

# 基于JPADMM EV 充电调度

June 7, 2023

## 1 问题描述

在这种模式下, 电动汽车用户将电动汽车连接到充电桩, 充电需求信息上传到聚合器. 每个电动汽车的充电计划将被发送回给电动汽车用户, 充电信息包括到达时间  $t_i^a$ ,

在所提的模型中的一些假设, 首先, 电动汽车用户对充电成本很敏感, 我们引入有序充电的策略, 对于一些不着急充电的用户来说可以选择在电价低时期进行充电, 电价高时不来充电, 从而达到削峰填谷的效果, 第二, 我们假设电动汽车用户的到达时间和离开时间服从以下的泊松分布:

### 1.1 研究现状

文献[1]提出了一个促进稀疏性的充电控制模型. 在该模型中, 通过优化充电时段的数量, 提高用户的满意度. 模型中引入了动态馈线过载约束, 以避免出现不可接受的负荷峰值, 从而保证网络的稳定性. 然后, 针对大多数电网典型的分布式管理方式, 提出了一种基于 *ADMM* 的分布式解决策略. 在求解过程中, 利用拉格朗日对偶将原问题转化为等价对偶问题, 该对偶问题可分解为一组齐次的小尺度子问题. 具体地说, 每个子问题要么有一个闭解, 要么可以用加速对偶梯度法局部求解. 并证明了算法的全局收敛性. 文献[2]创新性地将电动汽车快速充电问题建模为配电网中受耦合馈线容量约束的优化协调问题. 快速充电的需求用总充电时间和在期望时间内充满电的相对趋势来表示. 我们引入一个非凸的充电策略的0-范数来表示总充电时间, 并应用1-范数最小化来近似0-范数最小化的稀疏解. 充电周期越短, 用户的快速充电意愿越强. 优化问题的目标是权衡电动汽车电池退化成本、配电网负荷调节、充电满意度和总充电时间, 而充电时间在单个充电行为中是不可分的. 尽管 *ADMM* 在目标可分、约束耦合的分布式优化中得到了广泛的应用, 但其分散方案不能直接应用于潜在的不可分电动汽车充电协调问题. 为此, 提出了一种基于 *ADMM* 的分层算法, 保证在一定步长参数下收敛到最优策略. 基于以上两篇文献, 本文创新地将用户角度出发, 将充电成本、用户满意度以及充电时间建模为协调优化方法,

## 2 本文主要贡献

在电力系统中, 电网功率是由多个电源和负载之间的相互作用决定的. 这些电源和负载在运行过程中会受到各种因素的影响, 如天气变化、市场需求变化、设备故障等, 这些因素都会导致电网功率的波动. 通过对电网功率波动的统计分析, 可以计算出电网功率的方差, 方差越大说明电网功率波动越大. 如果

电网功率波动过大, 就可能导致电网运行不稳定, 出现电压跌落、频率波动等问题, 甚至可能引发电网崩溃, 造成严重的电力事故。

1) 本文创新地将电动汽车充电优化建模与经济优化调度相关的问题, 将充电总成本、峰谷负荷、用户满意度、

2) 设计了一种新的基于  $ADMM - Jacobi$  的算法来解决具有时间空间耦合不等式约束的不可分优化问题, 并从理论上证明了该算法的最优性和收敛性, 从理论和数值实验上证明了与经典的  $ADMM$  相比收敛速度快

### 3 电动汽车充电模型

考虑电动汽车的个数用  $\mathcal{N} \equiv \{1, \dots, N\}$ , 时间划分为  $\mathcal{T} \equiv \{1, \dots, T\}$  在这里假设电动汽车到达时间  $t_i^a$ , 预期离开时间  $t_i^d$ ,  $SOC$  连接充电桩  $SOC_i^s$ .  $SOC_i^d$  预期的  $SOC$ , 本文将规划范围设为 24 小时, 通过设置时间步长为 15 分钟, 将整个时间范围划分为 96 个周期. 以第  $i$  个 EV 的到达时间和出发时间用方程计算出来:

$$I_i^s = \lceil \frac{t_i^s}{\Delta t} \rceil \quad i = 1, 2, 3, \dots, N$$

$$I_i^d = \lceil \frac{t_i^d}{\Delta t} \rceil \quad i = 1, 2, 3, \dots, N$$

其中,  $I_i^s$  为第  $i$  个 EV 连接到充电桩时的时隙,  $I_i^d$  为第  $i$  个 EV 离开充电桩时的时隙,  $\lceil \frac{t_i^s}{\Delta t} \rceil$  is the smallest integer great than  $\frac{t_i^s}{\Delta t}$  and  $\lceil \frac{t_i^d}{\Delta t} \rceil$  is the smallest integer greater than  $\frac{t_i^d}{\Delta t}$

#### 3.1 约束

##### 3.1.1 EV 充电约束

基于分段线性模型[1], 电池动态描述如下:

$$SOC_{n,t+1} = SOC_{n,t} + \frac{p_{n,t} \times \Delta t \times \eta_n^+}{C_n} - \frac{q_{n,t} \times \Delta t}{\eta_n^- C_n} \quad (3.1)$$

$$SOC_n^{min} \leq SOC_{n,t} \leq SOC_n^{max}, \quad (3.2)$$

其中  $\eta_n^+, \eta_n^- \in (0, 1]$  各自代表充放电能量转换效率. 为了延长电池寿命, 建议  $SOC_{min}$  和  $SOC_{max}$  分别为 15% 和 90%[2].

$$\begin{aligned} \frac{C_n}{\eta_n^+ \Delta t} (SOC_n^{min} - SOC_n^{init}) + \frac{\eta_n^-}{\eta_n^+} \sum_{\tau=1}^t q_{n,\tau} &\leq \sum_{\tau=1}^t p_{n,\tau} \\ &\leq \frac{C_n}{\eta_n^+ \Delta t} (SOC_n^{max} - S_n^{init}) + \frac{\eta_n^-}{\eta_n^+} \sum_{\tau=1}^t q_{n,\tau} \end{aligned} \quad (3.3)$$

设  $p_n^{min}$  和  $p_n^{max}$  分别为 EV $n$  的最小和最大功率, 我们有以下不等式约束:

$$p_n^{min} \leq p_{n,t} \leq p_n^{max}, \quad n \in \mathcal{N}, \quad t \in \mathcal{T} \quad (3.4)$$

通过决定电动汽车在每个时隙中是否充电, 实现了电动汽车的协调充电调度, 第  $i$  个  $EV$  充电状态如下:

$$X_{i,j} = \begin{cases} 0 & \text{if } EV \text{ is being charged,} \\ 1 & \text{otherwise.} \end{cases} \quad (3.5)$$

且

$$(1 - X_{n,t})p_{n,t} = 0 \quad (3.6)$$

则满足以下约束:

$$\sum_{t=1}^T p_{n,t} \geq \frac{C_n}{q_n^+ \Delta t} (SOC_n^{final} - SOC_n^{init}) + \frac{\eta_n^-}{\eta_n^+} \sum_{t=1}^T q_{n,t} \quad (3.7)$$

为了简单表示, 定于如下:

$$\mathcal{P}^{(1)} = \{p_n | p_n \text{ satisfies (3.3) - (3.7)}\} \quad (3.8)$$

设

$$\mathcal{P}^{(1)} \triangleq \mathcal{P}_1^{(1)} \times \mathcal{P}_2^{(1)} \dots \times \mathcal{P}_N^{(1)} \quad (3.9)$$

### 3.2 馈线容量约束

对于分布式电网: 我们分别用  $\mathcal{L} \equiv \{1, \dots, L\}$  和  $\mathcal{M} \equiv \{1, \dots, M\}$  分别表示馈线和节点集合, 用  $\mathcal{N}_l, l \in \mathcal{L}$  表示通过馈线  $l$  供电的电动汽车的集合,  $\mathcal{N}_m, m \in \mathcal{M}$  表示通过根节点  $m$  处供电的电动汽车的集合. 定义连接在节点  $m$  的基本需求表示为  $d_m = (d_{mt}, t \in \mathcal{T})$ , 通过馈线  $l$  的总的基本需求  $D_{lt} \equiv \sum_{m \in \mathcal{M}_l} d_{mt}$ , 其中  $\mathcal{M}_l$  表示通过馈线  $l$  的节点集合, 其功率通过馈线  $l$  进行分配  $\mathbf{d} = (d_t, t \in \mathcal{T})$  表示配电网的总基本需求  $\mathcal{M}_l$ , 则配电网中所有馈线在充电期间的功率容量可表示为电动汽车的充电策略应满足馈线容量的约束:

$$c_{l,t} = C_l - D_{lt},$$

且

$$\sum_{n \in \Pi_n} p_{n,t} \leq c_{lt}.$$

### 3.3 目标函数

$$G_0(P) \triangleq \sum_{n \in \mathcal{N}} \left\{ f_n(p_{nt}) - \omega_n g_n \left( \sum_{t \in \mathcal{T}} p_{nt} \right) + \gamma \|p_n\|_0 \right\} + \delta_n \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2 \quad (3.10)$$

每一项的具体解释如下:

1) 第一项  $f_n(p_{nt})$ : 它代表第  $n$  辆电动汽车的充电总成本,如【】所示,  $f_{p_{nt}}$  的具体表达式:

$$f_n(p_{nt}) = \sum_{t \in \mathcal{T}} \lambda_t p_{nt} \Delta t$$

其中,  $\lambda_t$  为  $t$  时间段的电价,  $p_t$  为电动汽车  $t$  时间段的充电功率,  $\Delta t$  为时间间隔.

2) 第二项:它是关于在充电期间交付的总能量的满意函数, 电动汽车用户更愿意充电时, 直到他们在所需的时间内达到所需的容量  $\Gamma_n$ , 然后, 我们提出以下形式表示用户的满意度:

$$g_n(\sum_{t \in \mathcal{T}} p_{nt}) = -(\sum_{t \in \mathcal{T}} p_{nt} - \Gamma_n)^2.$$

3)第三项  $\|p_n\|$ :基数最小化  $\|p_n\|_0$  在稀疏解的建模中起着重要作用, 然而  $\|p_n\|_0$  的计算仍然是一个挑战, 因为  $\|p_n\|_0$  是一个非凸函数, 直接优化这个函数甚至在线性约束上被证明是一个  $NP$  困难问题. 无法通过常规方法求解最小  $\|p_n\|_0$  问题和最小  $\|p_n\|_1$  范数问题在一定条件下是等价的, 即可以通过求解最小  $\|p_n\|_1$  问题来得到最小  $\|p_n\|_0$  问题的解. 求解最小  $\|p_n\|_1$  问题因为是凸优化问题相对来说简单得多.

4) 第四项  $\|\mathbf{p} - \bar{\mathbf{p}}\|_2^2$ :表示负荷调节, 其中  $\bar{\mathbf{p}}$  表示平均功率,  
这里具体怎么描述  $\|\mathbf{p} - \bar{\mathbf{p}}\|_2^2 = \sum_{t \in \mathcal{T}} \left( \sqrt{(\sum_{n \in \mathcal{N}} p_{n1} - \bar{p})^2 + (\sum_{n \in \mathcal{N}} p_{n2} - \bar{p})^2 + \cdots + (\sum_{n \in \mathcal{N}} p_{nt} - \bar{p})^2} \right).$

插入  $l_1$  凸逼近, 我们将(2.1)重新建模为以下稀疏促进充电控制模型

$$G_0(P) \triangleq \sum_{n \in \mathcal{N}} \left\{ f_n(p_{nt}) - \omega_n g_n(\sum_{t \in \mathcal{T}} p_{nt}) + \gamma \|p_n\|_1 \right\} + \delta_n \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2 \quad (3.11)$$

## 4 雅可比邻近ADMM求解模型

ADMM方法已经成为大规模结构化的有力工具, 值得注意的是, ADMM可以解耦空间耦合约束, 并将解耦问题分解为许多个小问题, 其中每个子问题分解都很容易解决. 在[1] 中提出了并行分布式计算 *Jacobi - PADMM*. 近端项的灵活使用以不同的方法解决子问题, 提高了标准ADMM的收敛率.

值得注意的是, 在模型(2.1)中的第二项耦合了所有的变量  $p_{n,t}, n \in \mathcal{N}$ , 为了解耦此项, 根据文献引入辅助变量

$$p_{N+1} = (p_{N+1,1}, \dots, p_{N+1,1})^T \in \mathbb{R}^T$$

其中  $p_{N+1,t} = \sum_{t \in \mathcal{T}} p_{nt} - \Gamma_n, t \in \mathcal{T}$ , 因此模型(2.1)等价表示如下:

$$\min_{\{u\}} G(P) \quad (4.1)$$

$$s.t. \quad \sum_{n \in \Pi_n} p_{n,t} \leq c_{lt} \quad (4.2)$$

$$p_{N+1,t} = \sum_{t \in \mathcal{T}} p_{nt} - \Gamma_n, t \in \mathcal{T} \quad (4.3)$$

优化问题重写为如下形式:

$$\begin{aligned} \min_p \quad & G(P) + \mathcal{I}_1(P) + \mathcal{I}_2(Y) \\ \text{s.t.} \quad & P - Y = 0 \end{aligned}$$

定义

$$\mathcal{P}_t^{(2)} = \{P_t | P_t \text{ satisfies (3.2) and (3.3)}\}$$

且

$$\mathcal{P}_t^{(2)} = \mathcal{P}_1^{(2)} \times \dots \times \mathcal{P}_t^{(2)} \times \dots \times \mathcal{P}_T^{(2)}$$

其中约束  $P_t^{(2)}$  是一个全局约束, 在空间上耦合所有EV的充电时间表.  $P_t^{(2)}$  是闭的、非空凸集的指示函数是恰当的、闭的和凸的, 所以为了使用 *Jacobi - PADMM* 我们更进一步介绍以下凸集  $P^{(i)}, i = 1, 2$ , 如下,

$$\mathcal{I}_i(P) = \begin{cases} 0 & \text{if } P \in \mathcal{P}^{(i)}, \\ +\infty & \text{otherwise.} \end{cases} \quad (4.4)$$

很清楚地,  $\mathcal{I}_i(P), i = 1, 2$  是凸函数.

现在我们可以很容易地将 (3.1) – (3.3) 所描述的问题写成标准的ADMM形式, 如下所示

$$\begin{aligned} \min_{\{u\}} \quad & G(P) + \mathcal{I}_1(P) + \mathcal{I}_2(Y) \\ \text{s.t.} \quad & P - Y = 0. \end{aligned} \quad (4.5)$$

定义增广拉格朗日函数如下:

$$\mathcal{L}_\rho = G(P) + \mathcal{I}_1(P) + \mathcal{I}_2(X) + \mu^T(P - X) + \frac{\rho}{2}\|P - X\|_\Phi^2$$

其中  $\rho > 0$  是惩罚参数,  $\mu$  称为对偶变量或者拉格朗日乘子, 变量  $p, x, u$  的更新

$$\begin{aligned} P^{k+1} &\triangleq \underset{Y}{\operatorname{argmin}} \mathcal{L}_\rho(P^k, Y, \mu^k) + \frac{1}{2}\|P - P^k\|_\Phi^2 \\ Y^{k+1} &\triangleq \underset{Y}{\operatorname{argmin}} \mathcal{L}_\rho(P, Y^k, \mu^k) + \frac{1}{2}\|Y - Y^k\|_\Psi^2 \\ \mu^{k+1} &\triangleq \mu^k - \zeta \rho(P^{k+1} - P^k) \end{aligned}$$

**Definition 4.1.** 如果选择  $\delta_i > 0$  使得参数  $\rho, \zeta, \Phi, \Psi$  满足以下的情况:

$$\begin{cases} \Phi \succ \rho(\frac{1}{\delta_1} - 1)I, \\ \Psi \succ \rho(\frac{1}{\delta_2} - 1)I, \\ \delta_1 + \delta_2 < 2 - \zeta, \end{cases}$$

选择参数  $\rho, \zeta, \Phi, \Psi$  的方法多种多样, 本文采用一个简单的选择 【】  $\Phi = \phi I, \Psi = \psi I$  使得

$$\begin{cases} \phi \succ \rho(\frac{1}{\delta_1} - 1), \\ \psi \succ \rho(\frac{1}{\delta_2} - 1), \\ \delta_1 + \delta_2 < 2 - \zeta, \end{cases}$$

## 5 计算P

本节给出了在计算  $p$  的详细信息, 从指示函数的定义中可以容易地看出,  $p^{k+1}$  的实际优化策略如下:

$$P^{k+1} = \underset{P}{\operatorname{argmin}} G(P) + \frac{\rho}{2} \|P - Y + \frac{\mu}{\rho}\|_F^2 + \frac{1}{2} \|P - P^k\|_\Phi^2$$

当  $n = N + 1$  时

$$p_{N+1} = \frac{\rho}{\rho - 2\omega_n} \left( \frac{\mu_{N+1}^k}{\rho} - y^{k+1} \right)$$

对于每一个  $n$

$$p_n = \underset{p}{\operatorname{argmin}} G_n(p) + \frac{\rho}{2} \|p_n - y_n + \frac{\mu_n}{\rho}\|_F^2 + \frac{1}{2} (p_n - p_n^k)^T \phi(p_n - p_n^k)$$

## 6 计算Y

重新更改本节详细介绍()中变量  $x$  的更新, 等价于求解以下凸优化问题的解:

$$\begin{aligned} Y^{k+1} &= \underset{Y}{\operatorname{argmin}} \mathcal{I}_2(Y) + \frac{\rho}{2} \|P - Y + \frac{\mu}{\rho}\|_F^2 + \frac{1}{2} \|Y - Y^k\|_\Psi^2 \\ &= \underset{Y}{\operatorname{argmin}} \frac{\rho}{2} \|Y - (P^k + \frac{\mu^k}{\rho})\|_F^2 + \frac{1}{2} \|Y - Y^k\|_\Psi^2 \end{aligned}$$

对于  $x - \text{update}$  的更新, 可以由以下分散方式计算变量

$$y_n^{k+1} = \underset{y}{\operatorname{argmin}} \frac{\rho}{2} \{y_t^T y_t - 2(p_t^k + \frac{\mu_t^k}{\rho})^T y_t\} + \frac{1}{2} (y_t - y_t^k)^T \psi(y_t - y_t^k)$$

## 7 算法

算法收敛的表征如下:

$$r^{k+1} = \|P^{k+1} - Y^{k+1}\|_F$$

$$s^{k+1} = \|\rho(Y^{k+1} - Y^k)\|_F$$

一个合理的终止标准是原始残差和对偶残差要足够小, 如下:

$$\|r^k\|_2 \leq \varepsilon^{pri} \quad \|s^k\|_2 \leq \varepsilon^{dual}$$

where  $\varepsilon^{pri} > 0$  和  $\varepsilon^{dual} > 0$  are feasibility tolerances for the primal and dual feasibility conditions () and (). These tolerances can be chosen using an absolute and relative criterion, such as

$$\begin{aligned} \varepsilon^{pri} &= \sqrt{N} \varepsilon^{abs} + \varepsilon^{rel} \max\{\|p\|_2, \|x\|_2\}, \\ \varepsilon^{dual} &= \sqrt{N} \varepsilon^{abs} + \varepsilon^{rel} \|\mu^k\|_2 \end{aligned}$$

where  $\varepsilon^{abs} > 0$  is an absolute tolerance and  $\varepsilon^{rel} > 0$  is a relative tolerance. A reasonable value for the relative stopping criterion might be  $\varepsilon^{-3}$  or  $\varepsilon^{-4}$ . A pseudo-code for the proposed Prox-JADMM is then give in Algorithm 1.