the optimal solution by communicating in a star network and conducting local computation. To implement the proposed algorithm, only limited data are exchanged and the aggregator does not need any parameter information of EVs. The proposed algorithm converges much faster than traditional centralized methods/commercial solvers, as the mixed integer part is broken down into local subproblems and solved in parallel by each EV. Convergence proof of the proposed algorithm is presented and numerical experiments show the effectiveness of the proposed algorithm.

Key Words: Decentralized Algorithm, Electric Vehicle, Load Shifting, Mixed Integer Programming, Smart Grid, V2G

1 Introduction

EVs are recognized as a promising alternative to traditional gas-powered vehicles due to the rising environmental concerns and recent advances in battery technologies. Despite the environmental and economic benefits, EVs pose a great challenge to power systems in the sense that the uncoordinated charging behavior of EVs can worsen the load profile, e.g., adding to existing demand peaks, and consequently threaten the security and stability of power systems [1]. However, with well-designed control algorithms, EVs can not only be safely incorporated into power systems, but also provide demand side management services including energy arbitrary, spinning reserve, and load shifting, since EVs can be viewed as energy storage when plugged-in. The energy flow between EVs and power systems can be bidirectional, i.e., EVs can not only take energy from power grids by the charging process, but also release energy back into power grids through smart inverters, which are known as grid-to-vehicle (G2V) and vehicle-to-grid (V2G), respectively [2]. Specifically, G2V can achieve "valley filling" while V2G can be used for "peak shaving".

Since EVs are a challenge but also a great opportunity to future smart grids, recently the energy scheduling of EVs has been intensively investigated in a variety of scenarios. Existing works can basically be divided into two categories: those only considering G2V and the other considering both G2V and V2G. The charging problem is modelled as a finite-horizon dynamic game and a decentralized solution is proposed in [3]. In [4], a comprehensive and in-depth analysis on EVs' optimal charging scheduling for load shifting is presented and an optimal decentralized charging algorithm is proposed. A series of decentralized algorithms for charging scheduling based on the decentralized water-filling-based algorithm are proposed in [5–7] to optimally flatten the load curve of low-voltage transformers in various scenarios. The above papers only consider G2V while many other works have also considered V2G. Two decentralized algorithms for globally and locally optimal scheduling of charging/discharging are proposed in [8] which aims at the minimization of aggregated charging cost of EVs with random arrivals. The random behavior of EVs is also considered in [9] where the expected grid operation cost is minimized using a stochastic security-constrained unit commitment model. Reference [10] considers the uncertainties from renewable energy sources in smart grids and propose an optimal charging and discharging scheduling algorithm based on dynamic programming. In [11], the demand of a single household is locally regulated through EVs such that the system-wide demand is regulated. Our previous work [12] is the first paper to study the optimal load shifting problem via G2V and V2G in the context of decentralized algorithms. With a proper assumption, the original mixed discrete programming problem is approximated as a quadratic programming (QP) problem, and a decentralized algorithm based on the water-filling algorithm is proposed to solve the approximated QP problem.

This paper revisits the load shifting problem studied in [12]. We reformulate the problem as a mixed integer quadratic programming (MIQP) problem and proposes a new decentralized algorithm based on the alternating direction method of multipliers (ADMM). In comparison, the novel features of this paper include:

1. The decentralized water-filling algorithm proposed in [12] relies on a reasonable assumption on the night demand curve. However, in this paper we drop this assumption and propose a decentralized optimal algorithm to directly solve the original problem in a decentralized fashion. Therefore, the results in this paper are of more general applicability.
2. Adopting the idea of ADMM, the optimal load shifting problem is decomposed such that an aggregator solves an unconstrained QP problem as the master problem,

This work was supported by the National Natural Science Foundation of China (Grant No. 61673344).

while each EV solves a local MIQP problem as subproblems. To implement the iterative algorithm, only limited data are transmitted between the EVs and the aggregator. Furthermore, the optimal load shifting problem is NP-hard and the integer aspect is the major source of the computational complexity. With each EV updating the integer variables in parallel via local computation, the proposed decentralized algorithm requires much less CPU time than traditional centralized algorithms/off-the-shelf commercial solvers.

The rest of the paper is organized as follows. Preliminaries and the problem formulation are given in Section 2. We present the ADMM based decentralized algorithm in Section 3. Numerical experiments are presented in Section 4 to show the effectiveness of the proposed algorithm. Finally, we conclude our paper in Section 5.

2 Preliminaries and Problem Formulation

In this section, we first introduce some basics of ADMM. We then present the charging/discharging model of a single EV and the problem formulation of the optimal load shifting problem via EVs.

2.1 Alternating Direction Method of Multipliers

Consider the following problem:

$$\begin{array}{ll} \text{minimize} & f(x)+g(y), \\ \text{subject to} & Ax+By=c, \end{array} \tag{1}$$

with optimization variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, where A, B, and c are in proper dimensions.

We make the following assumptions for the problem (1):

Assumption 1 *The functions $f(x): \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g(y): \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are proper, closed and convex.*

Assumption 2 *The matrices A and B have full column ranks.*

The augmented Lagrangian of problem (1) is

$$\begin{aligned} L_\rho(x,y,\lambda) =& f(x)+g(y)+\lambda^T(Ax+By-c) \\ & +\rho/2\|Ax+By-c\|_2^2, \end{aligned} \tag{2}$$

where $\lambda \in \mathbb{R}^p$ is the Lagrange multiplier, and $\rho > 0$ is a penalty parameter. The alternating direction method of multipliers for problem (1) is given as follows [13]:

$$x^{k+1} = \operatorname*{argmin}_x L_\rho(x,y^k,\lambda^k); \tag{3}$$

$$y^{k+1} = \operatorname*{argmin}_y L_\rho(x^{k+1},y,\lambda^k); \tag{4}$$

$$\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1}+By^{k+1}-c). \tag{5}$$

Define

$$\epsilon^k = Ax^k + By^k - c \text{ and } \varepsilon^k = \rho A^T B(y^{k+1}-y^k)$$

as the primal residual and the dual residual at step k, respectively. The convergence results of ADMM are given below.

Lemma 1 *[13] If Assumptions 1 and 2 hold and $\rho > 0$, then the ADMM iterations* (3)-(5) *converge to the optimal solution x^*, y^*, and the optimal Lagrange multiplier λ^* of problem* (1)*, with*

$$\lim_{k\to\infty} \|\epsilon^k\|_2 = 0 \textit{ and } \lim_{k\to\infty} \|\varepsilon^k\|_2 = 0.$$

2.2 Dynamic Model of A Single EV

Suppose the sampling interval is Δt and let $t = 0,1,\dots,T-1$ denote the time steps. The state of charge (SOC) is defined as the energy level of a battery, given by

$$s(t) = \frac{E(t)}{C},$$

where $E(t)$ (kWh) is the amount of residual energy at time t and C (kWh) is the capacity of the battery.

Let $p(t)$ (kW) denote the rate of energy transfer between the EV and the power grid at time t. Suppose that $p(t)$ is constant in the interval between t and $t+1$, and

$$\begin{cases} p(t) \geqslant 0, & \text{if charging,} \\ p(t) < 0, & \text{if discharging.} \end{cases}$$

The rate of charging/discharging cannot exceed certain limits, i.e.,

$$-\bar{p}^{\mathrm{dc}} \leqslant p(t) \leqslant \bar{p}^{\mathrm{c}}, \tag{6}$$

where $\bar{p}^{\mathrm{c}}, \bar{p}^{\mathrm{dc}} > 0$ are the maximum rates of charging and discharging, respectively. It is assumed that with *smart chargers* and *smart inverters*, $p(t)$ can take any continuous value within the limits. We consider energy losses in both directions of energy conversion, and define $0 < \eta^{\mathrm{c}}, \eta^{\mathrm{dc}} < 1$ as efficiencies for charging and discharging, respectively.

The dynamic model of a single EV is given by

$$s(t+1) = s(t) + \frac{\eta(t)\Delta t p(t)}{C},$$

where $\eta(t)$ is given by

$$\eta(t) = \begin{cases} \eta^{\mathrm{c}} < 1, & \text{for } p(t) \geqslant 0, \\ \dfrac{1}{\eta^{\mathrm{dc}}} > 1, & \text{for } p(t) < 0. \end{cases} \tag{7}$$

To increase the service life of batteries, we require that the SOC is kept within a certain range. Therefore we have the following constraints,

$$\underline{s} \leqslant s(t) \leqslant \bar{s},\ t = 0,\dots,T, \tag{8}$$

where $s(0)$ is the initial SOC at time 0. In this paper we set $\underline{s}$ =20% and $\bar{s}$ =85%.

We further assume that at time T, each EV must be charged to a targeted SOC $s^* \in [\underline{s}, \bar{s}]$. Therefore we have the following equality constraint,

$$s(T) = s^*. \tag{9}$$

2.3 Problem Formulation

Suppose there are in total n EVs in the network indexed by $i = 1,\dots,n$, and denote by $d(t)$ the base demand at time t. Note that function (7) is not in a code-friendly fashion and it is nonlinear since $\eta_i(t)$ depends on the sign of $p_i(t)$. To overcome this issue, we introduce a binary variable $z_i(t) \in \{0,1\}$ to indicate the direction of energy conversion at time t, and establish two auxiliary variables $p_i^{\mathrm{c}}(t)$ and $p_i^{\mathrm{dc}}(t)$ such that

$$p_i(t) = p_i^{\mathrm{c}}(t) - p_i^{\mathrm{dc}}(t),\ p_i^{\mathrm{c}}(t)p_i^{\mathrm{dc}}(t) = 0,$$

i.e., we assume that the energy rate $p_i(t)$ consists of two components associated with the two directions of energy conversion, while (at least) one of them must be zero. Based on the above definitions, we then re-write (6)-(9) for all i, t as follows:

$$0 \leqslant p_i^{\mathrm{c}}(t) \leqslant \bar{p}_i^{\mathrm{c}} z_i(t), \tag{10}$$

$$0 \leqslant p_i^{\mathrm{dc}}(t) \leqslant \bar{p}_i^{\mathrm{dc}}(1 - z_i(t)), \tag{11}$$

$$s_i(0) + \Delta t \frac{\sum_{t=0}^{T-1} (\eta_i^{\mathrm{c}} p_i^{\mathrm{c}}(t) - p_i^{\mathrm{dc}}(t)/\eta_i^{\mathrm{dc}})}{C_i} = s_i^*, \tag{12}$$

$$\underline{s} \leqslant s_i(0) + \Delta t \frac{\sum_{\tau=0}^{t-1} (\eta_i^{\mathrm{c}} p_i^{\mathrm{c}}(\tau) - p_i^{\mathrm{dc}}(\tau)/\eta_i^{\mathrm{dc}})}{C_i} \leqslant \bar{s}. \tag{13}$$

One can easily verify that with the binary variable $z_i(t)$, $p_i^{\mathrm{c}}(t) p_i^{\mathrm{dc}}(t) = 0$ is guaranteed by constraints (10) and (11). The load scheduling problem studied in this paper aims at minimizing the variations in the total demand curve by controlling the rate of charging and discharging of EVs within their local constraints. The objective is given by

$$\text{minimize} \quad \sum_{t=0}^{T-1} \left(d(t) + \sum_{i=1}^{n} (p_i^{\mathrm{c}}(t) - p_i^{\mathrm{dc}}(t)) \right)^2. \tag{14}$$

Therefore, the optimal load shifting problem via EVs with both G2V and V2G are formulated as the MIQP problem (10)-(14), where each EV maintains $2nT$ continuous variables and nT binary variables.

3 Main Results

In this section we first analyze the mathematical properties of problem (10)-(14) and re-write it in an equivalent ADMM from. We then present the decentralized algorithm based on ADMM and its convergence analysis.

3.1 Problem Analysis

The problem (10)-(14) is a MIQP problem, as it includes the continuous variables $p_i^{\mathrm{c}}(t), p_i^{\mathrm{dc}}(t)$ and the binary variable $z_i(t)$. However, the binary variable $z_i(t)$ is only involved in the constraints, while the objective function (14) itself remains convex.

It is important to note that $z_i(t)$'s are essentially auxiliary variables which help us describe explicitly the feasible region of $p_i^{\mathrm{c}}(t)$ and $p_i^{\mathrm{dc}}(t)$. To be specific, the feasible set defined by the constraints (10)-(13) can be viewed as a union of 2^{nT} disjoint convex sets regarding the continuous variables $p_i^{\mathrm{c}}(t)$ and $p_i^{\mathrm{dc}}(t)$. Let us further define $\mathbf{p}$ and $\mathbf{z}$ as the aggregated vectors of $p_i^{\mathrm{c}}(t)$, $p_i^{\mathrm{dc}}(t)$ and $z_i(t)$ for all i, t, respectively. Problem (10)-(14) can be re-written equivalently in the following compact form:

$$\begin{aligned} &\text{minimize} && F(\mathbf{p}), \\ &\text{subject to} && \mathbf{p} \in \Omega = \bigcup_{\mathbf{z}} \Omega(\mathbf{z}), \\ & && \mathbf{z} \text{ is binary}, \end{aligned} \tag{15}$$

where $\Omega(\mathbf{z}) = \{\mathbf{p} : \text{Constraints } (10) - (13)\}$ is the convex feasible subset of $\mathbf{p}$ associated with a fixed realization of the binary variable vector $\mathbf{z}$; Ω is a non-convex set, since it is the union of the disjoint convex subsets.

Since some realizations of $\mathbf{z}$ may lead to empty feasible subsets of $\mathbf{p}$, we define M as the total number of the nonempty subsets. Therefore, problem (15) essentially consists of M convex QP subproblems. Intuitively, if the optimal solution to each QP subproblem is obtained, then the one associated with the minimal optimal objective value is the optimal solution to the whole problem. However, such scheme is time-consuming and even unrealistic, for M may probably be a very large number.

We aims at proposing a decentralized algorithm for the problem (10)-(14), so it is important to investigate the spatial correlations among EVs. One can verify that the constraints (10)-(13) are all local and only account for the temporal correlations among local variables at each EV, while the only source of spatial correlations is the objective function (14). For the ith EV, let us define $\mathbf{p}_i$ as the local aggregate variable and denote by Ω_i the local feasible region such that

$$\Omega = \prod_{i=1}^{n} \Omega_i,$$

where $\prod$ is the operator of Cartesian product of sets. Note that for simplicity, we omit the binary variable $\mathbf{z}$ and describe the sets in an implicit fashion instead. Then problem (15) can be rewritten as follow:

$$\begin{aligned} &\text{minimize} && F(\mathbf{p}), \\ &\text{subject to} && \mathbf{p}_i \in \Omega_i, \forall i = 1, \ldots, n. \end{aligned} \tag{16}$$

3.2 Rewriting (15) in ADMM Form

Before presenting our ADMM-based decentralized algorithm, we need to transform (15) into the ADMM form.

We establish a variable $\mathbf{q}$ as the duplicate of the variable $\mathbf{p}$. We label each non-empty convex feasible subset associated with certain realization of $\mathbf{z}$ by $m = 1, \ldots, M$. For each subset $\Omega(m)$, we define its indicator function $G_m(\mathbf{p})$ as

$$G_m(\mathbf{p}) = \begin{cases} 0, & \text{if } \mathbf{p} \in \Omega(m), \\ +\infty, & \text{otherwise.} \end{cases}$$

Then the indicator function of the overall feasible region Ω is given by

$$G(P) = \min\{G_1(\mathbf{p}), \ldots, G_M(\mathbf{p})\},$$

where min represents the point-wise minimum of functions. Note that the indicator functions $G_m(\mathbf{p})$'s satisfy Assumption 1, as they are associated with convex subsets. However, one can easily verify that the indicator function $G(\mathbf{p})$ is non-convex.

With the above definitions, we can equivalently transform problem (15) in the following ADMM form:

$$\begin{aligned} &\text{minimize} && F(\mathbf{p}) + G(\mathbf{q}), \\ &\text{subject to} && \mathbf{p} - \mathbf{q} = 0. \end{aligned} \tag{17}$$

Applying the ADMM to problem (17) yields

$$\mathbf{p}^{k+1} = \operatorname*{argmin}_{\mathbf{p}} L_\rho(\mathbf{p}, \mathbf{q}^k, \boldsymbol{\lambda}^k), \tag{18}$$

$$\mathbf{q}^{k+1} = \operatorname*{argmin}_{\mathbf{q}} L_\rho(\mathbf{p}^{k+1}, \mathbf{q}, \boldsymbol{\lambda}^k), \tag{19}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{p}^{k+1} - \mathbf{q}^{k+1}), \tag{20}$$

where $L_\rho(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda})$ is the augmented Lagrangian for problem (17), given by

$$L_\rho(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = F(\mathbf{p}) + G(\mathbf{q}) + \boldsymbol{\lambda}^T(\mathbf{p} - \mathbf{q}) + \frac{\rho}{2}\|\mathbf{p} - \mathbf{q}\|_2^2. \tag{21}$$

3.3 ADMM-based Decentralized Optimal Algorithm

In this subsection we propose the decentralized optimal decentralized based on ADMM. For implementing the proposed decentralized algorithm, a cyber layer is required. To be specific, we assume that a system aggregator which knows the base demand data, communicates bidirectionally with the EVs in a star network where the aggregator is the central hub node while the EVs are the leaf nodes. It is also assumed that the aggregator and EVs are capable of local computation. The detailed steps of the proposed algorithm are as follows.

For $\mathbf{p}$-update: We have

$$\begin{aligned}\mathbf{p}^{k+1} &= \underset{\mathbf{p}}{\operatorname{argmin}}\, L_\rho(\mathbf{p}, \mathbf{q}^k, \boldsymbol{\lambda}^k) \\ &= \underset{\mathbf{p}}{\operatorname{argmin}} \big(F(\mathbf{p}) + (\boldsymbol{\lambda}^k)^T(\mathbf{p}-\mathbf{q}^k) + \frac{\rho}{2}\|\mathbf{p}-\mathbf{q}^k\|_2^2\big) \\ &= \underset{\mathbf{p}}{\operatorname{argmin}} \big(F(\mathbf{p}) + \frac{\rho}{2}\|\mathbf{p}-\mathbf{q}^k+\boldsymbol{\lambda}^k\|_2^2\big).\end{aligned} \tag{22}$$

One can easily verify that the $\mathbf{p}$-update is an unconstrained QP problem which can be easily solved by existing QP methods, e.g., gradient method. Therefore based on $\mathbf{q}^k$ and $\boldsymbol{\lambda}^k$, the aggregator performs the $\mathbf{p}$-update without needing to know each EVs' parameters. Furthermore, note that $\mathbf{p}$-update consists of T independent subproblems

$$\min_{\mathbf{p}(t)} \ \big(d(t)+\sum_{i=1}^{n}(p_i^{\mathrm{c}}(t)-p_i^{\mathrm{dc}}(t))\big)^2+\frac{\rho}{2}\|\mathbf{p}(t)-\mathbf{q}^k(t)+\boldsymbol{\lambda}^k(t)\|_2^2.$$

Therefore the $\mathbf{p}$-update can be solved in parallel for faster convergence if the aggregator is capable of multi-core computing.

For $\mathbf{q}$-update: It follows that

$$\begin{aligned}\mathbf{q}^{k+1} &= \underset{\mathbf{q}}{\operatorname{argmin}}\, L_\rho(\mathbf{p}^{k+1}, \mathbf{q}, \boldsymbol{\lambda}^k) \\ &= \underset{\mathbf{q}}{\operatorname{argmin}} \big(G(\mathbf{q}) + \frac{\rho}{2}\|\mathbf{p}^{k+1}-\mathbf{q}+\boldsymbol{\lambda}^k\|_2^2\big). \\ &= \underset{\mathbf{q}\in\Omega}{\operatorname{argmin}} \|\mathbf{p}^{k+1}-\mathbf{q}+\boldsymbol{\lambda}^k\|_2^2.\end{aligned} \tag{23}$$

According to (23), the $\mathbf{q}$-update can be written as the following optimization problem:

$$\begin{aligned}&\text{minimize} && \|\mathbf{p}^{k+1}-\mathbf{q}+\boldsymbol{\lambda}^k\|_2^2, \\ &\text{subject to} && \mathbf{q}\in\Omega,\end{aligned} \tag{24}$$

which can be readily decomposed into n local optimization problems, given by $\forall i = 1, \ldots, n$,

$$\begin{aligned}&\text{minimize} && \|\mathbf{p}_i^{k+1}-\mathbf{q}_i+\boldsymbol{\lambda}_i^k\|_2^2, \\ &\text{subject to} && \mathbf{q}_i\in\Omega_i.\end{aligned} \tag{25}$$

We further re-write the subproblem (24) in the following explicit form:

$$\begin{aligned}&\text{minimize} && \|\mathbf{p}_i^{k+1}-\mathbf{q}_i+\boldsymbol{\lambda}_i^k\|_2^2, \\ &\text{over} && \mathbf{q}_i \text{ and } \mathbf{z}_i \\ &\text{subject to} && \text{local constraints } (10)-(13).\end{aligned} \tag{26}$$

Subproblem (26) is a MIQP problem and can be solved locally at each EV using off-the-shelf methods, e.g. the branch and bound method.

The $\boldsymbol{\lambda}$-update (20) is performed by the aggregator. Note that it can also be performed locally at each EV. But in the latter case the communication volume is increased, since more data (only $\mathbf{q}^{k+1}$ versus $\mathbf{q}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$) need to be transmitted between the aggregator and the EVs.

For clarity, we summarize the proposed decentralized algorithm in Algorithm 1.

Algorithm 1 Decentralized algorithm based on ADMM for the problem (10)-(14)

Input: base demand data and EVs' local parameters;
Output: optimal charging/discharging scheduling $\mathbf{q}^*$;
1: The aggregator gets base load demand $d(t)$;
2: **for** $k = 0, 1, 2, \ldots,$ **do**
3: The aggregator performs the $\mathbf{p}$-update (22);
4: The aggregator sends the corresponding $\mathbf{p}_i^{k+1}$ to each EV;
5: Each EV performs the $\mathbf{q}$-update by solving (26) using mixed integer program solver;
6: Each EV sends the their $\mathbf{q}_i^{k+1}$ to the aggregator;
7: The aggregator performs the $\boldsymbol{\lambda}$-update (20);
8: **end for**

Remark 1 *Algorithm 1 is decentralized in the sense that the aggregator and EVs communicates in a star network and the computational workload is spatially distributed among them.*

Remark 2 *For the MIQP problem* (10)-(14) *which is NP-hard, its computational complexity usually surges with the increase of the problem size, especially for the integer part. However, with Algorithm 1, the original MIQP problem* (10)-(14) *with nT binary variables is decomposed into n local MIQP problems with T binary variables. As we will show through simulations, Algorithm 1 requires much less CPU time than off-the-shelf commercial solvers when solving* (10)-(14) *in a centralized fashion.*

3.4 Convergence Analysis

It is important to note that ADMM is originally designed for convex optimization problems (1), while problem (17) is a continuous non-convex optimization problem due to the non-convexity of function $G(\mathbf{q})$. Therefore, it is not trivial to prove the convergence of the proposed algorithm.

For Algorithm 1, the primal residual and the dual residual at step k are given by

$$\epsilon^k = \mathbf{p}^k - \mathbf{q}^k \text{ and } \varepsilon^k = \rho(\mathbf{q}^{k+1}-\mathbf{q}^k),$$

respectively. The following theorem shows the convergence result of Algorithm 1.

Theorem 1 *For any $\rho > 0$, the decentralized algorithm 1 converges to the optimal solution of problem* (17) *and*

$$\lim_{k\to\infty}\|\epsilon^k\|_2 = 0 \textit{ and } \lim_{k\to\infty}\|\varepsilon^k\|_2 = 0.$$

Due to the limited space, in this paper we only sketch the general idea for proving Theorem 2 and a detailed proof will be presented in an extended journal version.

Denote the optimal solution to primal problem (17) by $(\mathbf{p}^*, \mathbf{q}^*)$. One can easily verify that $(\mathbf{p}^*, \mathbf{q}^*)$ is also the optimal solution to the following convex optimization problem:

$$\begin{aligned}&\text{minimize} && F(\mathbf{p}) + G_{m^*}(\mathbf{q}), \\ &\text{subject to} && \mathbf{p}-\mathbf{q} = 0.\end{aligned} \tag{27}$$

where $G_{m^*}(\mathbf{q})$ is associated with subset $\Omega(m^*)$ where $\mathbf{q}^* \in \Omega(m^*)$. Note that problem (27) is convex and therefore strong duality holds. The unaugmented Lagrangians of problem (17) and (27) are respectively given by

$$L(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = F(\mathbf{p}) + G(\mathbf{q}) + \boldsymbol{\lambda}^T(\mathbf{p} - \mathbf{q}),$$

$$L_{m^*}(\mathbf{p}, \mathbf{q}, \boldsymbol{\lambda}) = F(\mathbf{p}) + G_{m^*}(\mathbf{q}) + \boldsymbol{\lambda}^T(\mathbf{p} - \mathbf{q}).$$

Let us denote by $\boldsymbol{\lambda}^*$ the optimal Lagrange multiplier of problem (27) and define

$$V^k = (1/\rho)\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|_2^2 + \rho\|\mathbf{q}^k - \mathbf{q}^*\|_2^2$$

as the *Lyapunov function* of Algorithm 1.

Similar to the proof in [13], Theorem 2 can be readily proved if the following three inequalities hold:

$$O^* - O^k \leqslant (\boldsymbol{\lambda}^*(m^k))^T \epsilon^k, \tag{28}$$

$$O^k - O^* \leqslant -(\boldsymbol{\lambda}^k)^T \epsilon^k - (\varepsilon^{k-1})^T(\mathbf{q}^k - \mathbf{q}^* - \epsilon^k), \tag{29}$$

$$V^{k+1} - V^k \leqslant -\rho\|\epsilon^{k+1}\|_2^2 - (1/\rho)\|\varepsilon^k\|_2^2, \tag{30}$$

where

$$O^k = F(\mathbf{p}^k) + G(\mathbf{q}^k),$$

$$O^* = F(\mathbf{p}^*) + G(\mathbf{q}^*)$$

are the objective value at iteration k and the optimal objective value, respectively.

Combining $G(\mathbf{q}^*) = G_{m^*}(\mathbf{q}^*) = 0$, we have

$$L(\mathbf{p}^*, \mathbf{q}^*, \boldsymbol{\lambda}^*) = L_{m^*}(\mathbf{p}^*, \mathbf{q}^*, \boldsymbol{\lambda}^*).$$

Suppose that $\mathbf{q}^k$ belongs to subset $\Omega(m^k)$. From the analysis in Section 3.2, we have

$$L_{m^*}(\mathbf{p}^*, \mathbf{q}^*, \boldsymbol{\lambda}^*) \leqslant L_{m^k}(\mathbf{p}^*(m^k), \mathbf{q}^*(m^k), \boldsymbol{\lambda}^*(m^k)),$$

where $(\mathbf{p}^*(m^k), \mathbf{q}^*(m^k), \boldsymbol{\lambda}^*(m^k))$ is the saddle point of the Lagrangian L_{m^k} of the subproblem associated with $\Omega(m^k)$. From the saddle point property, we further have

$$L_{m^k}(\mathbf{p}^*(m^k), \mathbf{q}^*(m^k), \boldsymbol{\lambda}^*(m^k)) \leqslant L_{m^k}(\mathbf{p}^k, \mathbf{q}^k, \boldsymbol{\lambda}^*(m^k)).$$

Combining $G_{m^k}(\mathbf{q}^k) = G(\mathbf{q}^k) = 0$, we have

$$L(\mathbf{p}^k, \mathbf{q}^k, \boldsymbol{\lambda}^*(m^k)) = L_{m^k}(\mathbf{p}^k, \mathbf{q}^k, \boldsymbol{\lambda}^*(m^k)).$$

From the above inequalities, we have

$$L(\mathbf{p}^*, \mathbf{q}^*, \boldsymbol{\lambda}^*) \leqslant L(\mathbf{p}^k, \mathbf{q}^k, \boldsymbol{\lambda}^*(m^k)). \tag{31}$$

From $\mathbf{p}^* - \mathbf{q}^* = 0$, we have $L(\mathbf{p}^*, \mathbf{q}^*, \boldsymbol{\lambda}^*) = O^*$. From

$$O^k = F(\mathbf{p}^k) + G(\mathbf{q}^k),$$

let us rewrite (31) as

$$O^* \leqslant O^k + (\boldsymbol{\lambda}^*(m^k))^T \epsilon^k,$$

which gives inequality (28).

Similar to the proof of inequality (A.2) in [13], the proof of (29) relies on the following two facts:

1) $\mathbf{p}^k$ minimizes

$$F(\mathbf{p}) + (\boldsymbol{\lambda}^k + \varepsilon^{k-1})^T \mathbf{p},$$

which can be proved by substituting

$$\boldsymbol{\lambda}^{k-1} = \boldsymbol{\lambda}^k - \rho\epsilon^k$$

into $L_\rho(\mathbf{p}, \mathbf{q}^{k-1}, \boldsymbol{\lambda}^{k-1})$ and making some rearrangements. Also, it can be proved in a similar fashion that

2) $\mathbf{q}^k$ minimizes

$$G(\mathbf{q}) + (\boldsymbol{\lambda}^k)^T \mathbf{q}.$$

Therefore, the same trick for proving (A.2) in [13] could be readily applied to prove (29).

The proof of (30) only needs some substituting and rewriting based on (28) and (29), thus it can be proved in the same way to prove (A.1) in [13].

4 Numerical Experiments

In this section we present some simulation results of Algorithm 1, which is implemented in the MATLAB environment on a laptop with Intel Core i7-3610QM processor (8 logic cores) and 8 GB DDR3 memory. The commercial solver Gurobi is used for solving the QP and MIQP subproblems.

4.1 Dealing with 5 EVs

In this case we apply the proposed algorithm to the load shifting problem of 5 EVs, whose parameters are adopted from [12]. We assume a scheduling period of 12 hours with sampling interval $\Delta t = 15$ mins, which gives us 48 time steps. To implement the algorithm, we set $\mathbf{q}^0 = 0$, $\boldsymbol{\lambda}^0 = 0$, and $\rho = 25$. Fig. 1 shows the performance of the load shifting via EVs' charging and discharging. Although the base load curve is irregular and does not satisfy the assumption in [12], peak shaving and valley filling are optimally achieved through the proposed algorithm. The convergence results are given in Fig. 2, which shows that after around 10 ADMM iterations, the proposed algorithm converges with tolerable primal and dual residuals.

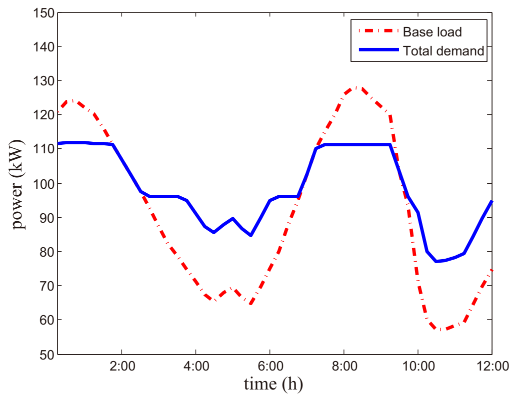

Fig. 1: Optimal load shifting using charging and discharging.

4.2 Dealing with Large Numbers of EVs

We next compare the propose algorithm in terms of convergence speed with Gurobi, which is a powerful commercial optimizer. For comparison, we apply both Algorithm 1

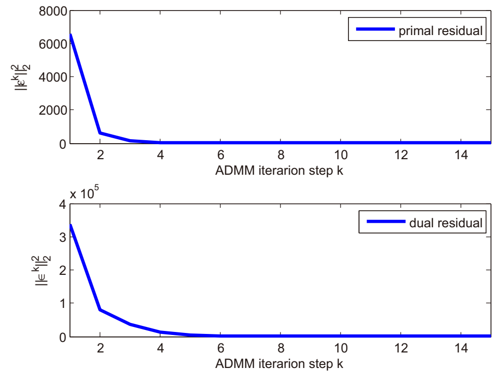

Fig. 2: Convergence results of Algorithm 1.

Table 1: Running times of Algorithm 1 and Gurobi, applied to various numbers of EVs

Time (sec) / Algorithm / Number	Algorithm 1	Gurobi
5	9.96	2.64
15	19.97	3.73
30	28.55	57.88
60	58.14	712.61
100	92.77	3967.36
150	153.88	14682.87
200	244.66	N/A

and the Gurobi optimizer to the cases with larger numbers of EVs. Since Algorithm 1 converges with only a few ADMM steps, for simplicity we set the ADMM steps to be 10. To simulate the implementation on a network of EVs, we use 1 logic core to solve each local subproblem (26), while for the sake of fairness, 8 logic cores are dispatched for Gurobi to solve the original problem (10)-(14). The running times required by Algorithm 1 and Gurobi are given in Table 1. We can see that Gurobi is faster when the number of EVs is small but the running time of Gurobi soars sharply as the number of EV grows. It is not surprising because problem (10)-(14) is a large-scale problem. For instance, in the case with 100 EVs, there are in total 4800 binary variables and 9600 continuous variables. On the contrary, the proposed algorithm is more scalable and much faster when the number of EVs is over 30. The reason is that the number of EVs only affects the complexity of the $\mathbf{p}$-update which is an unconstrained QP problem with $2nT$ variables, while the MIQP subproblems solved in parallel by each EV are always with T binary variables and $2T$ continuous variables. Besides, when $n = 200$, Gurobi returns an error message of "out of memory". Therefore, we conclude that the proposed decentralized algorithm is applicable to a large population of EVs for which centralized algorithms/solvers may fail.

5 Conclusion

This paper studies the optimal load shifting problem via EVs with both G2V and V2G which is formulated as a MIQP problem. A decentralized optimal algorithm based on ADMM is proposed to solve the problem in a decentralized fashion. The proposed algorithm converges much faster than off-the-shelf commercial solvers and also features privacy preservation of EVs. We also present the convergence analysis and show the performance of the proposed algorithm through numerical experiments. In the future work we may consider random behavior of EVs and also extend the proposed algorithm to other EV scheduling problems, e.g., minimization of charging costs.

References

[1] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 371–380, 2010.

[2] W. Kempton and J. Tomić, "Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy," *Journal of power sources*, vol. 144, no. 1, pp. 280–294, 2005.

[3] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control for large populations of plug-in electric vehicles," in *Proc. 49th IEEE Conference on Decision and Control*, 2010, pp. 206–212.

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

[5] Y. Mou, H. Xing, Z. Lin, and M. Fu, "A new approach to distributed charging control for plug-in hybrid electric vehicles," in *Control Conference (CCC), 2014 33rd Chinese*, 2014, pp. 8118–8123.

[6] ——, "Decentralized optimal demand-side management for PHEV charging in a smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.

[7] Y. Mou, H. Xing, M. Fu, and Z. Lin, "Decentralized PWM-based charging control for plug-in electric vehicles," in *Control Conference (ECC), 2015 European*, 2015, pp. 1070–1075.

[8] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

[9] M. E. Khodayar, L. Wu, and M. Shahidehpour, "Hourly coordination of electric vehicle operation and volatile wind power generation in SCUC," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1271–1279, 2012.

[10] A. Y. Saber and G. K. Venayagamoorthy, "Resource scheduling under uncertainty in a smart grid with renewables and plug-in vehicles," *IEEE Systems Journal*, vol. 6, no. 1, pp. 103–109, 2012.

[11] L. Jian, H. Xue, G. Xu, X. Zhu, D. Zhao, and Z. Shao, "Regulated charging of plug-in hybrid electric vehicles for minimizing load variance in household smart microgrid," *IEEE Systems Journal*, vol. 60, no. 8, pp. 3218–3226, 2013.

[12] H. Xing, M. Fu, Z. Lin, and Y. Mou, "Decentralized optimal scheduling for charging and discharging of plug-in electric vehicles in smart grids," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4118–4127, 2016.

[13] 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.