基于JPADMM EV 充电调度

July 8, 2023

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，本文创新地将从用户角度出发，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

首先我们提出了一种新的分布式雅可比近端交替方向乘子算法来解决大规模电动汽车充电调度问题, 该算法可以并行更新变量,

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分布式 $Jacobi-PADMM$ 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. 保护每位电动汽车车主的隐私.

3 电动汽车充电模型

引入记号: 设 $\boldsymbol{p}\in\mathbb{R}^n, \|\boldsymbol{p}\|_2=(\sum_{i=1}^n p_i^2)^{1/2}$ 表示标准的欧式范数, $\|\boldsymbol{p}\|_0=\sharp(i|p_i\neq 0)$ 表示向量 $\boldsymbol{p}$ 中非零元的个数.

Table 1: 符号表示

$\mathcal{N}$	电动汽车个数集合:$\mathcal{N} = \{1, ..., N\}$
$\mathcal{T}$	离散时间集合:$\mathcal{T} = \{1, ..., T\}$
$\mathcal{L}$	馈线集合:$\mathcal{L} = \{1, ..., L\}$
$p_{n,t}$	在 t 时刻第 n 辆电动汽车的充电速率
$\mathcal{N}_l$	通过馈线 l 供电的电动汽车的集合
$\mathcal{N}_m$	在节点 m 充电的电动汽车的集合
C_n	电动汽车的电车容量
d_t	在 t 时刻总基本需求

在本节中建立一个协调调度充电模型来控制电动汽车充电, 令 $\boldsymbol{p}_n = (p_{n,1}, ..., p_{n,T}) \in \mathbb{R}^T, \boldsymbol{p} = (\boldsymbol{p}_1, ..., \boldsymbol{p}_N)$

3.1 EV充电约束

基于分段线性模型[1], 电池动态描述如下：

$$SOC_{n,t+1} = SOC_{n,t} + \frac{p_{n,t} \times \triangle t \times \eta_n^+}{C_n} \tag{3.1}$$

$$SOC_n^{min} \leq SOC_{n,t} \leq SOC_n^{max}, \tag{3.2}$$

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].

$$\frac{C_n}{\eta_n^+ \Delta t}(SOC_n^{min} - SOC_n^{init}) \leq \sum_{t=1}^{T} p_{n,t} \leq \frac{C_n}{\eta_n^+ \Delta t}(SOC_n^{max} - S_n^{init}) \tag{3.3}$$

其中 SOC^{init} 是 $EV\ n$ 的初始 SOC 值.设 p_n^{min} 和 p_n^{max} 分别为 EVn 的最小和最大功率, 我们有以下不等式约束:

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

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:

$$X_{n,t} = \begin{cases} 1 & if\ EV\ is\ being\ charged, \\ 0 & otherwise. \end{cases} \tag{3.5}$$

约束满足以下形式:

$$(1 - X_{n,t})p_{n,t} = 0,\ n \in \mathcal{N},\ t \in \mathcal{T}$$

在实践中, 一般采用以下保守约束,即每个 EVn 在有限时间范围 T 结束时至少达到最终SOC 值, 表示为 SOC_n^{final},

$$\sum_{t=1}^{T} p_{n,t} \geq \frac{C_n}{q_n^+ \Delta t}(SOC_n^{final} - SOC_n^{init}). \tag{3.6}$$

为了简化表示, 定义 $\mathcal{P}_n^{(1)}$ 如下

$$\mathcal{P}_n^{(1)} = \{\{p_{n,t}\}_{n=1}^N | \{p_{n,t}\}_{n=1}^N \ satisfies(3.3)-(3.6)\} \tag{3.7}$$

设

$$\mathcal{P}^{(1)} \triangleq \mathcal{P}_1^{(1)} \times \mathcal{P}_2^{(1)} \cdots \times \mathcal{P}_N^{(1)} \tag{3.8}$$

3.2 馈线容量约束

对于分布式电网: 我们分别用 $\mathcal{L} \equiv \{1,...,L\}$ 和 $\mathcal{M} \equiv \{1,...,M\}$ 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 $d_m = (d_{mt}, t \in \mathcal{L})$, 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:

$$c_{l,t} = \beta_l - D_{lt},$$

且

$$\sum_{n \in \Pi_n} p_{n,t} \leq c_{lt}. \tag{3.9}$$

3.3 目标函数

$$G_0(\boldsymbol{p}) \triangleq \sum_{n \in \mathcal{N}} \left\{ f_n(p_{nt}) + g_n(\sum_{t \in \mathcal{N}} p_{nt}) + \|\boldsymbol{p}_n\|_0 \right\} + \sum_{t \in \mathcal{T}} h_n(\sum_{t \in \mathcal{T}} p_{nt}) \tag{3.10}$$

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,如文献[1]所示, $f_n(p_{nt})$ 的具体表达式:

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

其中, E_t 为 t 时间段的电价, p_{nt} 为电动汽车 t 时间段的充电功率, Δt 为时间间隔.

2) 第二项$g_n(\sum\limits_{t \in \mathcal{T}} p_{nt})$: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 Γ_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 $h_n(\sum_{t\in\mathcal{T}} p_{nt})$:表示负荷调节,也表示配电网对总电力需求的功率损耗,δ_n 为加权参数,反应了快速充电行为对电网的影响趋势,其中 $\overline{\mathbf{p}}$ 表示平均功率,具体表示为$\overline{\mathbf{p}} = \frac{1}{N}\sum\limits_{n\in\mathcal{N}}\sum\limits_{t\in\mathcal{T}} p_{nt}$

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

$$G(\boldsymbol{p}) \triangleq \sum_{n\in\mathcal{N}}\left\{f_n(p_{nt}) - \omega_n g_n(\sum_{t\in\mathcal{T}} p_{nt}) + \gamma\|p_n\|_1\right\} + \delta_n \sum_{t\in\mathcal{T}} h_n(\sum_{t\in\mathcal{T}} p_{nt}) \quad (3.11)$$

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(11)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

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

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

$$\min_{\{u\}} \quad G(\boldsymbol{p}) \quad (4.1)$$

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

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

为了使用 $Jacobi-PADMM$,定义

$$\mathcal{P}_t^{(2)} = \{P_t | P_t\ satisfies\ (4.2)\ and\ (4.3)\} \quad (4.4)$$

且

$$\mathcal{P}_t^{(2)} = \mathcal{P}_1^{(2)} \times \cdots \times \mathcal{P}_t^{(2)} \times \cdots \times \mathcal{P}_T^{(2)} \quad (4.5)$$

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 $Jacobi-PADMM$ 我们更一步介绍以下凸集 $P^{(i)}, i=1,2$, 如下，

$$\mathcal{I}_i(\boldsymbol{p}) = \begin{cases} 0 & if \quad \boldsymbol{p}\in\mathcal{P}^{(i)}, \\ +\infty & otherwise. \end{cases} \quad (4.6)$$

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

现在我们可以很容易地将 (4.1) – (4.3) 所描述的问题写成标准的ADMM 形式，如下所示

$$\begin{aligned} &\min_{\{u\}} G(P) + \mathcal{I}_1(P) + \mathcal{I}_2(Y) \\ &s.t \quad P - Y = 0. \end{aligned} \tag{4.7}$$

这里定义:

$$\begin{aligned} f(P) &\triangleq G(p) + \mathcal{I}_1(P) \\ g(Y) &\triangleq \mathcal{I}_2(Y) \end{aligned}$$

定义增广拉格朗日函数如下:

$$\begin{aligned} \mathcal{L}_\rho(P, Y, \mu) &= f(P) + g(Y) + \Lambda^T(P - Y) + \frac{\rho}{2}\|P - Y\|_F^2 \\ &= f(P) + g(Y) + \frac{\rho}{2}\|P - Y + \boldsymbol{u}\|_F^2 \end{aligned} \tag{4.8}$$

其中 $\rho > 0$ 是惩罚参数,Λ 称为对偶变量或者拉格朗日乘子,$\boldsymbol{u}$称为缩放后的对偶变量,i.e.$\boldsymbol{u} = (1/\rho)\Lambda$ 变量 p, x, u 的更新

$$P^{k+1} \triangleq \underset{P}{argmin}\{\mathcal{L}_\rho(P, Y, \boldsymbol{u}^k) + \frac{1}{2}\|P - P^k\|_\Phi^2\} \tag{4.9}$$

$$Y^{k+1} \triangleq \underset{Y}{argmin}\{\mathcal{L}_\rho(P^k, Y, \boldsymbol{u}^k) + \frac{1}{2}\|Y - Y^k\|_\Psi^2\} \tag{4.10}$$

$$\mu^{k+1} \triangleq \boldsymbol{u}^k - \zeta\rho(P^{k+1} - Y^{k+1}) \tag{4.11}$$

Definition 4.1. 如果选择 $\delta_i > 0$ 使得参数 ρ, ζ, Φ, Ψ 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 ρ, ζ, Φ, Ψ 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:

$$P^{k+1} = \underset{P \in \mathcal{P}^{(1)}}{argmin}\{G(P) + \frac{\rho}{2}\|P - Y + \boldsymbol{u}\|_F^2 + \frac{\phi}{2}(P - P^k)^T(P - P^k)\} \tag{5.1}$$

当 $n = N + 1$ 时

$$p_{N+1}^{k+1} = \frac{\rho}{\rho - 2\omega_n}(\frac{\mu_{N+1}^k}{\rho} - y_{N+1}^k) \tag{5.2}$$

对于每一个 n=1,...,N

$$p_n^{k+1} = \underset{Y\in\mathcal{P}^{(1)}}{argmin}\ \{G_n(p) + \frac{\rho}{2}\|p_n - y_n + \frac{\mu_n}{\rho}\|_F^2 + \frac{\psi}{2}(p_n - p_n^k)^T(p_n - p_n^k)\} \quad (5.3)$$

6 计算Y

本节详细介绍(4.10)中变量Y的更新, 等价于求解以下凸优化问题的解:

$$\begin{aligned} Y^{k+1} &= \underset{Y}{argmin}\ \mathcal{I}_2(Y) + \frac{\rho}{2}\|P^k - Y + \boldsymbol{u}\|_F^2 + \frac{\psi}{2}\|Y - Y^k\|_F^2 \quad (6.1)\\ &= \underset{Y\in\mathcal{P}^{(2)}}{argmin}\ \frac{\rho}{2}\|Y - (P^k + \boldsymbol{u})\|_F^2 + \frac{\psi}{2}\|Y - Y^k\|_F^2 \end{aligned}$$

对于 $y - update$ 的更新, 可以由以下分散方式计算变量

$$\underset{y}{min}\ \frac{\rho}{2}\{y_t^T y_t - 2(p_t^k + \frac{\mu_t^k}{\rho})y_t\} + \frac{\psi}{2}(y_t - y_t^k)^T(y_t - y_t^k) \quad (6.2)$$

子问题(6.2)是一个CQP问题。为了简化符号, 我们引入一个 $(L\times N)$ 矩阵 A^t 来表示时刻t的馈线用户关联矩阵及其在$a_{l,n}^t$处的分量

$$a_{l,n}^t = \begin{cases} 1 & if\ PEV\ n\ uses\ feeder\ l\ at\ time\ t \\ 0 & otherwise \end{cases} \quad (6.3)$$

设 $A^t = (a_1^t, ..., a_N^t)$ a_n^t 是 A^t 的第n列向量, 表示EVn使用的相应馈线) $a_{N+1}^t = 0 \in \mathcal{R}^l$, $c^t = (c_{1,t}, ...c_{l,t}), b_n - 1, n \in \mathcal{N}, b_{N+1} = 1$, 那么子问题重写为

$$\begin{aligned} min \quad & \sum_{n=1}^{N+1} y_{n,t}^2 - 2(x_{n,t}^k + U_{n,t}^k)y_{n,t} \quad (6.4)\\ s.t. \quad & \sum_{n=1}^{N+1} a_n^t y_{n,t} \le c^t \quad (6.5)\\ & \sum_{n=1}^{N+1} b_n y_{n,t} = d_t \quad (6.6) \end{aligned}$$

引入子问题 (6.4)-(6.6) 的拉格朗日函数

$$\begin{aligned} \mathcal{L}^t(y_t;\lambda_t,\mu_t) &= \sum_{n=1}^{N+1}[y_{n,t}^2 - 2(x_{n,t} + U_{n,t}^k)y_{n,t}] + \lambda_t^T[a_n^t y_{n,t} - c^t/(N+1)] \quad (6.7)\\ &+ \mu_t[b_n y_{n,t} - d_t/(N+1)] \quad (6.8)\\ &:= \sum_{n=1}^{N+1}\mathcal{L}_{n=1}^t(y_t;\lambda_t,\mu_t) \quad (6.9) \end{aligned}$$

(6.7)的对偶问题为

$$\underset{\lambda_t\succeq 0,\mu_t}{max}\left\{D^t(\lambda_t,\mu_t) = \sum_{n=1}^{N+1}\underset{y_t}{min}\,\mathcal{L}^t(y_t;\lambda_t,\mu_t)\right\} \quad (6.10)$$

根据(6.10), 目标函数被重写为

$$D^t(\lambda_t, \mu_t) = \sum_{n=1}^{N+1} \min_{y_{n,t}} \mathcal{L}_n^t(y_t; \lambda_t, \mu_t) \tag{6.11}$$

第二种思路:
由于(6.1)中的函数是解耦的, 因此计算其变量 y_i^k i=1,...,N. 的优化问题为

$$\min_{y_t \in \mathcal{P}_t^{(2)}} \quad [(\rho+\psi)y_t^2 - 2(\rho(p_i^k + \boldsymbol{u}_i^k) + \psi y_i^k)y_t] \tag{6.12}$$

$$s.t. \quad \hat{p}_{nt}^{min} \leq y_t \leq \hat{p}_{nt}^{max} \tag{6.13}$$

它的解析解很容易得:

$$y_t^{k+1} = \begin{cases} \hat{p}_{nt}^{min} & \overline{y}_t^{k+1} < \hat{p}_{nt}^{min} \\ \overline{y}_t^{k+1} & \hat{p}_{nt}^{min} \leq \overline{y}_t^{k+1} \leq \hat{p}_{nt}^{max} \\ \hat{p}_{nt}^{max} & \overline{y}_t^{k+1} < \hat{p}_{nt}^{max} \end{cases} \tag{6.14}$$

7 算法

算法收敛的表证如下：

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

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

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:

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

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

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

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 10^{-3} or 10^{-4}. A pseudocode for the proposed Prox-JADMM is then give in Algorithm 1.

Algorithm 1 algorithm of JPADMM

Input: tolerance error ϵ^{pri}, ϵ^{dual}, penalty parameter ρ
$P^0, Y^0,\ \mu^0$

Output: P, Y

while $\|P^{k+1} - Y^{k+1}\|_F > \epsilon^{pri}$ or $\|\rho(Y^{k+1} - Y^k)\|_F > \epsilon^{dual}$ **do**
P-update:
for $n = 1, ..., N,$
计算: $p_n^{k+1} = \underset{P \in \mathcal{P}^{(1)}}{argmin}\ G_n(p) + \frac{\rho}{2}\|p_n - y_n + \frac{\mu_n}{\rho}\|_F^2 + \frac{\psi}{2}(p_n - p_n^k)^T(p_n - p_n^k);$
for $n = N + 1;$
计算: $p_{N+1}^{k+1} = \frac{\rho}{\rho - 2\omega_n}(\frac{\mu_{N+1}^k}{\rho} - y_{N+1}^k);$
Y-update:
for $t = 1, ..., T;$
Y的计算根据子算法(算法2);
μ-update:
for n=1,...,N+1, t=1,...,T.
$\mu_{n,t}^{k+1} = \mu_{n,t}^k + \gamma\rho(P_{n,t}^{k+1} - Y_{n,t}^{k+1})$
更新 $r^{k+1} = \|P^{k+1} - Y^{k+1}\|_F$
更新 $s^{k+1} = \|\rho(Y^{k+1} - Y^{k+1})\|_F$
更新 $k := k + 1$
end while
return EEEEE

8 weiyifan note

$$P^{k+1} \triangleq \underset{P}{argmin}\{\mathcal{L}_\rho(P,Y,\mu^k)+\frac{1}{2}\|P-P^k\|_\Phi^2\} \tag{20}$$

$$Y^{k+1} \triangleq \underset{Y}{argmin}\{\mathcal{L}_\rho(P^k,Y,\mu^k)+\frac{1}{2}\|Y-Y^k\|_\Psi^2\} \tag{21}$$

$$\mu^{k+1} \triangleq \mu^k-\zeta\rho(P^{k+1}-Y^{k+1}) \tag{22}$$

8.1 优化问题

目标函数：

$$\begin{aligned}G(\{p_{n,t}\}_{n=1,t=1}^{N,T}) =& \sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}}E_t p_{nt}\Delta t+\sum_{n\in\mathcal{N}}(\sum_{t\in\mathcal{T}}p_{n,t}-\Gamma_n)^2+\sum_{n\in\mathcal{N}}\|\{p_{n,t}\}_{t=1}^T\|_0\\ &+\sum_{t\in\mathcal{T}}(\sum_{n\in\mathcal{N}}p_{n,t}-(\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}p_{n,t}))^2\end{aligned}$$

8.2 转化为多块ADMM

注意到n取值往往较大，如果所有车辆的优化函数可以并行运算，将极大提高优化效率。注意到：

$$\begin{aligned}G(\{p_{n,t}\}_{n=1,t=1}^{N,T}) =& \sum_{t\in\mathcal{T}}E_t p_{1,t}\Delta t+(\sum_{t\in\mathcal{T}}p_{1,t}-\Gamma_1)^2+\|\{p_{1,t}\}_{t=1}^T\|_0\\ &+\cdots\\ &+\sum_{t\in\mathcal{T}}E_t p_{N,t}\Delta t+(\sum_{t\in\mathcal{T}}p_{N,t}-\Gamma_N)^2+\|\{p_{N,t}\}_{t=1}^T\|_0\\ &+\sum_{t\in\mathcal{T}}(\sum_{n\in\mathcal{N}}p_{n,t}-(\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}p_{n,t}))^2\end{aligned}$$

为了简洁，引入符号：
E_t 为 t 时间段的电价,
$p_{N+1,t} := \sum_{n=1}^{N} p_{n,t}$
Γ_n 为电第n辆电动汽车的充电需求,
Δt 为时间间隔,

$$\mathbf{E}(T\times 1)=\begin{pmatrix}E_1\\ \vdots\\ E_t\\ \vdots\\ E_T\end{pmatrix},\mathbf{p}_n(T\times 1)=\begin{pmatrix}p_{n,1}\\ \vdots\\ p_{n,t}\\ \vdots\\ p_{n,T}\end{pmatrix},\mathbf{p}_{N+1}(T\times 1)=\begin{pmatrix}p_{N+1,1}\\ \vdots\\ p_{N+1,t}\\ \vdots\\ p_{N+1,T}\end{pmatrix} \tag{8.1}$$

$$f_n(\mathbf{p}_n) := \underbrace{\mathbf{E}^T\mathbf{p}_n\Delta t}_{\text{第n辆车的充电成本}} + \underbrace{(\|\mathbf{p}_n\|_1 - \Gamma_n)^2}_{\text{满足充电需求}} + \underbrace{\|\mathbf{p}_n\|_0}_{\text{充电时间尽可能少}} + \mathbf{I}_{\mathcal{X}_n}(\mathbf{p}_n), \quad \forall 1 \le n \le N \tag{8.2}$$

$$f_{N+1}(\mathbf{p}_{N+1}) := \underbrace{\|\mathbf{p}_{N+1} - \|\mathbf{p}_{N+1}\|_1/N\|_2^2}_{\text{削峰填谷}} \tag{8.3}$$

则上述问题可以等价表述为：

$$\begin{array}{rl} \text{minmize} & f_1(\mathbf{p}_1) + \cdots + f_N(\mathbf{p}_N) + f_{N+1}(\mathbf{p}_{N+1}) \\ \text{subject to} & \mathbf{p}_1 + \cdots + \mathbf{p}_N - \mathbf{p}_{N+1} = \mathbf{0} \end{array}$$

8.3 Jacobi-Proximal ADMM算法

Algorithm 2 Jacobi-Proximal ADMM

Input: $\mathbf{p}_n^0, A_n(n = 1, 2, ..., N), \lambda^0, \ \varepsilon^{pri}, \ \rho, \gamma$
Output: $\mathbf{p}_n^k(n = 1, 2, ..., N)$

1: **for** k=0,1,... **do**
2: Update $\mathbf{p}_n$ for $n = 1, ..., N$ in parallel by:
3: $\mathbf{p}_n^{k+1} = \text{argmin} \quad f_n(\mathbf{p}_n) + \frac{\rho}{2}\|A_n\mathbf{p}_n + \sum_{j\neq i} A_j\mathbf{p}_j^k - \underbrace{c}_{c=0} - \frac{\lambda^k}{\rho}\|_2^2$
4: $\quad + \frac{1}{2}\|\mathbf{p}_n - \mathbf{p}_n^k\|_{P_i}^2$
5: Update $\lambda^{k+1} = \lambda^k - \gamma\rho(\sum_{i=1}^N A_i\mathbf{p}_n^{k+1} - c)$
6: Update $r^{k+1} = \sum_{n=1}^{N+1} \|\mathbf{p}_n^{k+1} - \mathbf{p}_n^k\|_F$
7: **if** $r^{k+1} \le \varepsilon^{pri}$ **then**
8: Break.
9: **end if**
10: **end for**

Algorithm 3 Solve argmin

Input: $\mathbf{p}_n^k(n = 1, 2, ..., N), \lambda^k$
Output: $\mathbf{p}_n^k(n = 1, 2, ..., N)$

1: 使用matlab 优化工具箱，求解此带约束的二次优化函数，即可