A Stochastic Resource-Sharing Network for Electric Vehicle Charging

Bert Zwart

CWI Amsterdam and TU Eindhoven

Simons Institute
March 26, 2018

joint work with Angelos Aveklouris and Maria Vlasiou (TU Eindhoven)
The rise of Electric Vehicles

Evolution of the global electric car stock, 2010-16

- Electric car stock (millions)
- Countries: Others, Sweden, Germany, France, United Kingdom, Netherlands, Norway, Japan, United States, China, BEV, BEV + PHEV

Deployment scenarios for the stock of electric cars to 2030

- Electric cars in the vehicle stock (millions)
- Years: 2010, 2015, 2020, 2025, 2030
- Scenarios: IEA B2DS, IEA 2DS, Paris Declaration, IEA RTS, Historical
- Targets: Cumulative country targets (as of 2016), Cumulative OEMs announcements (estimate), India target (as of 2017)

Key points:
- The global electric car stock surpassed 2 million vehicles in 2016 after crossing the 1 million threshold in 2015.
- The electric car stock has been growing since 2010 and surpassed the 2 million-vehicle threshold in 2016.
- Key point: Consistent with the ambition of the EV30@30 campaign, provided that the carbon intensity of power generation declines rapidly.
Types of charging
Bottlenecks in EV charging

Current:
   Ability to charge a battery fast
   Number of charging stations

Future:
   Capacity of the grid
A 2025 scenario: charging electric vehicles, baking pizzas, and melting a fuse in Lochem

EV charging: stochastic process on interacting networks

- \( I \) charging stations
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces

Arrivals: $\lambda_i$

Charging requirement: $B_i$

Parking time: $D_i$

$Q_i(t)$: Number of EVs

$Z_i(t)$: Number of uncharged EVs

Aim: efficient charging schedule while keeping voltage drop bounded
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces
- Arrivals: $\lambda_i$
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces
- Arrivals: $\lambda_i$
- Charging requirement: $B$

Arrivals: $\lambda_i$

Charging requirement: $B$

Aim: efficient charging schedule while keeping voltage drop bounded
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces
- Arrivals: $\lambda_i$
- Charging requirement: $B$
- Parking time: $D$
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces
- Arrivals: $\lambda_i$
- Charging requirement: $B$
- Parking time: $D$
- $Q_i(t)$: Number of EVs

Aim: efficient charging schedule while keeping voltage drop bounded
EV charging: stochastic process on interacting networks

- $I$ charging stations
- $K_i$ parking spaces
- Arrivals: $\lambda_i$
- Charging requirement: $B$
- Parking time: $D$
- $Q_i(t)$: Number of EVs
- $Z_i(t)$: Number of uncharged EVs
EV charging: stochastic process on interacting networks

- \( l \) charging stations
- \( K_i \) parking spaces
- Arrivals: \( \lambda_i \)
- Charging requirement: \( B \)
- Parking time: \( D \)
- \( Q_i(t) \): Number of EVs
- \( Z_i(t) \): Number of uncharged EVs

Aim: efficient charging schedule while keeping voltage drop bounded


Assessing voltage drop: a tractable load flow model

**Linearized Distflow:** \( W_{kk}^{lin} := |V_k^{lin}|^2 \)

\[
W_{kk}^{lin} = W_00 - 2 \sum_{e_{ls} \in P(k)} R_{ls} \sum_{m \in I(s)} z_m p_m,
\]

- \( z = (z_i, i \geq 1) \) denotes the number of uncharged EVs in the network at some particular time
- **Each EV at node** \( i \) **receives power** \( p_i \)
- \( \sum_{m \in I(s)} z_m p_m \) is the consumed power by subtree rooted in node \( s \)
Assessing voltage drop: a tractable load flow model

- **Linearized Distflow:** \( W_{kk}^{lin} := |V_k^{lin}|^2 \)

\[
W_{kk}^{lin} = W_00 - 2 \sum_{\epsilon_l \in \mathcal{P}(k)} R_{ls} \sum_{m \in \mathcal{I}(s)} z_m p_m,
\]

- \( z = (z_i, i \geq 1) \) denotes the number of uncharged EVs in the network at some particular time
- **Each EV at node** \( i \) **receives power** \( p_i \)
- \( \sum_{m \in \mathcal{I}(s)} z_m p_m \) is the consumed power by subtree rooted in node \( s \)

- Some of our results also hold for more general AC (on trees)
Assessing voltage drop: a tractable load flow model

**Linearized Distflow:** \( W_{kk}^{lin} := |V_k^{lin}|^2 \)

\[
W_{kk}^{lin} = W_{00} - 2 \sum_{\epsilon_s \in \mathcal{P}(k)} R_{ls} \sum_{m \in \mathcal{I}(s)} z_m p_m,
\]

- \( z = (z_i, i \geq 1) \) denotes the number of uncharged EVs in the network at some particular time
- **Each EV at node** \( i \) **receives power** \( p_i \)
- \( \sum_{m \in \mathcal{I}(s)} z_m p_m \) is the consumed power by subtree rooted in node \( s \)

Some of our results also hold for more general AC (on trees)

Next: how to schedule amount of power for each battery
Power allocation: use network utility maximization

- \( z = (z_i, i \geq 1) \): number of uncharged EVs in the network
- \( p = (p_i, i \geq 1) \): allocated power to vehicles at node \( i \)

Each EV receives utility \( u_i(p_i) \). Example: \( u_i(p_i) = w_i \log(p_i) \)
**Power allocation: use network utility maximization**

- \( z = (z_i, i \geq 1) \): number of uncharged EVs in the network
- \( p = (p_i, i \geq 1) \): allocated power to vehicles at node \( i \)
- Each EV receives utility \( u_i(p_i) \). Example: \( u_i(p_i) = w_i \log(p_i) \)

\[
p = \arg \max \sum_{i=1}^{l} z_i u_i(p_i)
\]

subject to

- \( z_i p_i \leq M_i, \quad 0 \leq p_i \leq c^{\max}, \)
- \( \nu_i \leq W_{ii}^{lin}(p, z) \leq \overline{\nu}_i \)
Power allocation: use network utility maximization

- \( z = (z_i, i \geq 1) \): number of uncharged EVs in the network
- \( p = (p_i, i \geq 1) \): allocated power to vehicles at node \( i \)
- Each EV receives utility \( u_i(p_i) \). Example: \( u_i(p_i) = w_i \log(p_i) \)

\[
p = \arg \max \sum_{i=1}^{l} z_i u_i(p_i)
\]

subject to
- \( z_i p_i \leq M_i, \quad 0 \leq p_i \leq c^{\text{max}}, \)
- \( \underline{v}_i \leq W_{ii}^{\text{lin}}(p, z) \leq \bar{v}_i \)

- This talk: assess performance
Solvable special case: product-form property

Proportional fairness in line network

Consider line network with $K_i = M_i = \infty$ for all $i$, $c^{\max} = \infty$. Set $\bar{R}_i = \sum_{j=1}^{i} R_j$.

If $u_i(p_i) = \log p_i$, then for every $n \in \mathbb{N}_+$,

$$\lim_{t \to \infty} \mathbb{P}(Z(t) = n) = (1 - \rho)\left(\sum_{i=1}^{l} n_i\right)! \prod_{i=1}^{l} \frac{\rho_i^{n_i}}{n_i!},$$

provided $\rho = \sum_{i=1}^{l} \rho_i = \sum_{i=1}^{l} \lambda_i \mathbb{E}[B] \bar{R}_i / \delta < 1$. 

Proof idea: reduction to multi-class Processor queue

New application of Processor Sharing after mainframe computer systems (70s) and communication networks (90s).

Bert Zwart (CWI)
Electric vehicle charging
Solvable special case: product-form property

Proportional fairness in line network

Consider line network with $K_i = M_i = \infty$ for all $i$, $c^{\text{max}} = \infty$. Set $\bar{R}_i = \sum_{j=1}^{i} R_j$.

If $u_i(p_i) = \log p_i$, then for every $n \in \mathbb{N}_+$,

$$
\lim_{t \to \infty} \mathbb{P}(Z(t) = n) = (1 - \rho) \left(\sum_{i=1}^{l} n_i\right)! \prod_{i=1}^{l} \frac{\rho_i^{n_i}}{n_i!},
$$

provided $\rho = \sum_{i=1}^{l} \rho_i = \sum_{i=1}^{l} \lambda_i \mathbb{E}[B] \bar{R}_i / \delta < 1$.

- Proof idea: reduction to multi-class Processor queue
Solvable special case: product-form property

Proportional fairness in line network

Consider line network with \( K_i = M_i = \infty \) for all \( i \), \( c_{\text{max}} = \infty \).
Set \( R_i = \sum_{j=1}^{i} R_j \).
If \( u_i(p_i) = \log p_i \), then for every \( n \in \mathbb{N}_+ \),

\[
\lim_{t \to \infty} \mathbb{P}(Z(t) = n) = (1 - \rho) \left( \sum_{i=1}^{l} n_i \right)! \prod_{i=1}^{l} \frac{\rho_{n_i}}{n_i!},
\]

provided \( \rho = \sum_{i=1}^{l} \rho_i = \sum_{i=1}^{l} \lambda_i \mathbb{E}[B] R_i / \delta < 1 \).

- Proof idea: reduction to multi-class Processor queue
- New application of Processor Sharing after mainframe computer systems (70s) and communication networks (90s).
General case: Measure-valued process

Particles move to the left at rate \( p_r(z) \) and to the south at rate \( z_r \).
Computational analysis of stochastic model: hard

- Markov Chain (MC) model (assuming exponential rv’s)

\[ p(z) = \arg \max_p \sum_{i=1}^l z_i u_i(p_i) \]

- For a line network with identical parking lots of size \( K \) and \( R \) nodes, computing the equilibrium distribution of the MC requires solving \( O(K^2R) \) convex optimization problems.

- Things are even worse for non-exponential distributions, so we need another approach.
Fluid approximation
Fluid approximation

**Scaling:**
- Consider a family of models indexed by $n$
- Capacity of power in the $n^{th}$ system: $nM_i$
- Arrival rate in the $n^{th}$ system: $n\lambda_i$
- Number of parking spaces in the $n^{th}$ system: $nK_i$

Fluid approximation at time $t \geq 0$: $z(t) = (z_i(t), i \geq 1)$

$z_i(t) \rightarrow z_i^*$ as $t \rightarrow \infty$ and $z_i^*$ is an invariant point
Main result for fluid approximation

Characterization of performance

$z^*$ is given by $z^*_i = \frac{\Lambda^*_i}{g_i^{-1}(\Lambda^*_i)}$, with

$$\Lambda^* = \arg \max \sum_{i=1}^l G_i(\Lambda_i)$$

subject to $\Lambda_i \leq M_i$, $0 \leq \Lambda_i \leq g_i(c^{\text{max}})$,

$\nu_i \leq W^{|\text{lin}}_{ii}(\Lambda) \leq \bar{\nu}_i$,

$G_i'(\cdot) = u_i'(g_i^{-1}(\cdot))$ with $g_i(\cdot) := \gamma_i \mathbb{E}[\min\{Dx, B\}]$ and $\gamma_i = \min\{\lambda_i, K_i \mathbb{E}[D]\}$.
Main result for fluid approximation

Characterization of performance

\( z^* \) is given by \( z_i^* = \frac{\Lambda_i^*}{g_i^{-1}(\Lambda_i^*)} \), with

\[
\Lambda^* = \arg \max \sum_{i=1}^{l} G_i(\Lambda_i)
\]

subject to

\[
\Lambda_i \leq M_i, \quad 0 \leq \Lambda_i \leq g_i(c^{\text{max}}),
\]

\[
\nu_i \leq W_{ii}^{\text{lin}}(\Lambda) \leq \bar{\nu}_i,
\]

\( G_i'(\cdot) = u_i'(g_i^{-1}(\cdot)) \) with

\( g_i(x) := \gamma_i \mathbb{E}[\min\{D_x, B\}] \) and \( \gamma_i = \min\{\lambda_i, K_i \mathbb{E}[D]\} \).

Fraction of cars that get successfully charged: \( P(D > Bz_i^* / \Lambda_i^*) \).
Idea of the proof: Little is known

- Little’s law: \((\text{Expected number of customers}) = (\text{arrival rate}) \times (\text{sojourn time})\)

- Snapshot principle: constant service rate in equilibrium

\[ z_i^* = \gamma_i E\left[\min\{D, \frac{B}{p_i(z^*)}\}\right] \]

- Add this approximate version of Little’s law to Karush-Kuhn-Tucker (KKT) equations for optimization problem that defines \(p(z)\).

- Still works for AC in this case, as convex relaxation of associated semi-definite program is exact.
Validating Distflow (two lots, Markovian model)

Table: Stochastic model

<table>
<thead>
<tr>
<th>$K$</th>
<th>$\mathbb{E}[Z_1], \mathbb{E}[Z_2]$ (Distflow)</th>
<th>$\mathbb{E}[Z_1], \mathbb{E}[Z_2]$ (AC)</th>
<th>Rel. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.5336, 4.6179</td>
<td>4.6801, 4.6924</td>
<td>3.13 %, 1.58 %</td>
</tr>
<tr>
<td>20</td>
<td>14.0174, 14.0385</td>
<td>14.1725, 14.1948</td>
<td>1.09 %, 1.10 %</td>
</tr>
</tbody>
</table>

Table: Fluid approximation

<table>
<thead>
<tr>
<th>$K$</th>
<th>$z_1^<em>, z_2^</em>$ (Distflow)</th>
<th>$z_1^<em>, z_2^</em>$ (AC)</th>
<th>Rel. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.5769, 4.5769</td>
<td>4.7356, 4.7513</td>
<td>3.35%, 3.67%</td>
</tr>
<tr>
<td>20</td>
<td>14.0300, 14.0300</td>
<td>14.1849, 14.2069</td>
<td>1.09%, 1.25%</td>
</tr>
<tr>
<td>30</td>
<td>23.6820, 23.6820</td>
<td>23.8357, 23.8597</td>
<td>0.64%, 0.74%</td>
</tr>
<tr>
<td>40</td>
<td>33.4293, 33.4293</td>
<td>33.5823, 33.6073</td>
<td>0.45%, 0.53%</td>
</tr>
<tr>
<td>50</td>
<td>43.2330, 43.2330</td>
<td>43.3857, 43.4112</td>
<td>0.35%, 0.41%</td>
</tr>
</tbody>
</table>
Validating the fluid approximation

<table>
<thead>
<tr>
<th>$K$</th>
<th>Distflow</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.95%, 0.86%</td>
<td>1.18%, 1.25%</td>
</tr>
<tr>
<td>20</td>
<td>0.09%, 0.06%</td>
<td>0.08%, 0.08%</td>
</tr>
</tbody>
</table>

Figure: $K = (10, 10)$ and $\lambda = (12, 12)$
Consider a line network, recall $\bar{R}_i = \sum_{j=1}^{i} R_i$.

**Fairness property**

Assume that $M_i = c^{\text{max}} = \infty$, and $u_i(p_i) = w_i \log(p_i)$. If $w_i = \bar{R}_i$, then we have that $p_i(z) = p(z) > 0$. Moreover, $p(z) = \delta \left( \sum_{i=1}^{l} \bar{R}_i z_i \right)^{-1}$.

Not very efficient
Goal: Choose weights that maximize the fraction of EVs that get successfully charged under weighted proportional fairness, i.e.,
\[ u_i(p_i) = w_i \log(p_i) \]
**Goal**: Choose weights that maximize the fraction of EVs that get successfully charged under weighted proportional fairness, i.e., 
\[ u_i(p_i) = w_i \log(p_i) \]

For a line network topology, \( M_i = c^{\text{max}} = \infty \), it can be shown that this fraction can be optimized by choosing the weights as follows:

\[
\max_{w} \sum_{i=1}^{l} \gamma_i \mathbb{P}(w_i D > R_i B) \\
\text{subject to } \sum_{i=1}^{l} \gamma_i \mathbb{E}[\min\{w_i D, R_i B\}] \leq \delta
\]

Non-convex, depends on the joint distribution of \((B, D)\)
\( B = HD, \ H \) deterministic

Weight-finding problem can be transformed into

\[
\max_x \sum_{i=1}^I \gamma_i x_i \\
\text{subject to} \sum_{i=1}^I \mathbb{E}[D] \gamma_i \overline{R}_i H x_i \leq \delta, \ x_i \in \{0, 1\}
\]

Knapsack problem

Yields distributionally robust solution (i.e. homogeneous user preferences are the worst if the system is overloaded)
$B = HD$, $H$ decreasing hazard rate.

Example: $H \sim \text{Pareto}(\alpha, \kappa)$, i.e., $P(H > x) = \left(\frac{\kappa}{x + \kappa}\right)^\alpha$, $x \geq 0$, $\alpha > 1$, $\kappa > 0$.

Weight-finding problem can be transformed into a convex programming problem:

$$\max_y \sum_{i=1}^l \gamma_i (1 - y_i^{\alpha/(\alpha-1)})$$

subject to $\sum_{i=1}^l \frac{\mathbb{E}[D] \kappa \gamma_i \overline{R_i}}{1 - \alpha} (y_i - 1) \leq \delta$, $0 \leq y_i \leq 1$
Weight selection: numerics

Assume 10 nodes with \( \sum_{i=1}^{10} \frac{\mathbb{E}[B] \gamma_i R_i}{\delta} = 1.2 \)

The optimal weights are given by:

<table>
<thead>
<tr>
<th>Det.</th>
<th>0.06</th>
<th>0.09</th>
<th>0.11</th>
<th>0.12</th>
<th>0.13</th>
<th>0.14</th>
<th>0.15</th>
<th>0.16</th>
<th>0</th>
<th>0</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto (3)</td>
<td>0.12</td>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
<td>0.09</td>
<td>0.09</td>
<td>0.08</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>9.5</td>
</tr>
<tr>
<td>Pareto (1.1)</td>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

The more variability, the better the system performance
Dynamic fluid equations \((K_r = \infty)\)

\[ z_r(t) = \int_0^t \lambda_i \mathbb{P}(D_r > t-s) B_r > \int_s^t p_r(z(u)) du \, ds \quad \text{for} \quad r = 1, \ldots, R, \quad t \geq 0 \]

Recall

\[ p(z) = \arg \max \sum_{i=1}^l z_i u_i(p_i) \]

subject to

\[ z_i p_i \leq M_i, \quad 0 \leq p_i \leq c^{\max}, \]

\[ \bar{v}_i \leq W^{\text{lin}}_{ii}(p, z) \leq \underline{v}_i \]

Unique solution thanks to the properties: \(p(z)\) is Lipschitz continuous, \(p(z)\) is nonincreasing in \(z\).
Convergence to equilibrium

Proposition: $z(t) \to z^*.$

Proof: Define

$$l_r = \liminf_{t \to \infty} z_r(t) \quad u_r = \limsup_{t \to \infty} z_r(t).$$

Using the dynamic fluid equations, it can be shown that

$$l_r = \lambda_r \mathbb{E}[\min\{D, \frac{B}{M_r(l, h)}\}] \quad u_r = \lambda_r \mathbb{E}[\min\{D, \frac{B}{m_r(l, h)}\}].$$

with

$$M_r(l, h) = \sup_{z \in [l, h]} p_r(z) \quad m_r(l, h) = \inf_{z \in [l, h]} p_r(z)$$

using monotonicity of $p(z)$ and the characterization of $z^*$ using Little’s law, uniqueness follows.
Concluding comments


- Multiclass extension: see preprint

- Currently working out a rigorous justification of the fluid scaling, allowing time-varying arrival rates

- Challenges: time-dependent behavior, controlling arrival rates of cars, incorporating markets explicitly, reducing communication overhead by discretizing time, adding reactive power support, allowing for additional (noisy) behavior of other types of users, ...