RTSS 2019

ADMM-Based Decentralized Electric Vehicle Charging with Trip Duration Limits

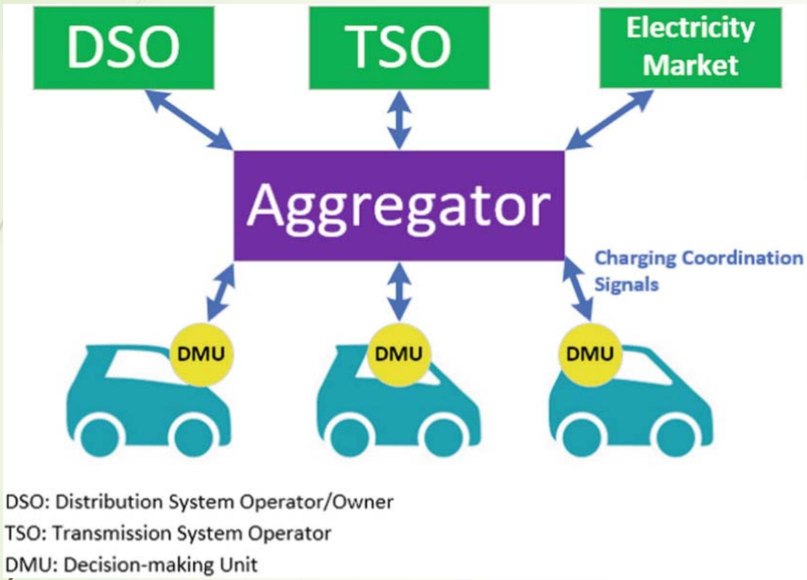
Gaoqi He[1,2], Zhifu Chai[1], Xingjian Lu[2], Fanxin Kong[3], Bing Sheng[4]

[1]East China Normal University, China
[2]*East* China University of Science and Technology, China
[3]Syracuse University, US
[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

Related Works

Minimizing waiting time	Reducing charging fee	Considering charging deadlines
• [6]-[9]	• [10]-[11]	• [12]-[13]

most of them focus on the above aspects separately

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

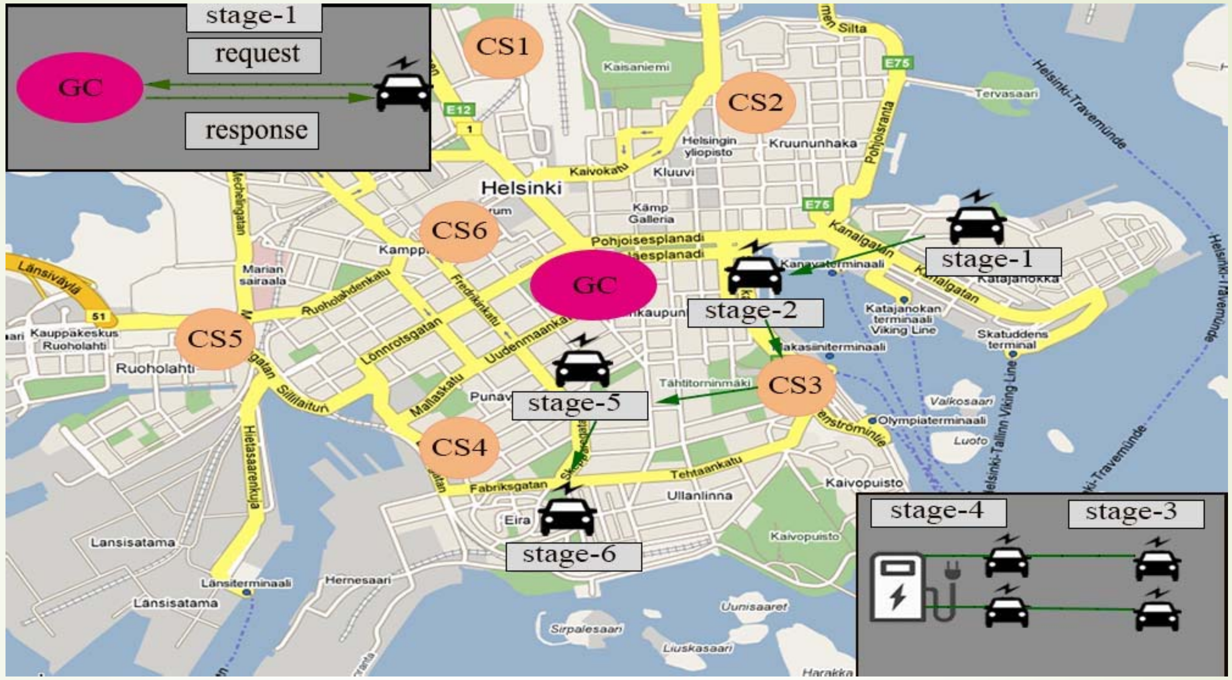
Contributions

- 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

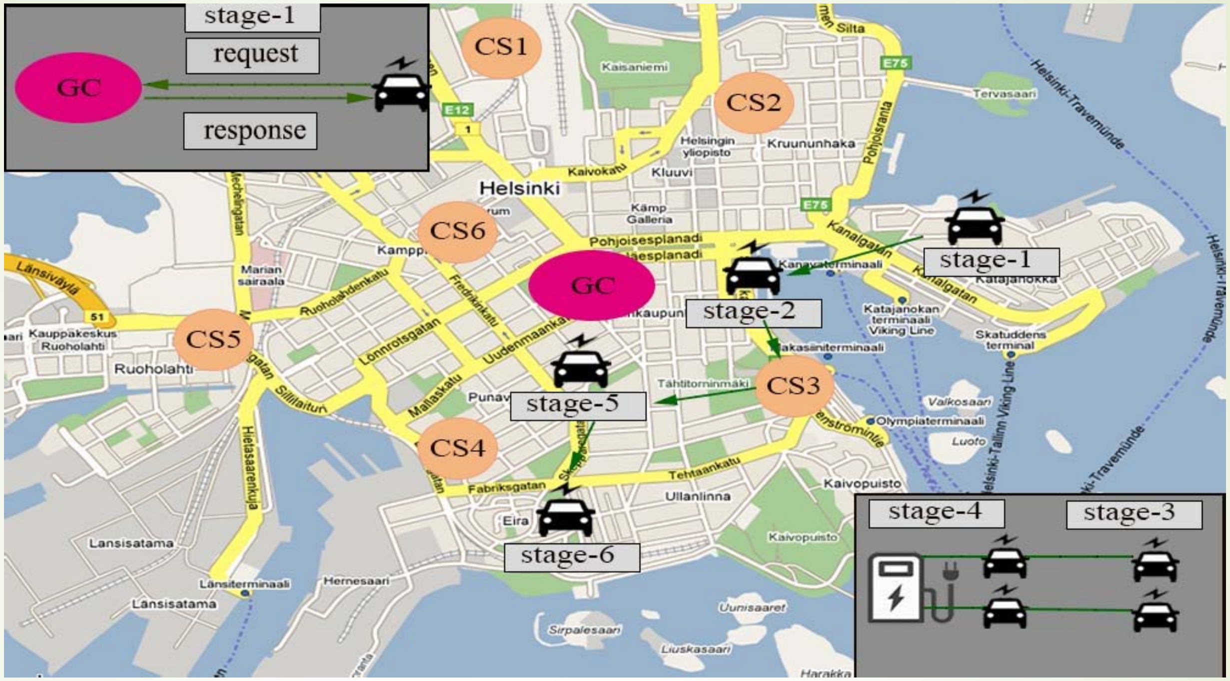
Basic definitions



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: $REQ(Cap, SOC^{cur}, a, l^{req}, l^{des}, T^{deadline})$.

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

trip duration

Charging fee

$$\min \quad \sum_{j=1}^{M} \gamma_j \cdot (\theta \cdot t_j^{trip} + (1-\theta) \cdot c_j)$$

$$\boxed{s.t. \sum_{j=1}^{M} \gamma_j = 1, \gamma_j \in \{0,1\}}$$

Constraint: no more than one CS can selected

$$0 \leq \theta \leq 1$$

$$\boxed{t_j^{trip} \cdot \gamma_j \leq T^{deadline}}$$

Trip duration constraint

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.

$$\begin{aligned} \min \quad & f(x) + g(z) \\ s.t. \quad & Ax + Bz = c \\ & x \in C_1, z \in C_2 \end{aligned}$$

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

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

Problem Reformulation

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

$$\min \quad \sum_{j=1}^{M} (\gamma_j \cdot \theta \cdot t_j^{trip} + z_j \cdot (1-\theta) \cdot c_j) \tag{20a}$$

$$s.t. \quad \gamma_j = z_j \tag{20b}$$

$$\sum_{j=1}^{M} (D_j \cdot \gamma_j) = 0, 0 \leq \gamma_j \tag{20c}$$

$$\sum_{j=1}^{M} z_j = 1, 0 \leq z_j \tag{20d}$$

$$0 \leq \theta \leq 1 \tag{20e}$$

- Solving process with iterations

γ-minimization， z-minimization， dual variable updating

Subproblem P1

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

Each CS Agent independently solve the subproblem (P1)

$$
\begin{aligned}
(P_1) \quad & \min_{\gamma_j} \quad \gamma_j \cdot \theta \cdot t_j^{trip} + \frac{\rho}{2}\left(\gamma_j - z_j^k + \frac{\lambda_j^k}{\rho}\right)^2 \\
& s.t. \quad 0 \le \gamma_j \\
& \qquad\quad \sum_{j=1}^{M} (D_j \cdot \gamma_j) = 0
\end{aligned}
$$

solution to $P1$

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

Input: z_j^k, λ_j^k

/* z_j^k : k-th iteration solution to (27a); */

/* λ_j^k : dual variable; */

Output: γ_j^{k+1};

/* γ_j^{k+1} : decision variable associated with (23a) */

Initialize $\eta_{min} = -(\theta \cdot t_j^{trip} + \rho + \lambda_j^k))$ and $\eta_{max} = -(\theta \cdot t_j^{trip} - \rho + \lambda_j^k)$;

while $\eta_{max} - \eta_{min} \geq 10^{-10}$ **do**

    $\eta = (\eta_{max} + \eta_{min})/2$;

    for $j = 1, 2, \cdots, M$ **do**

        Obtain γ_j according to (25);

    if $\sum_{j=1}^{M}(D_j \cdot \gamma_j) > 0$ **then**

        Set $\eta_{min} = \eta$;

    else if $\sum_{j=1}^{M}(D_j \cdot \gamma_j) < 0$ **then**

        Set $\eta_{max} = \eta$;

    else

        Set $\eta_{max} = \eta_{min}$;

Subproblem P2

- k+1-th iteration, the *Z*-minimization:

Each CS Agent independently solve the subproblem (P2)

$$
\begin{aligned}
(P_2) \quad & \min_{z_j} \quad z_j \cdot (1-\theta) \cdot c_j + \frac{\rho}{2}(\gamma_j^{k+1} - z_j + \frac{\lambda_j^k}{\rho})^2 \\
& s.t. \quad 0 \le z_j, \sum_{j=1}^{M} z_j = 1
\end{aligned}
$$

solution to *P2*

Algorithm 4: Obtain k+1-th iteration solution to (27a), denoted as z_j^{k+1}.

Input: $\gamma_j^{k+1}, \lambda_j^k$

/* γ_j^{k+1} : k+1-th iteration solution to (23a) */

/* λ_j^k : dual variable */

Output: z_j^{k+1};

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

```
Initialize η_min = −((1 − θ)c_j + ρ + λ_j^{k+1}) and
  η_max = −((1 − θ)c_j − ρ + λ_j^{k+1});
while η_max − η_min ≥ 10^{−10} do
    η = (η_max + η_min)/2;
    for j = 1, 2, ··· , M do
        Obtain z_j according to (29);
    if Σ_{j=1}^{M} z_j > 1 then
        Set η_min = η;
    else if Σ_{j=1}^{M} z_j < 1 then
        Set η_max = η;
    else
        η_max = η_min;
```

dual variable updating

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

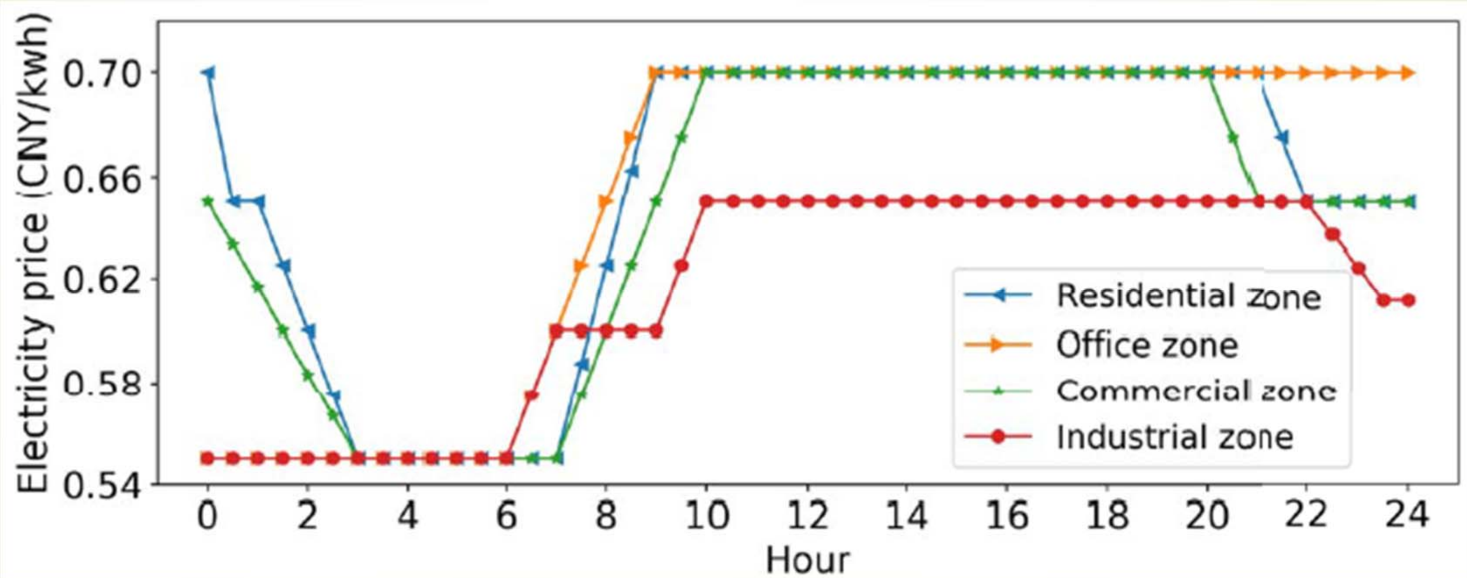
$$\lambda_j^{k+1} = \lambda_j^k + \rho(\gamma_j^{k+1} - z_j^{k+1})$$

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.

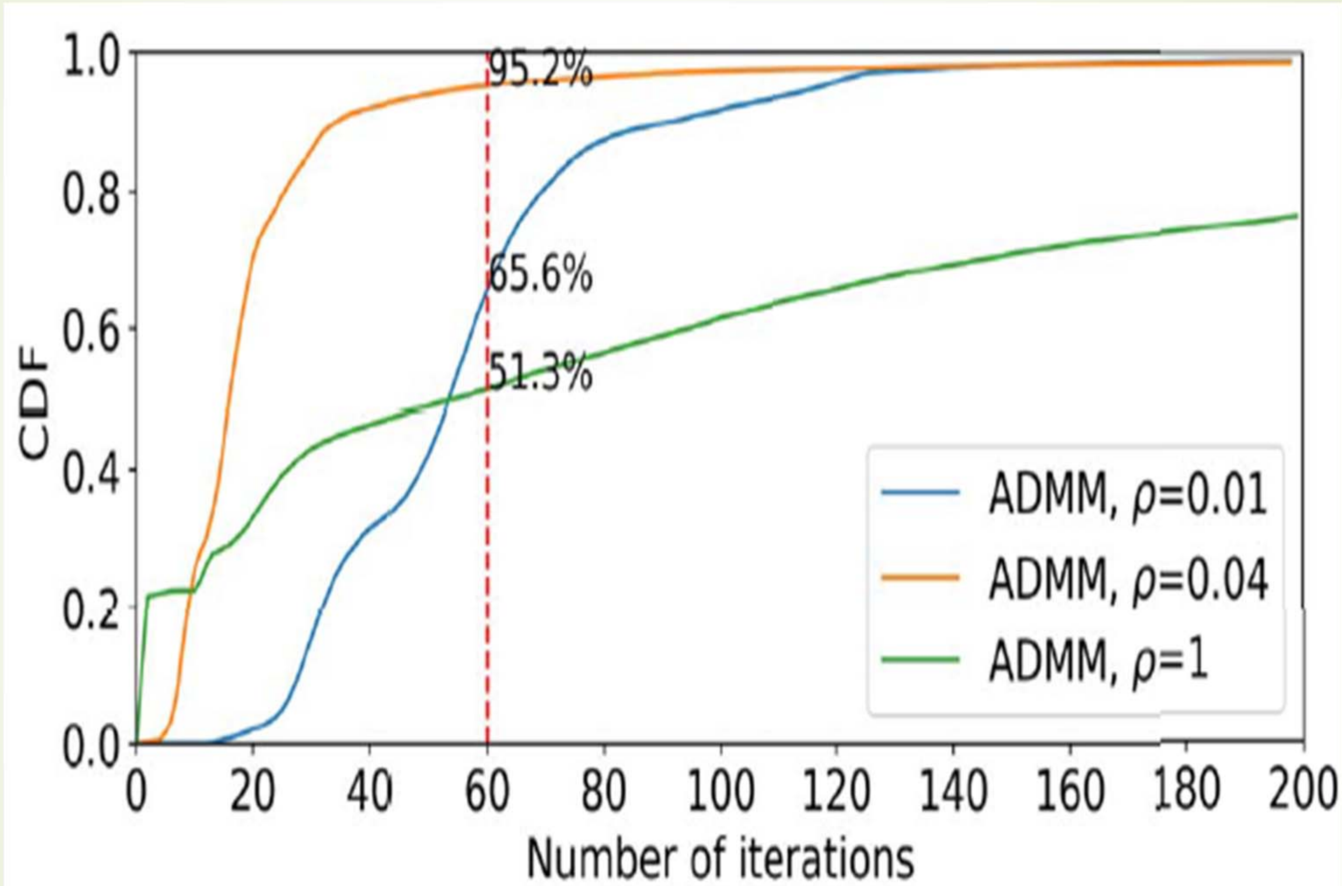


The regional time-of-use electricity price.

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

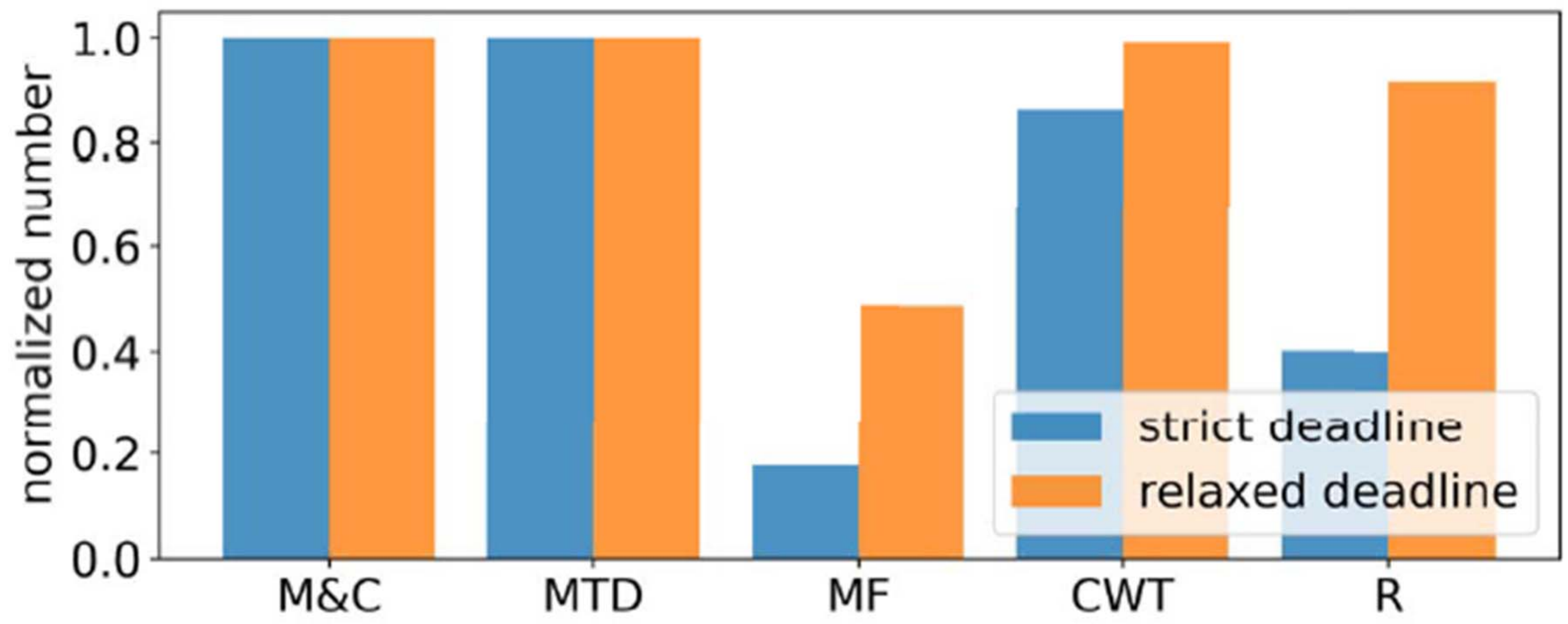
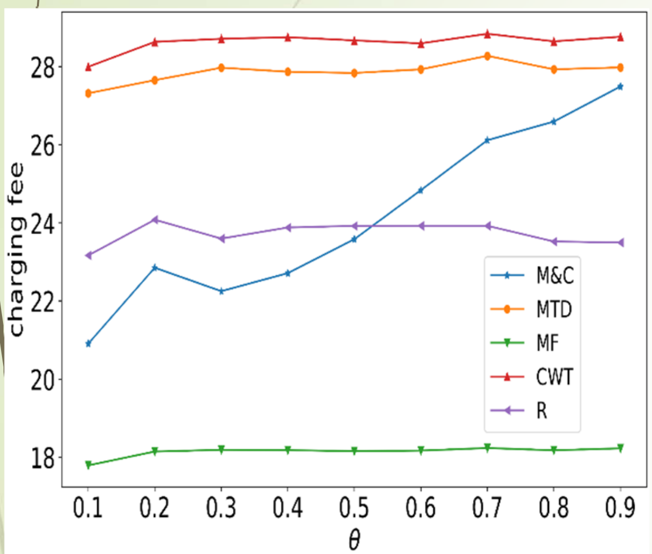
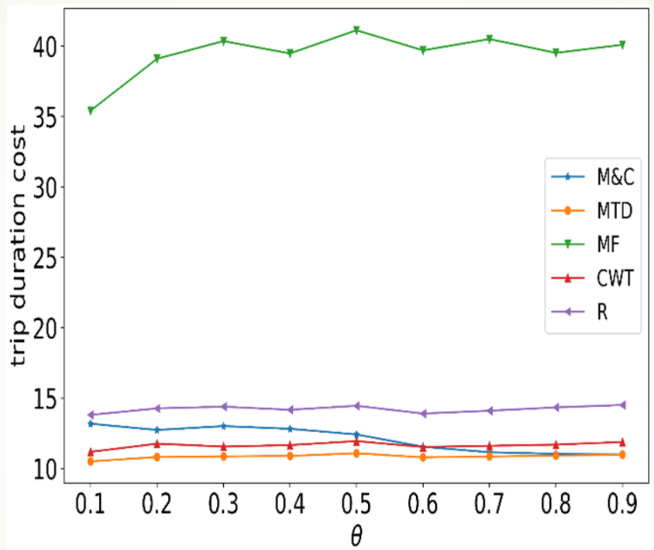


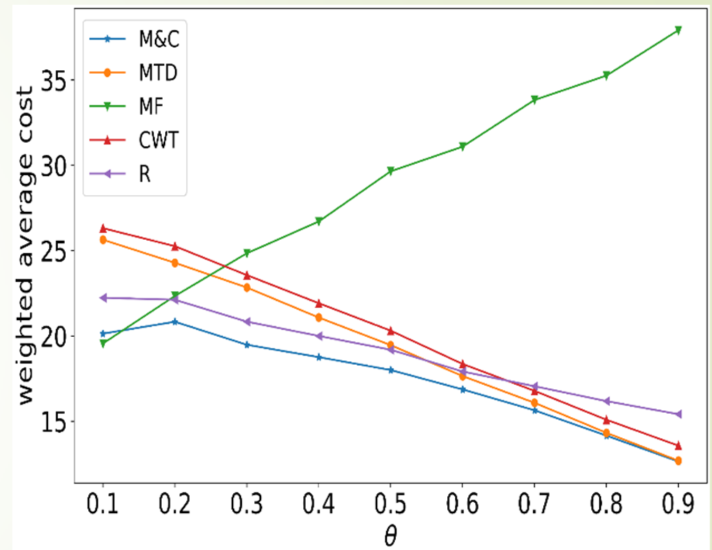
Fig. 5. Normalized number of trip results generated by different methods that satisfy trip deadline when EV drivers set strict and relaxed deadlines

- 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 θ

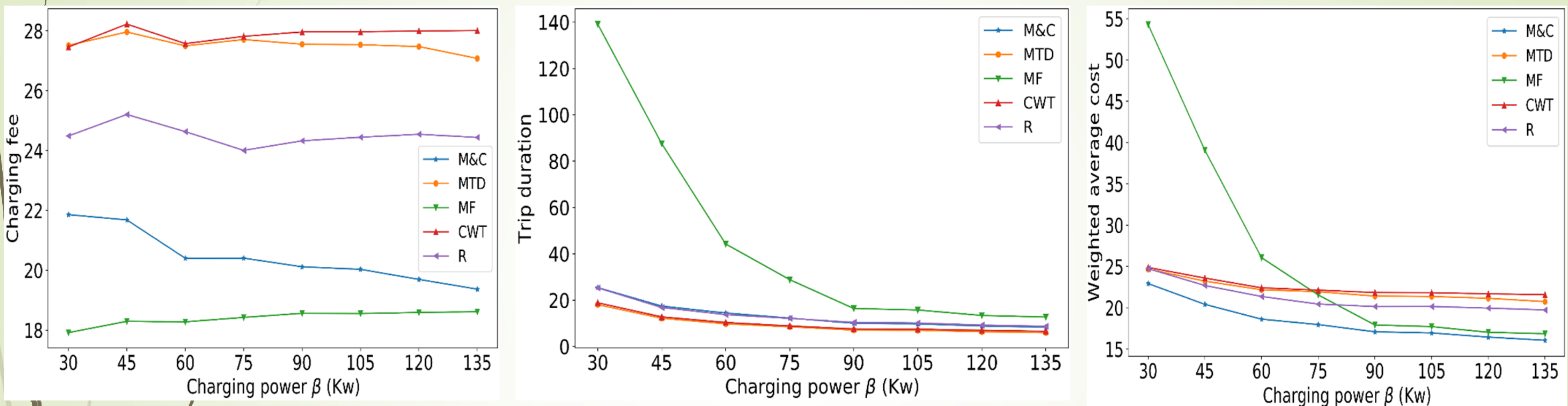






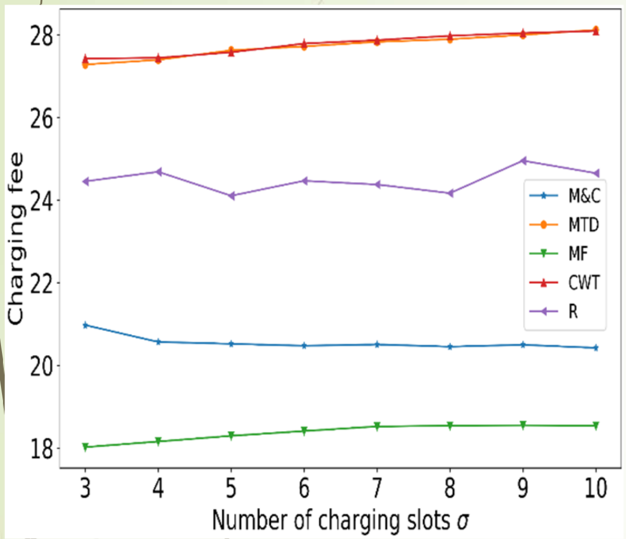
- Only our method (M&C) can help to save the weighted average cost when θ varies

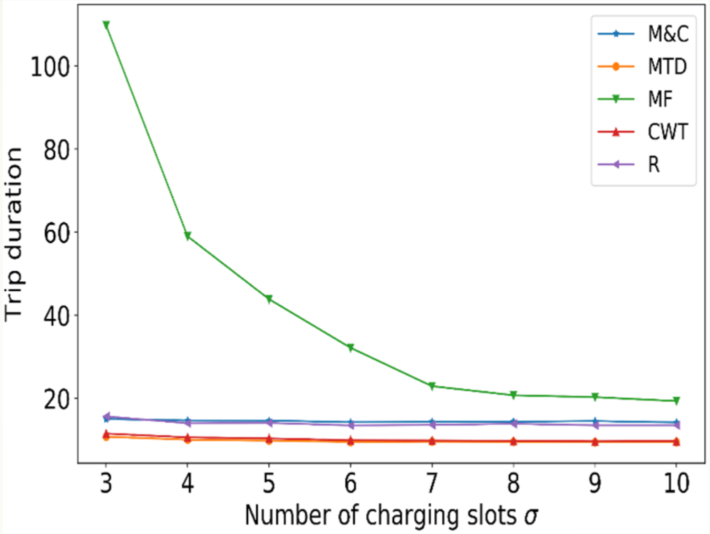
Influence of charging power

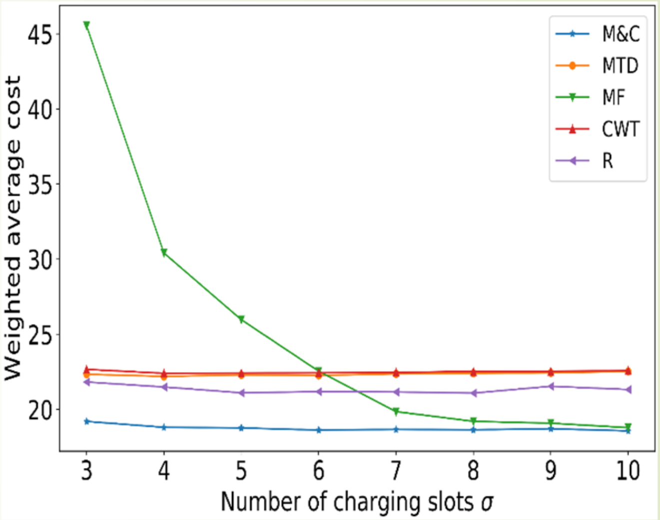


- Only our method (M&C) gains the advantages of saving money and trip duration
- save more weighted average cost than other methods when the charging power β varies

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 σ 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