Demand response – scheduling and aggregation

Josh Taylor

1 Overview

Basic idea:

- Loads adjust power consumption to help power systems
- Industrial loads (steel mill), buildings (Ikea, Walmart, supermarkets), devices (lights, HVAC, 'smart' appliances, EVs)
- Key issue - communication between load and aggregator

Functionalities

- Curtailment
- Load-shifting/arbitrage
- Power balancing
- Regulation

Paradigms

- Direct load control
  - Aggregator controls load
  - Versatile, fast, better for power system
  - Minimal attention from load
- Indirect load control
  - Load operator (e.g., home resident) controls load in response to signal, e.g., price
  - Less reliable, implausible for fast services

  - Gives choice to load
- Hybrids
  - Can be worst of both worlds
  - Less flexible than DLC
  - Little consumer choice with programmed response
  - Can induce volatility
  - Some let consumers choose degree of involvement

Problems in DR

- Managing loads
  - Scheduling
  - Controlling
- Representing loads
  - Aggregator communicates capabilities to SO
- Paying loads
  - Dynamic pricing for ILC
  - Contracts that incentivize participation in DLC.
- Uncertainty
  - Few measurements, power not information measurements, inaccurate models, exogenous inputs (humans, weather ...)
  - Identification - what's the model?
  - Estimation - what's the state?
  - Learning - balancing usage with estimation/learning

2 Basic problem formulation

2.1 Tasks

Many flexible resources must be scheduled. Time of energy delivery not important.

- Laundry
- Dishwasher
- EV charging

Parameters

- Task i present from time $\{a_i, ..., d_i\}$ (arrival and departure)
- Power i receives at time k: $p_i(k)$
- Total energy need: E_i

Constraints

- Task completion:
$$\sum_k p_i(k) = E_i$$
- Power limit:
$$0 \leq p_i(k) \leq m_i$$
- Arrival/departure constraint
$$p_i(k) = 0 \quad k \notin \{a_i, ..., d_i\}$$

Assumptions: no other constraints on $p_i \in \mathbb{R}^T$, e.g.,

- discreteness
- interruptibility

2.2 System

- Inflexible demand and generation (renewable): $q(k)$
- Dispatchable generation: $g(k)$
- Flexible EV consumption
- Cost: $f(g(k))$

Standard optimization problem:

$$\begin{aligned} \min_{p,g} \quad & \sum_k f(g(k)) \\ \text{s.t.} \quad & g(k) + q(k) = \sum_i p_i(k) \\ & 0 \leq p_i(k) \leq m_i \\ & p_i(k) = 0, \quad k \notin \{a_i, ..., d_i\} \\ & \sum_k p_i(k) = E_i \end{aligned}$$

3 Scheduling

Standard approach is impractical due to uncertainty

- $q(k)$ is random
- a_i, d_i, E_i are all unknown until task arrival, a_i.

Need a policy:

- Makes decision based on current information, well-suited to uncertainty.
- Convex optimization/multi-period OPF – trajectory, based on predictions
- Dynamic programming/inventory control – policy, incorporates new info online
- Problem: optimal scheduling policy NP-hard
- Follow formulation from [5].

Real-time formulation

- $g(k)$: scheduled in advance, fixed
- $r(k)$: reserves, expensive
- Energy state of load i:

$$e_i(k) = E_i - \sum_{l=a_i}^{k} p_i(l)$$

... remaining energy to deliver.

3.1 Earliest deadline first (EDF)

Algorithm at time k:

1. Observe $q(k)$, new arrivals, departures, energy needs.
2. Set $\hat{g}(k) = q(k) + g(k)$.
3. Compute $e_i(k)$ for all i with $a_i \leq k \leq d_i$.
4. Rank loads by increasing d_i, $i_1, i_2...$
5. Deliver $p_{i_j}(k) = \min(m_{i_j}, e_{i_j}(k), \hat{g}(k))$ to i_j, set $\hat{g}(k) = \hat{g}(k) - p_{i_j}(k)$
6. If $\hat{g}(k) = 0$, and $d_i = k$ and $e_i(k) > 0$ for some load i, satisfy with $r(k)$

Comments

- A type of greedy algorithm
- From Processor Time Allocation in computers
- Optimal when $m_i = \infty$ for all i; unfortunately, unrealistic

3.2 Least laxity first (LLF)

Define laxity:

$$\sigma_i(k) = (d_i - k) - \frac{e_i(k)}{m_i}$$

Interpretation:

- (time to deadline) - (min periods necessary to satisfy remaining load) ...
- ... i.e., number of periods before max charging until departure is required

Algorithm: similar to EDF, but with $\sigma_i(k)$ instead of $e_i(k)$.

1. Observe $q(k)$, new arrivals, departures.
2. Compute $e_i(k)$ and $\sigma_i(k)$ for all i with $a_i \leq k \leq d_i$.
3. Rank loads by increasing $\sigma_i(k)$, $i_1, i_2...$,
4. Deliver $p_{i_j}(k) = \min(m_{i_j}, e_{i_j}(k), \hat{g}(k))$ to i_j, set $\hat{g}(k) = \hat{g}(k) - p_{i_j}(k)$
5. If $\hat{g}(k) = 0$, and $\sigma_i(k) < 0$ for some load, satisfy with $r(k)$

Comments

- Better than EDF; more realistic, same complexity

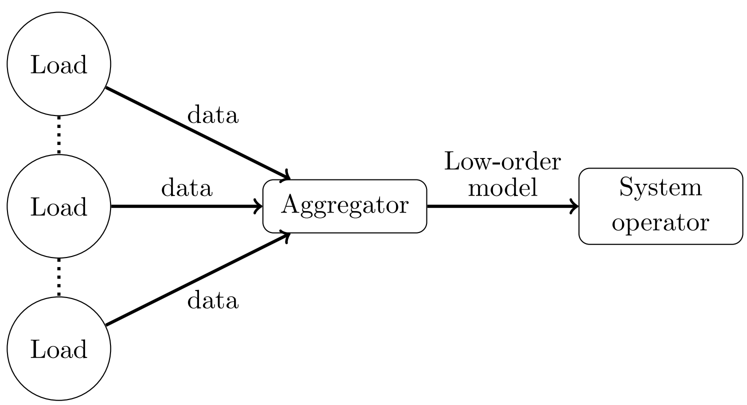
4 Aggregation

Entities

- Many loads
- Aggregator - manages the loads
- System operator - purchases services from aggregator

Problem

- System operator must know aggregator's capabilities to plan decisions.
- System operator doesn't want individual models of 10^6 AC's and heaters.
- Aggregator must provide a reduced order model of the aggregation.



Special case: polytopes

- $Ax \leq b$, $A \in \mathbb{R}^{N \times T}$, $b \in \mathbb{R}^N$
- $x(t)$ is the energy use of the load in time period t
- Bounded $\Rightarrow$ finite power consumption

4.1 Example: charging electric vehicles

Three constraints:

- Energy demand: $\sum_{t=1}^{T} x(t) = E$
- Only charging: $x(t) \geq 0$ for all t
- Power limit: $x(t) \leq P$

Matrix form:

$$A = \begin{bmatrix} 1, ..., 1 \\ -1, ..., -1 \\ -I \\ I \end{bmatrix}, \quad b = \begin{bmatrix} E \\ -E \\ \mathbf{0} \\ \mathbf{P} \end{bmatrix}$$

4.2 Load aggregation as the Minkowski sum

Given M loads, what are the capabilities of the aggregation?

- Loads $i = 1, ..., M$
- Define $Q^i = \{x \mid A^i x \leq b^i\} \subset \mathbb{R}^T$
- Aggregate capabilities:

$$Q = \left\{ x \;\middle|\; x = \sum_{i=1}^{M} x^i,\ x^i \in Q^i,\ i = 1, ..., M \right\} \subset \mathbb{R}^T$$

- Q is the Minkowski sum of the Q^i's.

Representations of polytopes.

- H-representation: half-planes, $a_k x \leq b_k$...rows of $Ax \leq b$
- V-representation: a collection of vertices, $x^k \in \mathbb{R}^T$
- Example: unit square in $\mathbb{R}^2$.
  - Half-plances: $x_1 \geq 0$, $x_2 \geq 0$, $x_1 \leq 1$, $x_2 \leq 1$
  - Vertices: $(0, 0), (1, 0), (0, 1), (1, 1)$

How hard is computing Minkowski sum?

- If all Q^i are in V-representation, easy. Compute each vertex pair $\{x^1 + x^2 \mid x^1 \in Q^1, x^2 \in Q^2\}$, and take convex hull.
- If all Q^i are in H-representation, no known tractable algorithm. No good way to do it in H-representation. Converting from H to V-representation is computationally intractable [3].

Various approximations to this problem

- Outer approximation: If $Q^1 = \{x \mid Ax \leq b^1\}$ and $Q^2 = \{x \mid Ax \leq b^2\}$, an outer approximation is $Q^3 = \{x \mid Ax \leq b^1 + b^2\}$ [1].
- Formulas for electric vehicles [4]
- Formulas for TCLs [2]
- More on TCLs based on homothets [6]

References

[1] S.F. Barot and J.A. Taylor. A concise, approximate representation of a collection of loads described by polytopes. *International Journal of Electrical Power & Energy Systems*, 84:55 – 63, 2017.

[2] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent. Aggregate flexibility of thermostatically controlled loads. *IEEE Transactions on Power Systems*, 30(1):189–198, Jan 2015.

[3] Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled Elbassioni, and Vladimir Gurvich. Generating all vertices of a polyhedron is hard. *Discrete & Computational Geometry*, 39(1-3):174–190, 2008.

[4] A. Nayyar, J.A. Taylor, A. Subramanian, D.S. Callaway, and K. Poolla. Aggregate flexibility of collections of loads. In *Decision and Control (CDC), IEEE 52nd Annual Conference on*, pages 5600–5607, Dec. 2013. Invited.

[5] Anand Subramanian, Manuel J. Garcia, Duncan S. Callaway, Kameshwar Poolla, and Pravin Varaiya. Real-time scheduling of distributed resources. *Smart Grid, IEEE Transactions on*, 4(4):2122–2130, 2013.

[6] L. Zhao, W. Zhang, H. Hao, and K. Kalsi. A geometric approach to aggregate flexibility modeling of thermostatically controlled loads. *IEEE Transactions on Power Systems*, 32(6):4721–4731, Nov 2017.