ADMM-Based Decentralized Electric Vehicle Charging with Trip Duration Limits

Gaoqi He\textsuperscript{1,2}, Zhifu Chai\textsuperscript{1}, Xingjian Lu\textsuperscript{2}, Fanxin Kong\textsuperscript{3}, Bing Sheng\textsuperscript{4}

\textsuperscript{1}East China Normal University, China
\textsuperscript{2}East China University of Science and Technology, China
\textsuperscript{3}Syracuse University, US
\textsuperscript{4}Shanghai Jiaotong University, China
fkong03@syr.edu
Outline

- Background
- Problem Formulation
- Algorithm Design
- Experiments
- Conclusion
**Background**

- **Charging coordination** is essential for EV owners to reduce charging fee and charging duration.
- Most existing works is confined to finding CSs to optimize the two parts individually.
Background

Our Motivation: exploring how to jointly minimize charging duration and charging fee for trips with scheduled deadlines.
Contributions

- A novel charging coordination problem that jointly optimize charging duration and charging fee, which is formulated as a 0-1 Integer Linear Programming problem.

- We propose a novel distributed EV charging algorithm based on ADMM, which gradually converges to the optimal solution.

- The extensive experimental studies based upon real-life transport network data are conducted.
Outline

- Background
- Problem Formulation
- Algorithm Design
- Experiments
- Conclusion
Basic definitions

**GC**: global coordinator (receive requests and scheduling)

**CS**: charging station (with a CS Agent to collect necessary information)

**Four zones**: industrial, commercial, residential and office zone

**Regional time-of-use price model**: electricity price varies in different zones and different time

**Transportation network** as a graph $G = (E, L)$

**Request** $REQ(Cap, SOC_{\text{cur}}, \alpha, l_{req}, l_{des}, T_{\text{deadline}})$. 

- **Battery capacity**
- **Unit energy consumption**
- **Destination location**
- **Deadline**
**Typical procedure**

**stage1:** EVs send requests, GC forwards requests to all CS Agents, all CS Agents and GC coordinate to get the optimal CS.

**Stage2:** EV heads to the CS, updates its information periodically.

**Stage3:** Once arrived, the EV joins the waiting queue. Or **stage4** (directly go to charging).

**Stage5:** After charging, the EV heads on to the destination.

**Stage6:** Finally, the EV reaches the destination.
Problem formulation

Zero-one linear programming problem

\[
\begin{align*}
\text{min} & \quad \sum_{j=1}^{M} \gamma_j \cdot (\theta \cdot t_j^{\text{trip}} + (1 - \theta) \cdot c_j) \\
\text{s.t.} & \quad \sum_{j=1}^{M} \gamma_j = 1, \gamma_j \in \{0, 1\} \\
& \quad 0 \leq \theta \leq 1 \\
& \quad t_j^{\text{trip}} \cdot \gamma_j \leq T^{\text{deadline}}
\end{align*}
\]
Outline

- Background
- Problem Formulation
- Algorithm Design
- Experiments
- Conclusion
**ADMM Framework**

- **ADMM** is well suited to distributed convex optimization and especially the large-scale problems in statistics, machine learning and related areas.

\[
\begin{align*}
\min & \quad f(x) + g(z) \\
\text{s.t.} & \quad Ax + Bz = c \\
& \quad x \in C_1, z \in C_2
\end{align*}
\]

- **ADMM** solves the problem with the iterations:

\[
\begin{align*}
x^{k+1} &= \arg\min_{x \in C_1} L_\rho(x, z_k, \lambda_k) \\
z^{k+1} &= \arg\min_{z \in C_2} L_\rho(x_k, z, \lambda_k) \\
\lambda_{k+1} &= \lambda + \rho (Ax + By - c)
\end{align*}
\]
Problem Reformulation

- Introduce a new set of **auxiliary variables** \( z_j \), Reformulate the optimization to use ADMM

\[
\begin{align*}
\min & \quad \sum_{j=1}^{M} (\gamma_j \cdot \theta \cdot t_j^{trip} + z_j \cdot (1 - \theta) \cdot c_j) \\
\text{s.t.} & \quad \gamma_j = z_j \\
& \quad \sum_{j=1}^{M} (D_j \cdot \gamma_j) = 0, 0 \leq \gamma_j \\
& \quad \sum_{j=1}^{M} z_j = 1, 0 \leq z_j \\
& \quad 0 \leq \theta \leq 1
\end{align*}
\]

- Solving process with iterations
  - \( \gamma \)-minimization,
  - \( z \)-minimization,
  - dual variable updating
Subproblem P1

- k+1-th iteration, the γ-minimization:

Each CS Agent independently solve the subproblem (P1)

\[
(P_1) \min_{\gamma_j} \gamma_j \cdot \theta \cdot t_j^{\text{trip}} + \frac{\rho}{2}(\gamma_j - z_j^k + \frac{\lambda_j^k}{\rho})^2
\]

s.t. \quad 0 \leq \gamma_j

\[
\sum_{j=1}^{M} (D_j \cdot \gamma_j) = 0
\]
solution to P1

Algorithm 3: Obtain $k+1$-th iteration solution to (23a), denoted as $\gamma^{k+1}_j$.

**Input:** $z^k_j$, $\lambda^k_j$
/* $z^k_j$: k-th iteration solution to (27a); */
/* $\lambda^k_j$: dual variable; */

**Output:** $\gamma^{k+1}_j$
/* $\gamma^{k+1}_j$: decision variable associated with (23a) */

1. **Initialize**
   
   \[
   \eta_{\min} = - (\theta \cdot t^{\text{trip}}_j + \rho + \lambda^k_j) \]
   \[
   \eta_{\max} = - (\theta \cdot t^{\text{trip}}_j - \rho + \lambda^k_j) \]

2. **while** $\eta_{\max} - \eta_{\min} \geq 10^{-10}$ **do**
   
3. \[
   \eta = (\eta_{\max} + \eta_{\min})/2; 
   \]
4. **for** $j = 1, 2, \ldots, M$ **do**
   
5. \[
   \text{Obtain } \gamma_j \text{ according to (25)}; 
   \]
6. **if** $\sum_{j=1}^M (D_j \cdot \gamma_j) > 0$ **then**
   
7. \[
    \text{Set } \eta_{\min} = \eta; 
   \]
8. **else if** $\sum_{j=1}^M (D_j \cdot \gamma_j) < 0$ **then**
   
9. \[
    \text{Set } \eta_{\max} = \eta; 
   \]
10. **else**
11. \[
    \text{Set } \eta_{\max} = \eta_{\min}; 
   \]
**Subproblem P2**

- \( k+1 \)-th iteration, the \( Z \)-minimization:
  
  Each CS Agent independently solve the subproblem (P2)

\[
(P_2) \quad \min_{z_j} \quad z_j \cdot (1 - \theta) \cdot c_j + \frac{\rho}{2} \left( \gamma_j^{k+1} - z_j + \frac{\lambda_j^k}{\rho} \right)^2 \\
\text{s.t.} \quad 0 \leq z_j, \quad \sum_{j=1}^{M} z_j = 1
\]
Algorithm 4: Obtain k+1-th iteration solution to (27a), denoted as $z_j^{k+1}$.

**Input:** $\gamma_j^{k+1}$, $\lambda_j^k$
/* $\gamma_j^{k+1}$ : k+1-th iteration solution to (23a) */
/* $\lambda_j^k$ : dual variable */

**Output:** $z_j^{k+1}$
/* $z_j^{k+1}$ : k+1-th iteration solution to (27a) */

1. Initialize $\eta_{\text{min}} = -((1 - \theta) c_j + \rho + \lambda_j^{k+1})$ and $\eta_{\text{max}} = -((1 - \theta) c_j - \rho + \lambda_j^{k+1})$;
2. while $\eta_{\text{max}} - \eta_{\text{min}} \geq 10^{-10}$ do
   3. $\eta = (\eta_{\text{max}} + \eta_{\text{min}})/2$
   4. for $j = 1, 2, \ldots, M$ do
      Obtain $z_j$ according to (29);
   5. if $\sum_{j=1}^{M} z_j > 1$ then
      Set $\eta_{\text{min}} = \eta$
   6. else if $\sum_{j=1}^{M} z_j < 1$ then
      Set $\eta_{\text{max}} = \eta$
   7. else
      $\eta_{\text{max}} = \eta_{\text{min}}$;
dual variable updating

- after obtaining the optimal $\gamma$ and $z$, the final step is to perform the dual variable update by each CS agent.

\[
\lambda_{j}^{k+1} = \lambda_{j}^{k} + \rho(\gamma_{j}^{k+1} - z_{j}^{k+1})
\]
Outline

- Background
- Problem Formulation
- Algorithm Design
- Experiments
- Conclusion
Experiments

- **Evaluation Settings**
  - Simulator based on Opportunistic Network Environment
  - A period of 24h on the scenario of Helsinki city
  - 6 CSs, each equipped with 5 charging slots, charging power 60 kw.

The regional time-of-use electricity price.
Experiments

- **Comparison methods**
  - **M&C**: (ours)
  - **MTD**: minimizing trip duration
  - **MF**: minimizing charging fee
  - **CWT**: minimizing charging waiting time
  - **R**: random selection

- **Metrics**
  - **Average Trip Duration**
  - **Average Charging Fee**
  - **Weighted Average Cost**
  - **Number of Trips Satisfying the Deadline**
Convergence Evaluation

converges quickly to near the optimum
Influence of Trip Deadline

- Only our method (M&C) and MTD can strictly satisfying the deadline limits.
- For strict deadline, MF, CWT, R is even worse.
Influence of trip duration weight $\theta$

- Only our method (M&C) can help to save the weighted average cost when $\theta$ varies.
Influence of charging power

- Only our method (M&C) gains the advantages of saving money and trip duration
- save more weighted average cost than other methods when the charging power $\beta$ varies
Influence of charging slots

- Only our method (M&C) gains the advantages of saving money and trip duration
- save more weighted average cost than other methods when charging slots $\sigma$ varies
Outline

- Background
- Problem Formulation
- Algorithm Design
- Experiments
- Conclusion
Conclusion

- We study a novel charging coordination problem that jointly optimize charging duration and charging fee.
- We formulated the charging coordination problem as a 0-1 Integer Linear Programming
- We propose a distributed EV charging algorithm based on ADMM, which gradually converges to the optimal solution
- The experiments demonstrate the effectiveness and the efficiency of our proposed method.

Future work

- The effect of charging manners to the power grid