Optimal Demand-Side Energy Management Under Real-time Demand-Response Pricing

Jin Xiao¹, Jae Yoon Chung², Jian Li², Raouf Boutaba¹,³, and James Won-Ki Hong¹,²

¹IT Convergence Engineering, POSTECH, Pohang, Korea
²Dept. of Computer Science and Engineering, POSTECH, Pohang, Korea
³School of Computer Science, University of Waterloo, Canada

Introduction and Motivation

The ability to reduce electricity usage and wastage through better demand-side management and control is considered a key solution ingredient to the global energy crisis. One effective measure that has been put into place in many countries around the globe is the Demand Response (DR) program. Performance in many countries around the globe is the Demand Response (DR) program.

Key Deterring Factors of Current Grid

- Lack of information
- Lack of smart planning
- Customers are risk-averse

Research Goals

- Design and Implement Green-Home Service (GHS) architecture to provide advanced metering and control
- Design scheduling algorithm to provide decision making capabilities

Demand-side Energy Management

- minMax Algorithm Overview
  - Request d with a starting time s and an ending time f
  - Schedule d to somewhere within the specified time frame
  - Assign d to the time slots with the lowest cost among the candidate timeslots

- Algorithm Improvement using Battery
  - Input: the schedule produced by the minMax algorithm
  - Find the peak cumulative cost
  - Shifts part of its demand forward in time filling in the time slots that are under-utilized
  - Repeat operation 2) and 3) until no shifting can be performed.

minMax Algorithm

Algorithm 1: minMax Scheduling

Input:
- n number of time slots
- m number of demands
- Time slot where 1 ≤ t ≤ n
- Consumer demand d_j = \{s_j, f_j, r_j, l_j\} ∈ D where 1 ≤ j ≤ m
- s ∈ T is start time, f ∈ T is finish time, r is consuming rate, l is number of time slots of demand.
- Cost for demand rate r and slot s: \(B(s)\)

Output:
- scheduleMatrix[n][m] such that scheduleMatrix[i][j] is the energy cost of demand at time slot t

1. Initialize scheduleMatrix[n][m] to zeros
2. Initialize r_j to zeros
3. Assign tasks in D with fixed schedules (i.e., f - t + 1 = l) to timeslots, and adjust r accordingly
4. Sort D based on r as the major key in descending order
5. Sort D based on t as the minor key in descending order
6. for i = 1 to n do
7.     minMax = ∞
8.     schedulingSlot = 0
9.     for t = s to f, l + 1 do
10.    minMaxAF = \max(\sum c_i + P(t), c_{i+1} + P_{i+1}(t), c_{i+1} + P_{i+1}(r_j))
11.    if minMaxAF < minMax then
12.        minMax = minMaxAF
13.        schedulingSlot = t
14.    end if
15.    end for
16.    for j = schedulingSlot to l - 1 do
17.        scheduleMatrix[i][j] ← P_i(t)
18.    end for
19. end for

GHS Implementation

GHS Components

- Server Stub: Web service interface to the client applications and power utility
- Services: a collection of GHS services such as metering, decision engine, service security, etc
- Repository: manages metering data as well as device specific information such as the adaptor-to-appliance mapping.
- Adapter: generates communication messages depending on the manufacturer’s message format and data model. Multiplicity of communication technologies are enabled through the use of appliance specific adapters.

Conclusion & Future Work

- Conclusion
  - Designed and implemented GHS
  - Modeled the demand-side energy management problem (NP-hard)
  - Designed a scheduling algorithm for demand side energy management
  - Showed that our algorithm can find near-optimal
  - Showed the effect of battery on demand smoothing

- Future Work
  - Integrate the minMax algorithm in the Green-Home Service implementation
  - Conduct field-test experiment in real home and large enterprise settings.