When to Reap and When to Sow: Lowering Peak Usage With Realistic Batteries

A. Bar-Noy Y. Feng M.P. Johnson O. Liu

CUNY Graduate Center

Research supported by NSF PFI and NYSTAR TIPP, and Gaia Power Technologies

May 27, 2008

Outline

High energy demand

- Key challenge in electricity markets:

High energy demand

- Key challenge in electricity markets:
  - High simultaneous demand
  - Limited supply (per unit time)
- ConEd wants to sell you lots of energy

High energy demand

- Key challenge in electricity markets:
  - High simultaneous demand
  - Limited supply (per unit time)
- ConEd wants to sell you lots of energy but not all right now
- Extreme simultaneous usage is difficult for provider
- Puts strain on grid...

High energy demand

- Key challenge in electricity markets:
  - High simultaneous demand
  - Limited supply (per unit time)
- ConEd wants to sell you lots of energy but not all right now
- Extreme simultaneous usage is difficult for provider
- Puts strain on grid...

Demand charges

- Response by utilities: disincentivize peak usage
- Charge large client sites based on both:
  - How much kWh electricity
  - How quickly (at peak) $\text{kWh/h} = \text{kW}$ power
    - "demand charges"
    - 30-minute rolling averages
- Bill based on: usage charges + demand charges
  - per-kW peak charge ~100x per-kWh usage charge
  - Incentive: spread out usage over time
- Extreme simultaneous usage is difficult for provider
- Places strain on grid

Buffering with batteries

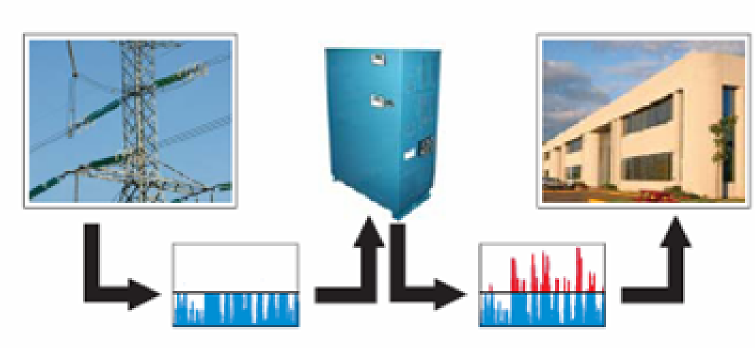

Figure: Gaia PowerTower

Buffering with batteries

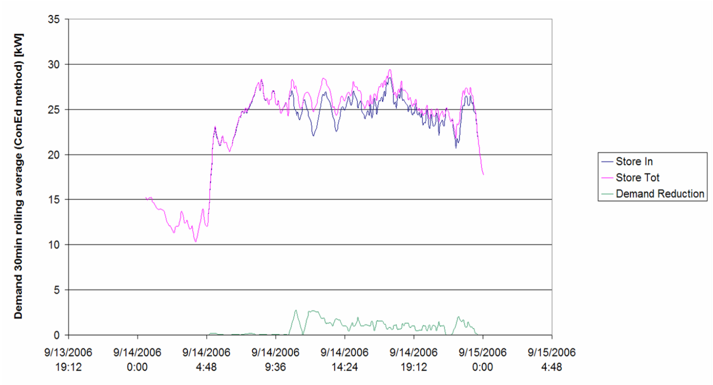

Figure: Peak reduction with Gaia's proprietary algorithms

Other interpretations

- Resource is unsold products
  - xboxes...
- Resource is water
  - Power Tower ? water tower
- ...

Offline problem definition)

- How much to buffer? How much extra to request?
- Goal: make request curve as smooth as possible
  - While always satisfying demand
  - Ideally without wasting any power
- An algorithmic problem:
  - Request nothing waste battery
  - Request too much introduce new peaks

Offline problem definition

- Notation:
  - n timesteps
  - d_i: demand at time i
  - r_i: request at time i
  - $D = \max_i d_i$
  - $R = \max_i r_i$
  - b_i: battery level after time i ($b_0 = B$)
- Goal: make request curve as smooth as possible
  - choose requests d_i to minimize R
  - with no underflow: $b_i \geq 0 \forall i$
  - NB: $b_{i+1} = b_i + r_i - d_i$ (except when underflow/overflow)

Buffering algorithms (offline)

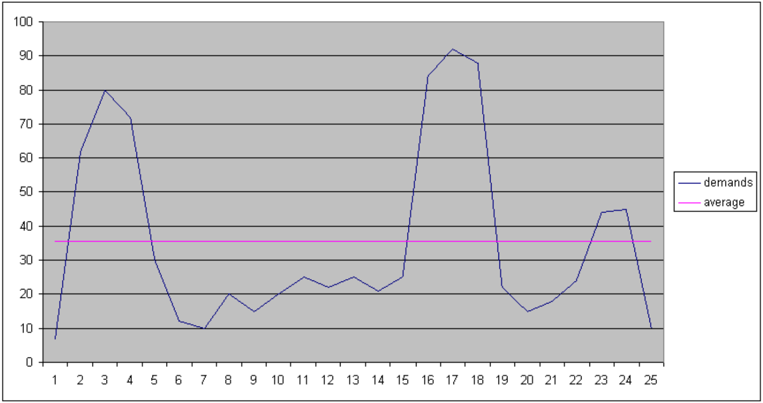

Figure: Demands and mean

Buffering algorithms (offline)

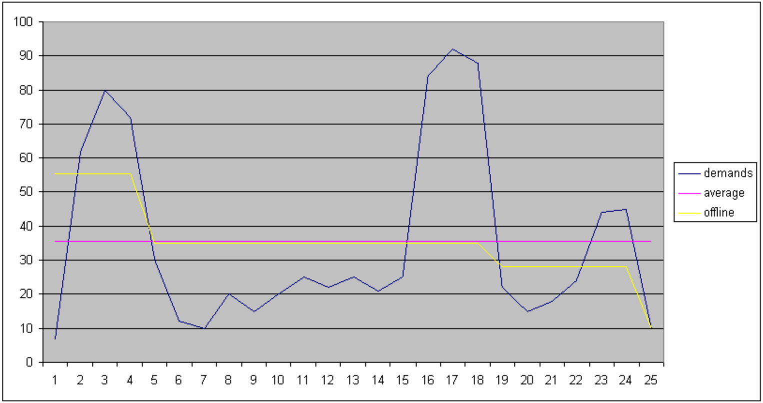

Figure: Demands, mean, and optimal

Threshold algorithms

- All algorithms are based on thresholds
- Offline: global threshold T
- Online: threshold T_i at timestep i
- At each time, (try to) request T_i, and charge/discharge the rest
- Two issues:
  - Overflow: battery too full
  - Underflow: battery empty

Threshold algorithms

for each timeslot i
  if $d_i < T_i$
    charge $min(B - b_i, T_i - d_i)$
  else
    discharge $min(d_i - T_i, b_i)$
    if $d_i - T_i < b_i$
      $T_i \leftarrow T_i + (d_i - T_i - b_i)$

Figure: Threshold algorithm schema

Offline problem

- Solvable in poly-time [Bar-Noy 07]
    - By LP
    - Efficiently: $O(n)$ .. $O(n^2 \log n)$, depending on setting
- Settings:
    - Bounded battery v. unbounded
    - Initial/final conditions essential dont matter
    - With or without *entry loss*
- Settings:
    - Unbounded: find hardest (on avg) prefix
    - Bounded: find hardest (on avg) subsequence
    - Initial/final conditions: slightly preprocess input
    - Lossy battery: instead of average, "generalized average"

Online algorithms

- Change: demands d_i now arrive online
- Goal: competitiveness
  - Constant-factor approx with offline optimal
- One idea: *alpha policy* [someone]
  - Intuition: maybe the future will be like the past
  - Don't know the future, but do know the past
- $\rightarrow$ at each moment, rerun the optimal algorithm on the full history up until now
- Then choose accordingly
  - i.e., request optimal's *max-so-far*
  - This is just maximum mean "maximean"

Request graph

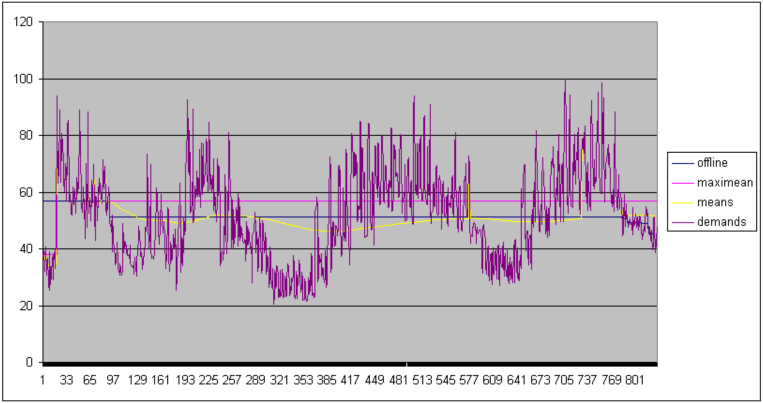

Figure: Demands, mean, and optimal

Request graph: means

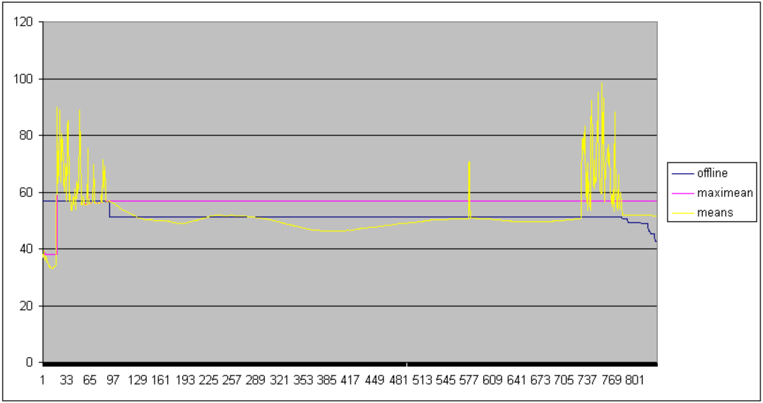

Figure: Demands, mean, and optimal

Request graph: maximean

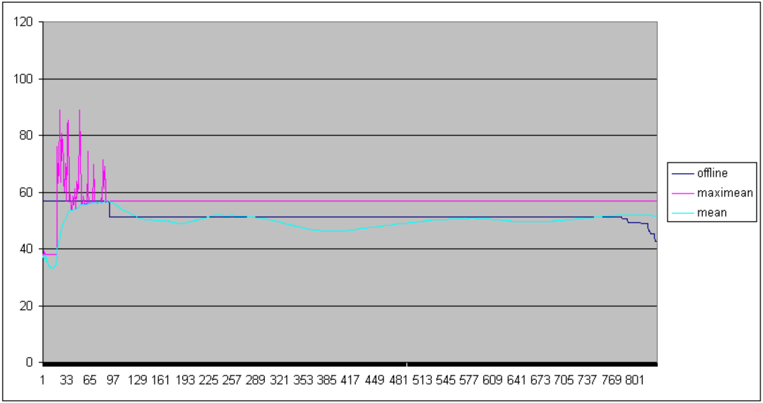

Figure: Demands, mean, and optimal

Competitive online algorithm?

Unfortunately, there are no competitive algorithms for our problem as stated

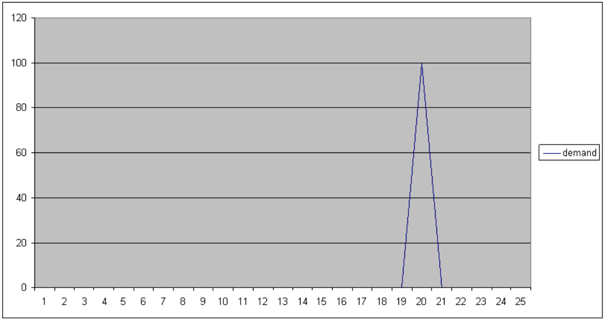

Figure: Competitiveness counterexample

Semi-online model

- Assume we can guess peak demand D
  - (From studying history...)
- Now do "alpha from above"
  - Always request so that savings is exactly $1/H_n$ of optimal savings so far (compared to D)
  - This is H_n-competitive (optimal), for lossless batteries [Bar-Noy07]
  - Proof idea: we have H_n-approx, as long as no *underflow*
- We were previously unable to prove competitiveness for lossy batteries

Factoring-revealing MPs

- Factor-revealing LP technique proves properties of an alg by optimally solving an LP (e.g. maximizing approx ratio) [Jain & Vazirani]
- Lossless setting: we provide a family LPs (indexed by length n)
  - Idea: minimize final battery level, over all instances
  - Non-negative $\longrightarrow$ never runs out
  - Reproves algorithm feasibility (up to length n)
- Lossy setting: we provide factor-revealing quadratically-constrained LPs or mixed-integer LPs
  - Idea: minimize final battery level, over all instances
  - Non-negative $\longrightarrow$ never runs out
  - We solved the QCLPs with solvers on the NEOS server and obtained 0
  - Not a proof! But some "quasi-empirical" evidence

Factoring-revealing MPs

$$
\begin{aligned}
\textbf{min:}\quad & b_{n+1} \\
\textbf{s.t.:}\quad & b_{i+1} = b_i + T_i - d_i, \text{ for all } i \\
& b_i \leq B \\
& T_i = D - (D - opt_i)/H_n, \text{ for all } i \\
& opt_i \geq (1/i)(-B + \textstyle\sum_{j=1}^{i} d_j), \text{ for all } i \\
& \\
& b_1 = B \\
& d_i \leq D, \text{ for all } i \\
& D \geq 0, B = 1
\end{aligned}
$$

Figure: Factor-revealing linear program for lossless batteries (LP1)

Factoring-revealing MPs

min: b_{n+1}

s.t.: $b_{i+1} = b_i + L \cdot ch_i - dis_i$, for all i

$ch_i \cdot dis_i = 0$ (*)

$b_i \leq B$, for all i; $ch_i, dis_i \geq 0$, for all i

$b_1 = B$; $B = 1, D \geq 0$

$D \geq d_i$, for all i

$T_i = D - (D - opt_i)/H_n$

$T_i = d_i - dis_i + ch_i$

$opt_i \geq ga_i$, for all i

$B + L \cdot (\sum_{j=1}^{i} cho_{i,j}) = \sum_{j=1}^{i} diso_{i,j}$, for all i (*)

$cho_{i,j} \cdot diso_{i,j} = 0$, for all $(j,i) : j \leq i$ (*)

$ga_i = d_j - diso_{i,j} + cho_{i,j}$, for all $(j,i) : j \leq i$

$cho_{i,j}, diso_{i,j} \geq 0$, for all $(j,i) : j \leq i$

Figure: Factor-revealing quadratically-constrained LP for lossy batteries

Bolder algorithms

- H_n-approx is X or Y, for typical settings (N_x, N, y)
- Sadly, this is basically the exact performance, not just a guarantee
- Can be greedier and attempt to get better than $1/H_n$ of the savings
  - Dangerous, but can have good performance in practice
- Perhaps we can guess not just D but all demands
  - Then can run optimal alg on predictions

Experiment Setup

- Input
  - A starbucks store
  - A simulated residential user
  - A random user
  - Verification patterns
- Measurement
  - Peak reduction
- Variation
  - Aggressiveness
  - Energy loss factor
  - Error to predict future

Input - Starbucks

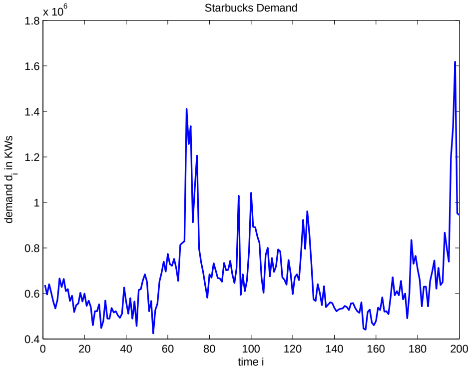

Figure: Demand v.s. time

Input - Residential User

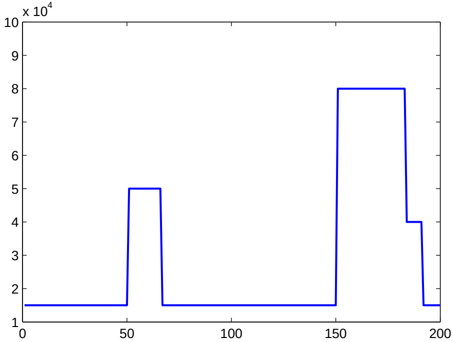

Figure: Demand v.s. time

Peak reduction - Starbucks

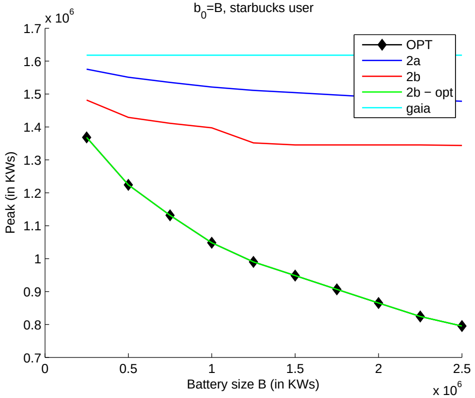

Figure: Peak v.s. B

Peak reduction - Starbucks

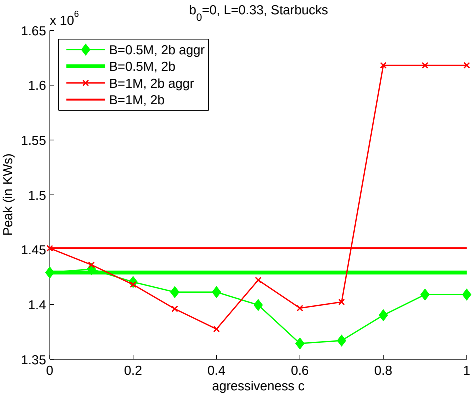

Figure: Peak v.s. Aggressiveness

Peak reduction - Starbucks

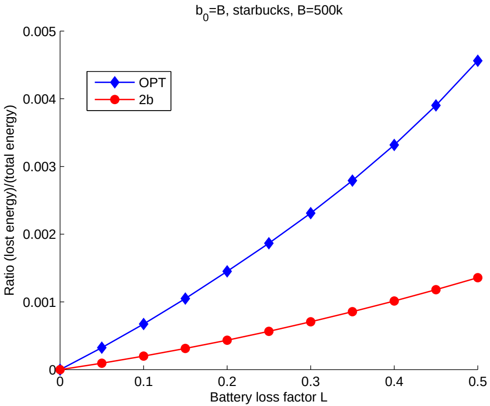

Figure: Peak v.s. Energy loss

Peak reduction - Starbucks

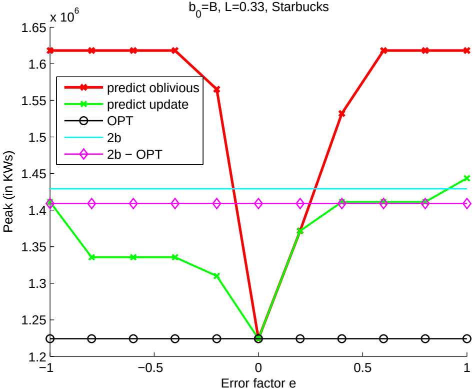

Figure: Peak v.s. Prediction error

Other algorithm models

- Current algorithms are very austere
  - Use nothing but *local* history
- Other resources and techniques available:
  - *Predict from* local history
  - Generic predictions from history data
  - Fine-grain predictions from history data
  - Machine-learning

Other pricing models

- Static variable rates (cell phone plans)
- Dynamic variable rates
- Consumer devices on these models:
  - "Tivo for energy"

Thanks!

- "When to Reap and When to Sow: Lowering Peak Usage With Realistic Batteries", A. Bar-Noy, Y. Feng M.P. Johnson, O. Liu, WEA2008
- "Peak Shaving Through Resource Buffering", A. Bar-Noy, M.P. Johnson, O. Liu, CUNY GC CS TR2007018

blocs

title of the bloc

bloc text

title of the bloc

bloc text

title of the bloc

bloc text