Demand response of electric vehicles using reinforcement learning

Research topics

Frederik Ruelens$^1$

$^1$KULeuven ELECTA
EnergyVille (University of Leuven, VITO (Flemish Institute of Technology) and Imec)
frederik.ruelens@esat.kuleuven.be

December 13, 2012
Background

Education

- Graduated as an electrical engineer
  - KULeuven, 2010
- Currently PhD Student at Department of Electrical Engineering, div ELECTA
  - IWT scholarship 2012-2016

Topic of this presentation

- Present my ongoing research
  - Specific problem where I used reinforcement learning
- Get feedback!
  - Learn new techniques
  - Learn new scenarios
Opportunities for demand response

Problem statement

I Capacity shortage
- Nuclear phase-out

II Intraday problem
- New fluctuating load, e.g. electric vehicles
- New renewable generation, e.g. solar panels

Solutions

- Old solutions
  - switch on gas-fired power plants
  - import/export
- New solution: Demand response
  - demand \textit{adapts} generation
Research topics

What is my research about?

- Demand response algorithms for Smart Grids
  - Residential load, e.g. EVs, heat pumps, electrical boilers
  - Stochastic optimization
  - Learning

Why is learning import in smart grid?

- We don’t now the flexibility a household
  - EVs it is reasonable to assume
  - but what about heat pumps and electrical boilers
- We don’t now the transition model of the distribution grid
  - No information about voltage deviations and congestions
Intelligator

I. Aggregation
   ▶ Bidfunctions
II. Optimization
   ▶ Q-learning
III. Dis-aggregation
   ▶ Heuristic

Advantages
▶ Simple
▶ Compatible with PowerMatcher
▶ Limited local intelligence
Problem statement

- Aggregator of a fleet of electric vehicles
  - **Objective:** make money by charging his fleet as cheap as possible as long as the EV batteries are charged at the end of the flexibility horizon
  - **EV settings**
    - Required energy 10 kWh, 3 kW capacity, flexibility window: 5 pm until 8 am

- Distribution System Operator
  - **Objective:** operate the distribution grid within specified safety margins
    - Congestions
    - Voltage deviations
    - Aging of its assets
Experiment: Droop charger (2)

- Droop mechanism for EV chargers
  - Droop controllers can help succeed in limiting voltage deviation

![Graph showing the droop characteristic](image)

- The droop is an easy solution for the DSO, in order to safeguard the grid
- As a result the aggregator will have problems charging some EVs in peak moments
Experiment: Droop charger (2)

- Droop mechanism for EV chargers
  - Droop controllers can help succeed in limiting voltage deviation

The droop is an easy solution for the DSO, in order to safeguard the grid

As a result the aggregator will have problems charging some EVs in peak moments
Experiment: Distribution feeder (3)

- Uni-branched feeder
- 40 household (Linear profiles)
- 15 EVs
- Proportional controller
Experiment: Problem statement (4)

We want to learn a sequential decision policy that minimizes the cumulative cost.

- Discrete-time system whose dynamics over $T$ stages is given by

$$x_{t+1} = f^M(x_t, u_t, w_t), x_t \in X, u_t \in U \text{ and } w_t \in P_w$$ (1)

- Cost:

$$r_t = \lambda^t_{el} u_t + \text{cost of non delivered energy}, \quad \text{where } \lambda_{el} \text{ is a electricity price}$$ (2)

- As result of the Droop controller some batteries are not fully charged at the end of their flexibility horizon
Experiment: Fitted Q-iteration (5)

- Problem:
  
  The functions $f^M$, $\rho$ and $P_w$ are unknown. \hfill (3)

- We have a set of sample transitions

  \[
  \mathcal{F} = \{x^l, u^l, r^l, y^l\}_{l=1}^n,
  \] \hfill (4)

  where the pairs $(x^l, u^l)$ are chosen by the aggregator and the pairs $(r^l, y^l)$ are determined by the $f^M$ and $\rho$.

- Fitted Q-iteration

  compute from $\mathcal{F}$ the functions $\hat{Q}_1, \hat{Q}_2, \ldots, \hat{Q}_N$, approximations of $Q_1, Q_2, \ldots, Q_N$

  - Regression strategy: Extremely randomized trees
Experiment: Results (6)

Results using fitted Q-iteration

▷ Still in the simulation phase

▷ Promising results after 30-40 episodes
  ▷ Using different price signals in each episode
  ▷ We were able to learn the charging plan
    ▷ That minimizes the charging cost
    ▷ The non delivered energy
  ▷ More simulation are needed to make quantitative statements

▷ Future research
  ▷ Try different scenarios
  ▷ Computation burden of storing large set of trajectories and approximating them
Problem: Making day-ahead planning

- A pool-based energy market is a short-term market place, that generally consists of a day-ahead and real-time (balancing) market.
- BRP is responsible for keeping its energy balance for each settlement period
- Many sources of uncertainty: imbalance prices, flexibility and wind generation
- Nested optimization problem
  I Day-ahead nomination
  II Real-time planning
Problem: Making day-ahead planning

\[
\begin{align*}
\text{Maximize} & \quad P_{n}, P_{EV} \\
& \quad \lambda_c \top P_n + \mathbb{E} \left\{ \lambda_p \top [P_n - P]_+ - \lambda_n \top [P - P_n]_+ \right\}, \quad (5) \\
\text{Subject to} & \quad P = P_{EV} + P_w \quad (6) \\
& \quad P_{\min,t} \leq P_{n,t} \leq P_{\max,t} \quad \forall t \in T_{24} \quad (7) \\
& \quad E_{\min,t} \leq \sum P_{EV,t} \leq E_{\max,t} \quad \forall t \in T_{96} \quad (8)
\end{align*}
\]

- **Prices**: \( \lambda_c \) day-ahead price, \( \lambda_p \) positive imbalance price, \( \lambda_n \) negative imbalance price
- **control variables**: \( P_n \) day-ahead nomination and \( P_{EV} \) the real-time planning of the EVs

I finding the day-ahead nomination \( P_n \): CE optimization method

II finding a real-time planning \( P_{EV} \): Approximate policy iteration (fitness evaluations)
**Algorithm 1** CE-PI optimization method

**Input:** initialize CE parameters, $N_{CE}$, $\tau_{\text{max}}$ and $\rho_{CE}$

1: generate initial population of sample nominations from initial seed
2: while $\tau < \tau_{\text{max}}$ do
3: \hspace{1em} $\tau \leftarrow \tau + 1$
4: \hspace{1em} generate samples $P_{n,i}, \ldots, P_{n,N_{CE}}$ from Gaussian distribution given by $\mu_{\tau-1}$ and $\sigma_{\tau-1}$
5: \hspace{1em} for $i = 1, \ldots, N_{CE}$ do
6: \hspace{2em} run approximate policy iteration with extremely randomized trees (Algorithm 2) and calculate the score function
7: \hspace{1em} end for
8: reorder and reindex samples and recalculate $\mu_{\tau}$ and $\sigma_{\tau}$, save the $\rho_{CE}$ samples with the highest score functions
9: end while

**Output:** sample $P_n$ with highest score function
Thank you for listening!