Distributed Computing and Systems
Chalmers university of technology

Balancing energy demand and supply without forecasts: online approaches and algorithms

Giorgos Georgiadis

Overview

- Papers
  - Barker et al (2012), *SmartCap: Flattening peak electricity demand in smart homes*
  - Georgiadis, Papatriantafilou (2014), *Dealing with storage without forecasts in Smart Grids: problem transformation and online scheduling algorithm*
- Focus
  - Online/offline approach
  - Modeling
  - Applicability

Overview (2)

- **Barker et al (2012)**
  - **Premise: home, background loads, slack**
  - **Problem and algorithm**

- Georgiadis, Papatriantafilou (2014)
  - Premise: online, renewables, storage
  - Modeling
  - Greedy algorithm
  - Experiments

Scheduling invisible house loads
Premise

- Load management scheme for flattening household electricity usage or demand
- Modifying background electrical loads that are completely transparent to home occupants and have no impact on their perceived comfort.
  - I.e. air conditioners (A/Cs), refrigerators, freezers, dehumidifiers, heaters
- Online
- Least Slack First (LSF) policy (inspired by the Earliest Deadline First algorithm)

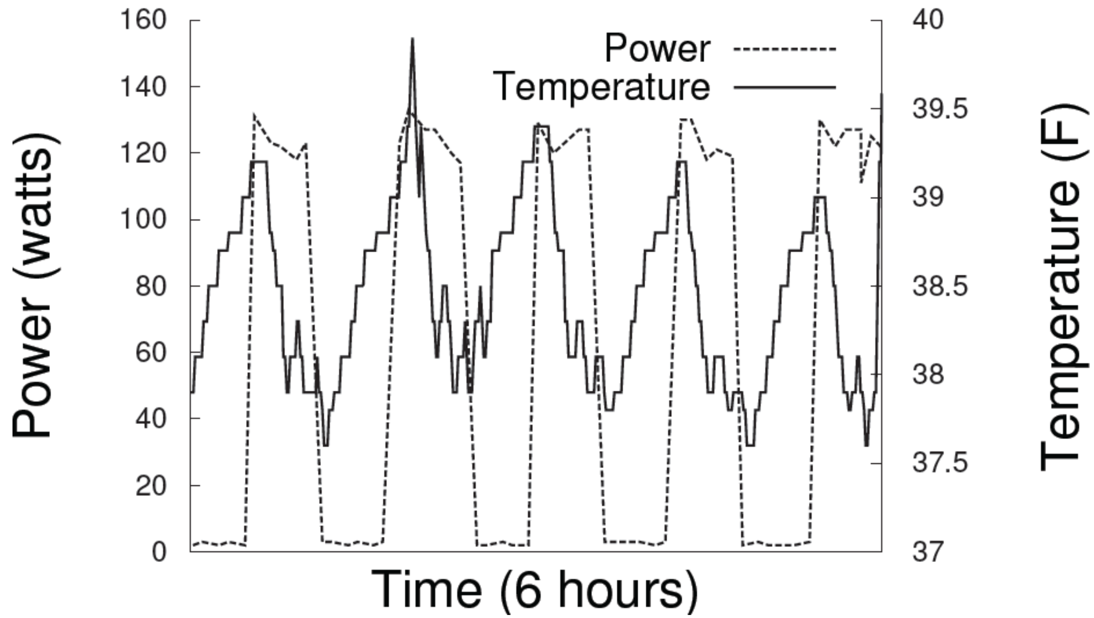
Load	*Peak*	*Average*	*Quantity*
Refrigerator	456W	74W	1
Freezer	437W	82W	1
HRV	1129W	24W	1
Dehumidifier	505W	371W	1
Main A/C	1046W	305W	1
Bedroom A/C 1	571W	280W	1
Bedroom A/C 2	571W	141W	1
Background	**4715W**	**1277W**	7
Interactive	**9963W**	**887W**	85

Barker et al (2012)

Scheduling invisible house loads

Definitions

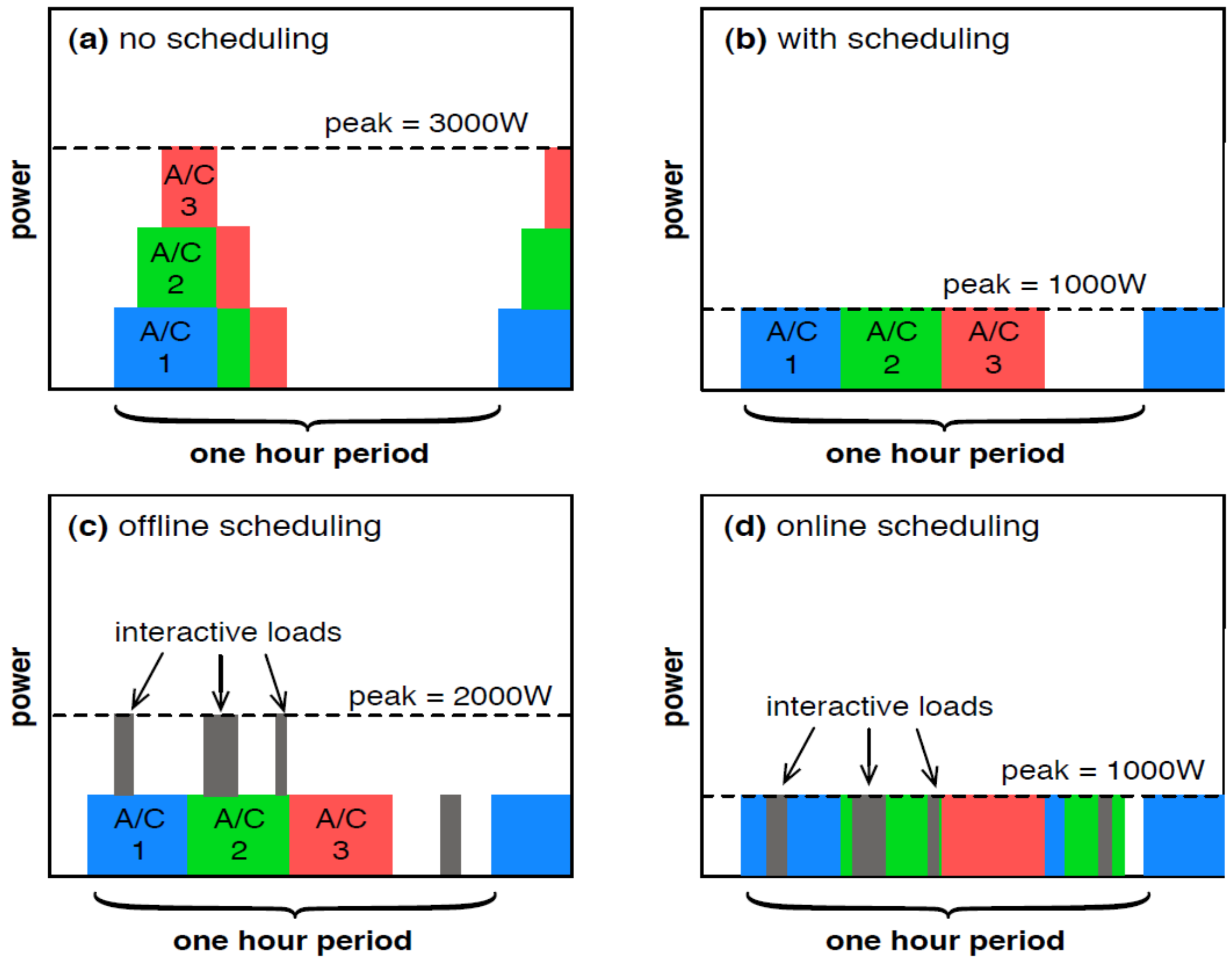
- **Slack:** the remaining length of time the load can be off, i.e., disconnected from power, without assuring that it will violate its objective.
- May change over time (*online problem*)



Barker et al (2012)

Scheduling invisible house loads

Definitions



Barker et al (2012)

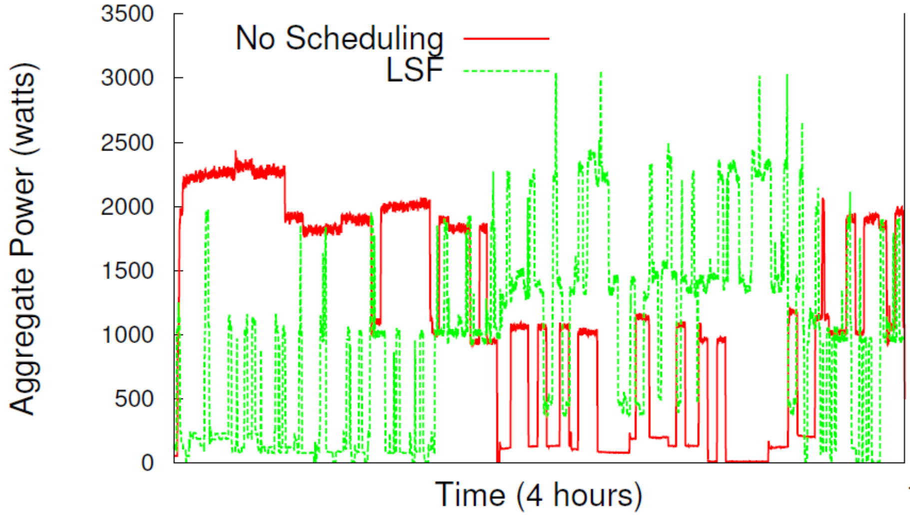
Scheduling invisible house loads

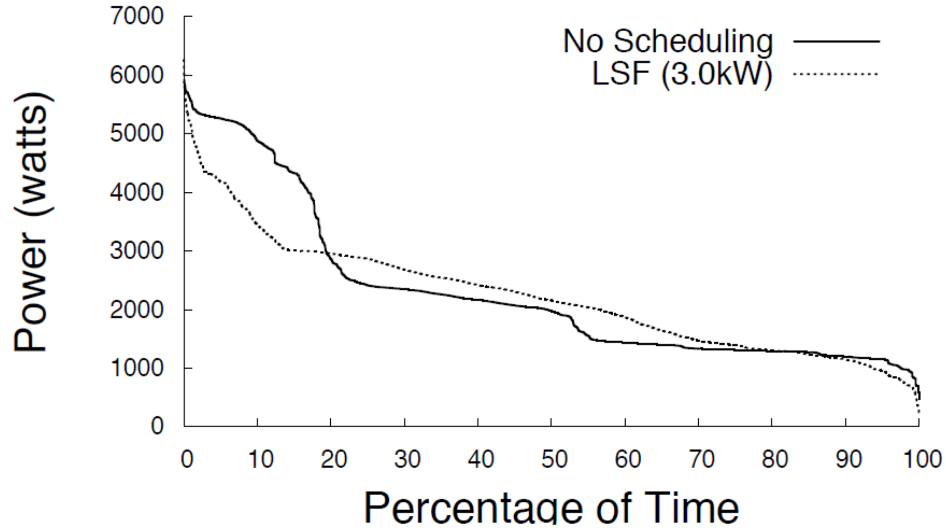
Algorithm

- Least Slack First (LSF)
  - supplies power to loads in ascending order of their current slack value.
- ++ target capacity threshold
  - Once the sum of the background loads' power usage reaches the capacity threshold, the scheduler stops powering additional background loads.
- Concerns
  - Threshold too low: defers too many loads, resulting in their slack values approaching zero together…
  - Threshold too high: power too many background loads at a time. Spikes…

Barker et al (2012)

Scheduling invisible house loads
Some results





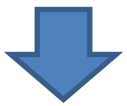
Barker et al (2012)

Overview

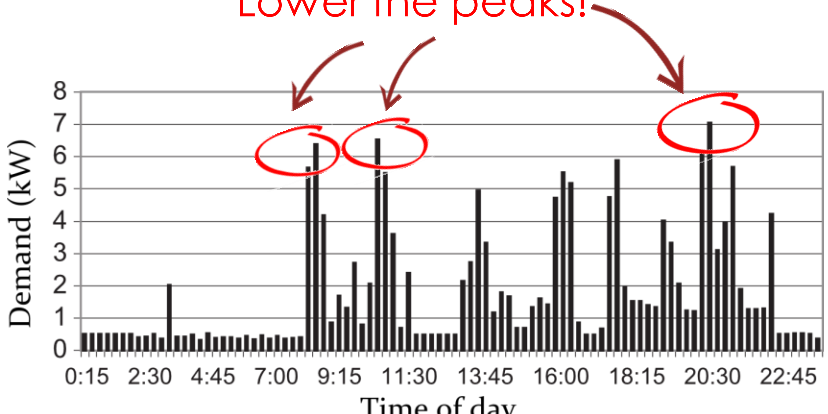
- **Barker et al (2012)**
  - **Premise: home, background loads, slack**
  - **Problem and algorithm**

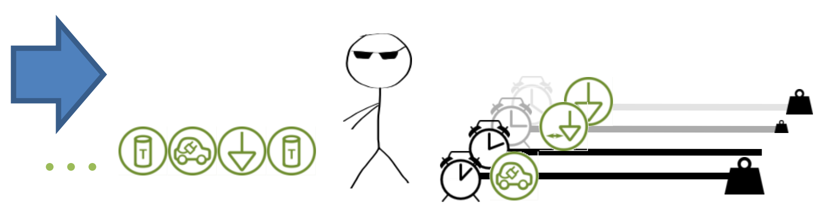
- Georgiadis, Papatriantafilou (2014)
  - Premise: online, renewables, storage
  - Modeling
  - Greedy algorithm
  - Experiments

Online load balancing with storage



Lower the peaks!





Both electrical and thermal energy

Georgiadis, Papatriantafilou (2014)

Definitions

Definition 1 (Unforecasted energy dispatch problem with storage). *Given a distribution system subgrid, we call* unforecasted energy dispatch problem with storage *the problem of dispatching generated electrical and thermal energy to end consumers without using forecasts and by taking into account any storage capabilities present, while trying to minimize peak energy consumption within a given time interval.*

Problem

Premise

Goal

Definition 2 (Online load demand balancing problem with storage) *Let $M_i, i = 0 \dots n-1$ be a set of machines where variable load credit (i.e. storage) can accumulate and $t_0, \dots, t_j, \dots$ be an input task sequence of two task types, simple and storage, with the following properties: each task t_j of both types has load w_j and restrictions on the allowable machines it can run on, while storage tasks additionally create on all machines load credit equal or less to their load (with the possibility of 0 on some but not all machines[2]). We define the* online load demand balancing problem with storage *as the problem of assigning the tasks to the machines while minimizing the maximum load on the machines.*

- Types of tasks
  - Elastic/inelastic, electrical/thermal, storage/simple
- Simplifications and assumptions
  - No distinction of local/global storage
  - Diurnal pattern, hourly slots

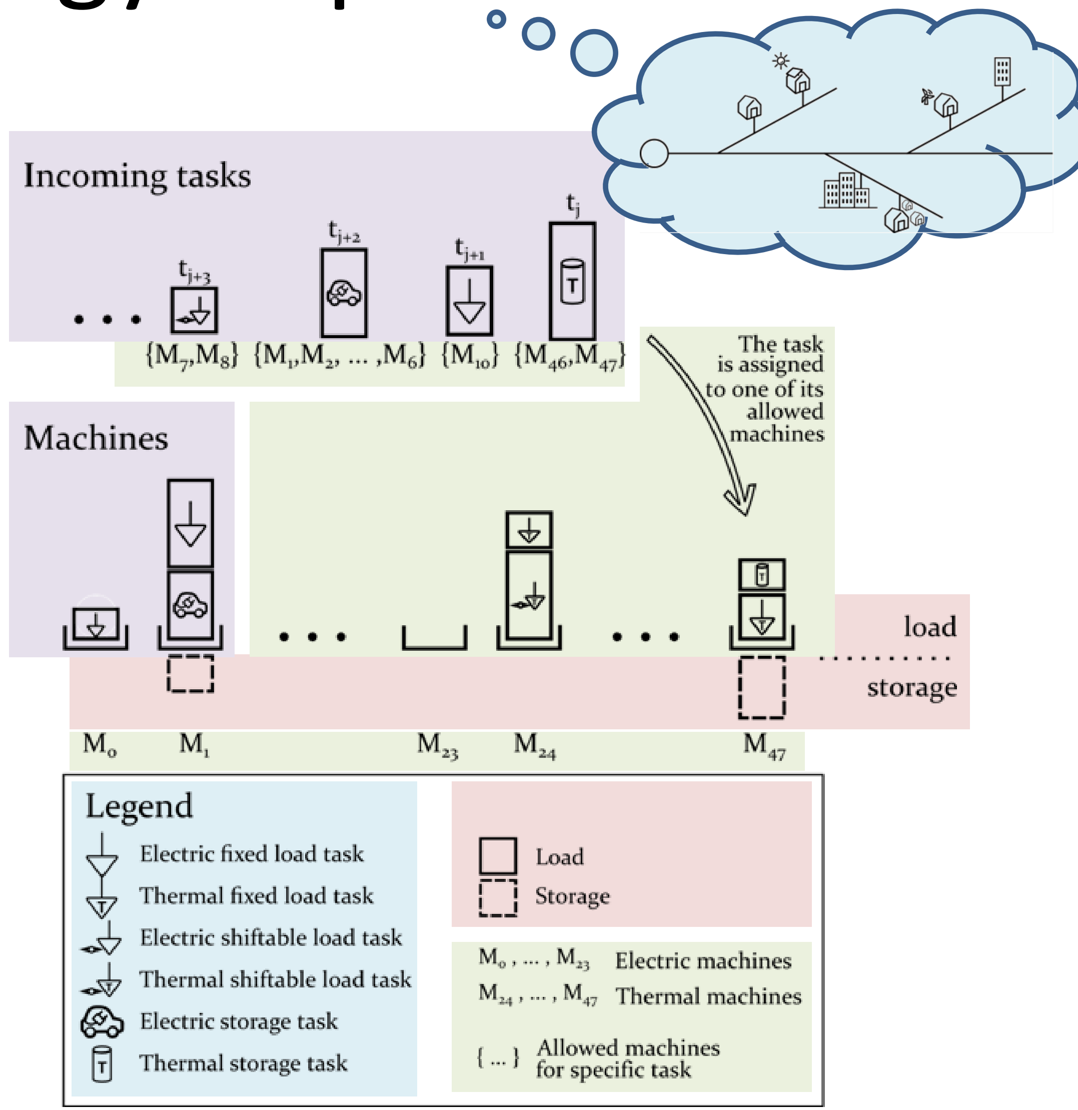
Modeling energy dispatch

Identifying task types

Scheduling tasks to machines

Eliminate time parameter (for flexible tasks)

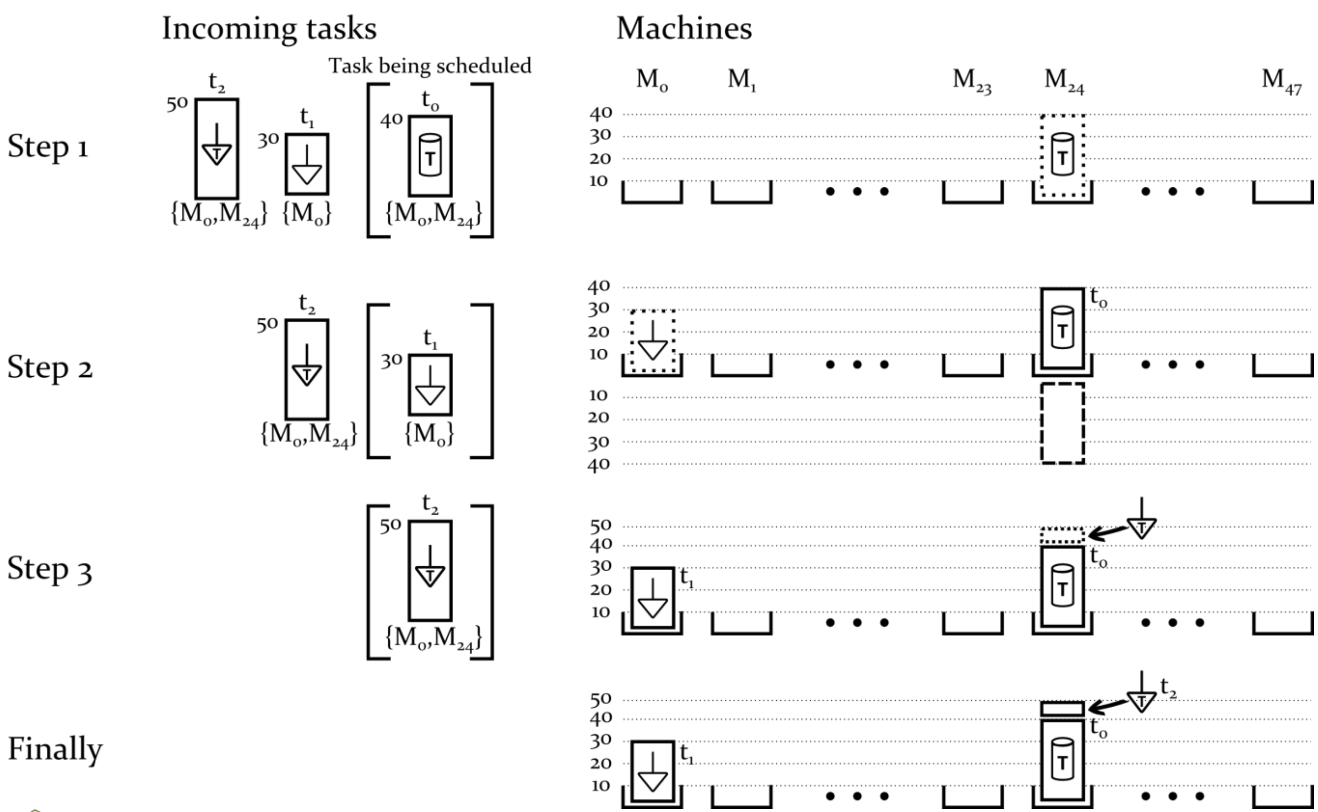
Incorporate storage



Georgiadis, Papatriantafilou (2014)

Demand assignment algorithm

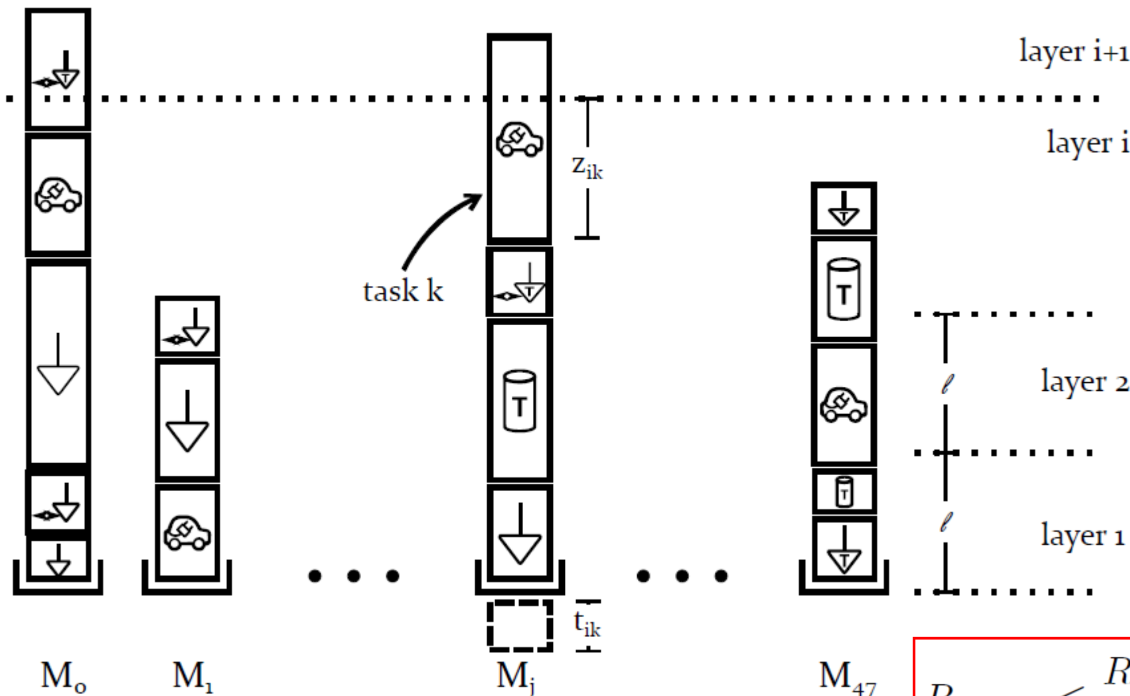
Simple: Assign incoming task to machine with min load-storage difference
Efficient: Within $\lceil \log n \rceil + 1$ of the OPT



Georgiadis, Papatriantafilou (2014)

Algorithm proof (core idea)

Theorem 1 *Algorithm* STORAGEGREEDY *achieves a competitive ratio of* $\lceil \log n \rceil + 1$ *for the online load demand balancing problem with storage, where n is the number of machines.*

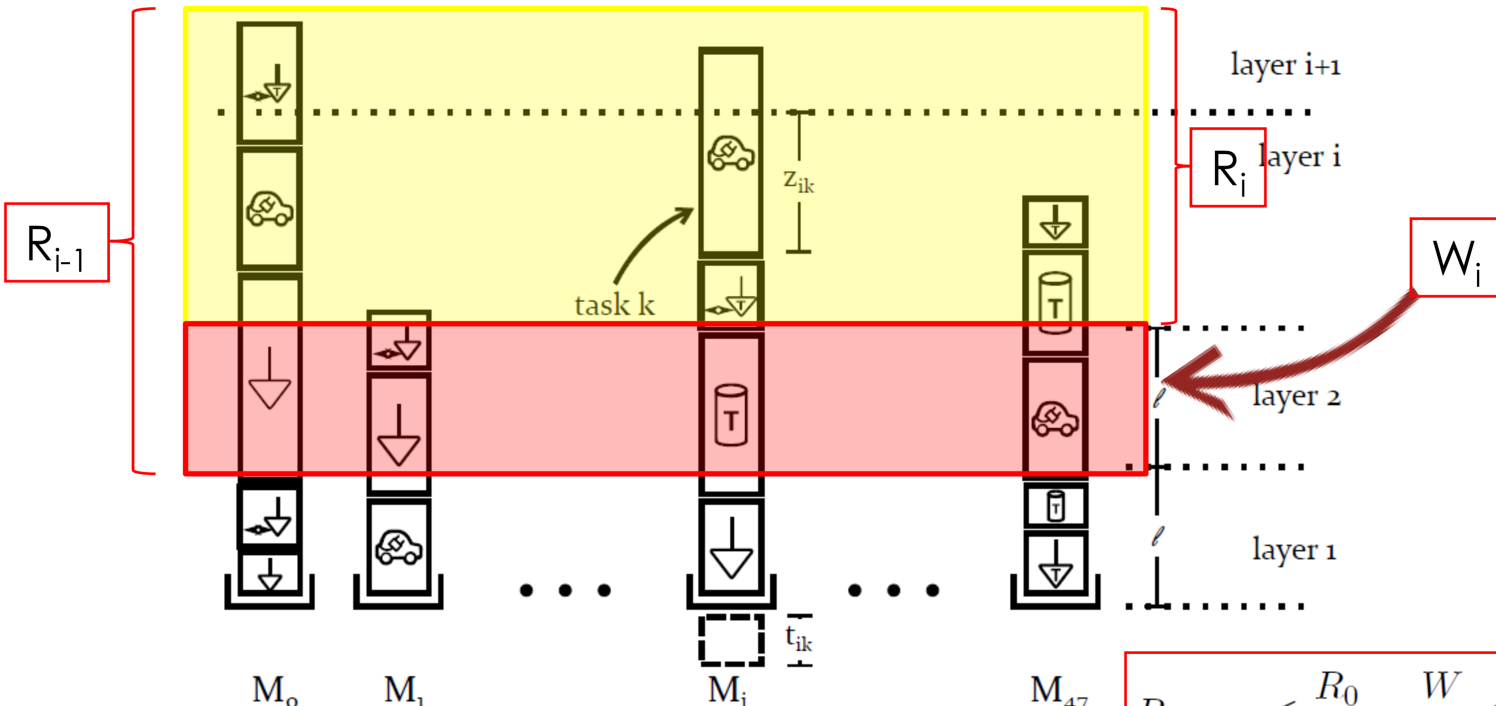


$$R_{\lceil \log n \rceil} \leq \frac{R_0}{n} = \frac{W}{n} \leq OPT(\sigma)$$

- By definition: $W_i = R_{i-1} - R_i$
- If $W_i \geq R_i$ then $R_i \leq R_{i-1} - R_i \Rightarrow R_i \leq \frac{R_{i-1}}{2}$
- Goal: prove $W_i \geq R_i$

Algorithm proof (core idea)

Theorem 1 *Algorithm* STORAGEGREEDY *achieves a competitive ratio of* $\lceil \log n \rceil + 1$ *for the online load demand balancing problem with storage, where* n *is the number of machines.*

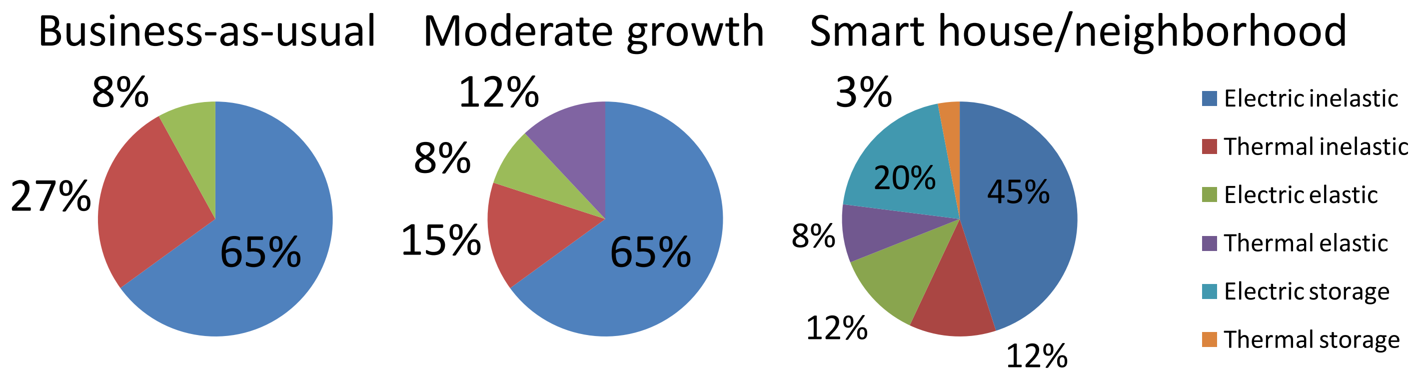


$$R_{\lceil \log n \rceil} \leq \frac{R_0}{n} = \frac{W}{n} \leq OPT(\sigma)$$

- By definition: $W_i = R_{i-1} - R_i$
- If $W_i \geq R_i$ then $R_i \leq R_{i-1} - R_i \Rightarrow R_i \leq \frac{R_{i-1}}{2}$
- Goal: prove $W_i \geq R_i$

Experimental setup

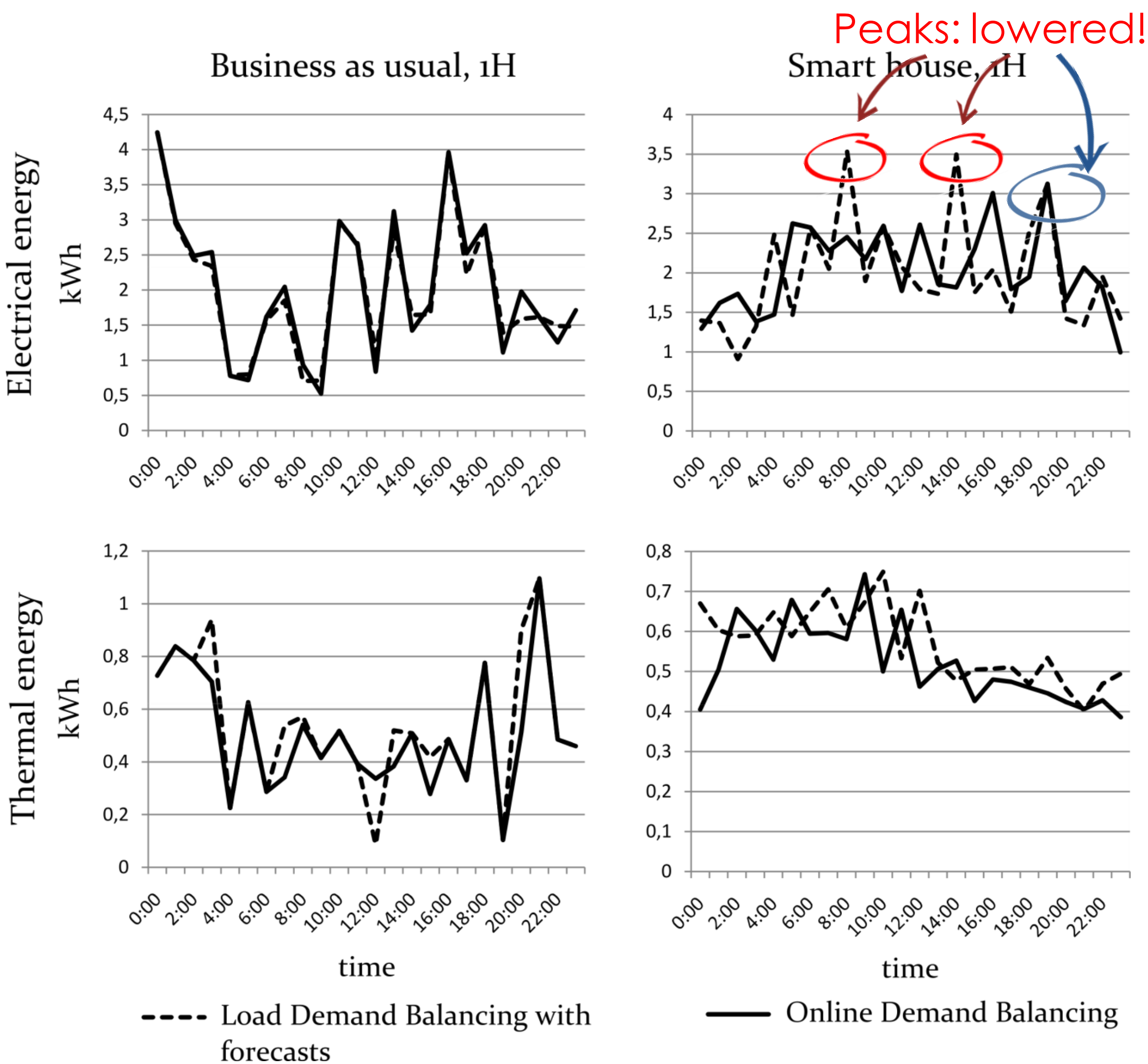
- Two axis
  1) Demand mix



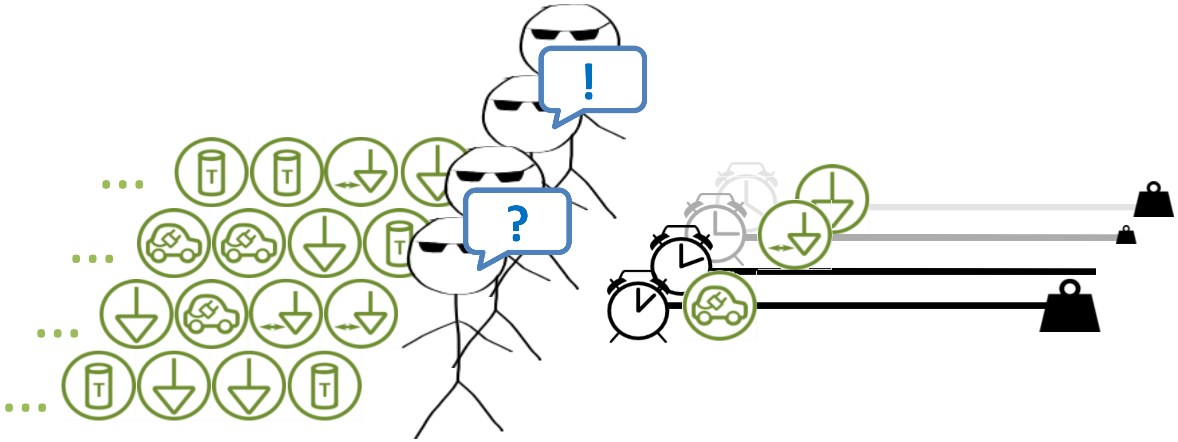
  2) Number and type of households

- Comparison
  - **Longest Processing Time (LPT):** *sorts tasks by decreasing processing time and then assigns each task to the machine that has the least load (breaking ties arbitrarily)*

Experimental results



What's next?



- Mixed algorithms
  - Communication with global optimizer
  - Allow budget for scheduling over forecasted
  - Call optimizer when over budget

- Strategic games
- New modeling extensions/applications

Summary

- Background loads, threshold, online-ness (forecasts?)

- *In focus: online load balancing with storage*
  - Energy dispatch: assignment/matching problem with guarantees
  - Transformation of time and unforecastability: resource allocation
  - High quality solution: analytical results and experiments based on real data

- Next: cooperation, strategies, extensions