Near-Optimal Scheduling of Residential Smart Home Appliances Using Heuristic Approach

Chris Ogwumike*, Michael Short and Mouloud Denai†

*School of Science and Engineering, Teesside University, Middlesborough, United Kingdom
†School of Science and Technology, University of Hertfordshire, Hatfield, United Kingdom
C.Ogwumike@tees.ac.uk, M.Short@tees.ac.uk, M.Denai@herts.ac.uk

Abstract—This paper presents an efficient heuristic approach for scheduling residential smart home appliances. Using available hourly prices for electricity, the starting times of a supplied set of appliances are optimized so that the economic cost of the energy consumed is reduced, while satisfying the operational and peak power constraints. The algorithm schedules appliances one after the other based on a greedy strategy. The heuristic (c.f. exact) approach is taken to reduce the computational burden to a level allowing re-optimization to take place at regular intervals by a modest computing device without specialized software, which could be embedded in a smart meter. The proposed algorithm is evaluated through a preliminary experimental study comparing the obtained costs and computation times with an exact algorithm. Results indicate that the obtained cost was within 5% of the optimal cost, while the computation time reduced by exponential factors.

Keywords—Energy optimization; Residential appliances; Heuristic algorithm; Demand Response; Smart home

I. INTRODUCTION

Efficient load scheduling is critical to achieve Demand Response (DR) at the consumption level of smart grid. The objective of scheduling residential load is to balance the controllable loads to minimize the cost of energy consumed. One important approach to load scheduling is the engagement of consumers to participate in the decision-making aimed at using the energy infrastructure more efficiently. Economic time varying tariff incentives such as Time-of-Use Pricing (TOUP) and Real Time Pricing (RTP) encourage the consumers to shift the energy consumption from peak to off-peak periods [1]. However, it is unrealistic for consumers to keep track of hourly varying pricing of electricity so as to manually schedule the appliances. As such, energy management system in form of an automatic decision support is needed to provide scheduling advice and recommendation to the electricity consumers.

Load scheduling is shown to be NP-Complete problem, whose optimal solution cannot be found in a polynomial time [2]. Mathematical optimization paradigms such as Linear programming (LP) is useful in task scheduling problems across various disciplines, particularly in engineering related fields such as energy, communication and transportation industries etc. [3]. With Integer Programming (IP), additional constraints are added to linear programming such that some or all of the variables take on integer values. This increases the scheduling problem and makes it difficult to solve within polynomial time. This means that if a polynomial time algorithm finds optimal solution to the load scheduling problem in a polynomial time, then all the NP-Complete problems can be solved in a polynomial time. However, it is generally thought that no such algorithm exists, although it cannot yet be ruled out [4]. Exact methods for NP-complete and NP-hard problems are only useful for solving problems of reduced size since they require excessive computational time. Since smart home appliance scheduling, like other forms of real-world scheduling problem, is effected by uncertainties, regular re-optimization in a ‘rolling-horizon’ frame work can be beneficial. For example, small changes in the electrical grid frequency - within national limits - can affect power consumption profiles by a small but not negligible margin; wear-and-tear of physical components such as motor brushes can also cause deviations from nominal appliance behavior. In this paper we therefore propose to sacrifice optimality at the expense of excessive computation time for near-optimality with low computation time, such that regular re-optimization with updated state information can take place.

A heuristic algorithm applies a set of rules based on the modelling of processes and is realized in individual steps in a problem solution [4]. The advantage of a heuristic approach is that a ‘good’, but not necessarily optimal solution to the optimization problem can be found in a reasonable time. Therefore, heuristic scheduling algorithms are very important to achieve faster solutions. In the context of the current paper, a heuristic solution which is simple enough to be embedded in a microcontroller or computer for the purposes of a consumer decision support system is desired. This decision support system should not require specialized integer programming software, and be implementable on a small computing device such as a smart meter.

Previous studies have considered heuristic task scheduling algorithms with an emphasis on load balancing. Ni et al. [5] developed a heuristic task algorithm to obtain a scheduling strategy by employing a mean load as heuristic information. Tasks are reassigned between two machines to raise the loads of the machines with lower-load and reduce that of the machines with higher-loads under the mean load heuristic. They argued that the efficiency of the algorithm is significantly improved and can almost achieve an optimal scheduling strategy. Shuhui et al. [6] compares optimal and heuristic Demand Response (DR) algorithms through a computational experiment strategy. The greedy heuristic which combines Min-min and Max-min heuristic by using better solution is discussed in [7]. Min-min algorithm selects
tasks capable of finishing at the minimum time while the Max-
mix algorithm selects the tasks that take a maximum time to
finish the operation of the machines [8]. Barbato and
Carpentieri [9] proposed a set of heuristics and an opti-
mization model for the online demand side management.
They combined online and offline approaches to control home
appliances and energy storage systems for efficient
management of the energy resources.

In this paper, we present exact and heuristic algorithms to
schedule a set of smart home loads and optimize the operating
cost. Our algorithms determine the best hourly timeslot within
the simulation horizon to schedule the appliances while also
adhering to the appliance operational, sequential and peak
constraints. The exact algorithm takes possible combination of
costs across all hourly \( H \) timeslots for all appliances to
determine the best scheduling with an exponential runtime
complexity \( O(H^N) \). In the heuristic algorithm, the appliances
are scheduled in decreasing priority of user preference using
the greedy strategy such that an appliance is not changed
anymore once scheduled; this gives runtime complexity
\( O(HN) \). The experimental results show that the optimality gap
between our proposed heuristic with the exact algorithm was
not more than 5% on the range of problems considered in a
prototype tests using representative data. The runtime, as may be
expected, was exponentially faster.

The remainder of the paper is organized as follows: Section II
describes the optimization problem. Section III
presents the scheduling algorithm, explains the functionality
and step-by-step processes. Experiments, results and
discussions on the exemplary numerical studies of the
algorithms are conducted in section IV. Section V is the
conclusion.

II. PROBLEM DESCRIPTION

We assume electricity supplied to the residential home is
generated through a variety of means, including traditional
forms of generation (coal, gas, etc) along with renewable
means (solar, wind etc), with an hourly price that reflects the
value of the energy during that hour. The hourly price will
typically be a function of the ‘spot price’ for energy that the
distribution/supply company for the residence has paid to
secure the supply during that hour, and is advertised to the
resident 24 hours in advance. The appliance scheduling
problem can be modeled mathematically as a Linear or Mixed
Integer Linear Programming (MILP) as shown in [10]-[12].
Using a similar approach to Sou et al [13], we model the
appliance operations as a set of uninterruptible (non-
preemptable) and sequential energy phases, each with a
specified valid timeslot, length of operation and power
consumption profile. This involves the mathematical
formulation of the objective function and constraints.

Given the number of appliance \( i \) denoted as \( N \) and the
number of phase \( j \) denoted as \( ni \). The scheduling execution
period is discretized into \( m \) uniform time slots \( k \), such that
\( k \in [1, m] \). Let the phase for each appliance \( i \in [1, N] \) be
indexed by \( j \in [1, ni] \). The power assigned to every phase \( j \) of
an appliance \( i \) at any timeslot \( k \) is denoted by \( P_{i,j}^k \) while the
spot price of electricity at any timeslot is \( C_k \). The objective
function to be minimized is given by:

\[
  j = \min \sum_{k=1}^{m} C_k (\sum_{i=1}^{N} \sum_{j=1}^{ni} P_{i,j}^k) \tag{1}
\]

Subject to the following constraints [13]:

**Phase Energy constraint**: ensures the energy allocated for
multiple phase \( j \) of an appliance \( i \) fulfill the energy
requirements \( E_{i,j} \)

\[
  \sum_{k=1}^{m} P_{i,j}^k = E_{i,j} \quad \forall \ i,j
\]

**Peak power constraint**: ensures the maximum power
consumption for all the appliances at any time slot \( k \) does not
exceed the peak power limit

\[
  \sum_{i=1}^{m} P_{i,j}^k \leq \text{Peak}^k \quad \forall \ k
\]

where \( \text{Peak}^k \) is the peak demand response signal
provided by the utility power provider.

**Sequential processing constraint**: An energy phase
operation cannot start until the previous phase has finished.
Auxiliary decision variables \( S^k_{i,j} \) and \( X^k_{i,j} \) are used to describe
the constraint

\[
  X^k_{i,j} \leq S^k_{i,(j-1)} \forall \ i,k, \ \forall \ j = 2,..ni \tag{4}
\]

**Energy phase duration constraint**: ensures the energy
phase operation period of an appliance operates within a given
time limit.

\[
  T^i_{i,j} \leq \sum_{k=1}^{m} X^k_{i,j} \leq \bar{T}^i_{i,j} \quad \forall \ i,j
\]

where \( T^i_{i,j} \) and \( \bar{T}^i_{i,j} \) are the lower and upper limits for the
number of timeslots allocated to the phase \( j \) of an appliance \( i \)

**Uninterruptible Operation constraint**: ensures that energy
phase \( j \) of an appliance \( i \) finish operation without interruption.

\[
  X^k_{i,j} \leq 1 - S^k_{i,j} \quad \forall \ i,j,k \tag{6}
\]

During timeslot \( k \), \( S_{i,j} = 1 \) if energy phase \( j \) of an appliance \( i \)
is already finished. As such, \( X^k_{i,j} \) must be zero.

**Between-phase delay constraint**: imposes a specified
amount of time-delay at the end of one energy phase before
the start of the next phase.

\[
  D^i_{i,j} \leq \sum_{k=1}^{m} t^k_{i,j} \leq \bar{D}^i_{i,j} \quad \forall \ i, \forall \ j = 2,..ni \tag{7}
\]

where \( D^i_{i,j} \) and \( \bar{D}^i_{i,j} \) are the upper and lower bounds that
describes the between-phase delay.

**User Time preference**: ensures appliance does not operate
outside of the time preference interval.

\[
  X^k_{i,j} \leq TP^k_{i} \quad \forall \ i,j,k \tag{8}
\]
where $TP_k^i$ is the time interval and it’s equal to zero if and only if energy phase $j$ of an appliance $i$ cannot be processed at any timeslot $k$.

The MILP formulations above can be solved using the IBM ILOG CPLEX and the YALMIP interface to Matlab; however it is only feasible for relatively small instances of about 4 to 6 appliances [14]. Algorithms such as cutting plane method and branch and bound method [15] can also be used to reduce the average-case complexity.

III. SCHEDULING ALGORITHMS

In this section, we present our scheduling algorithms which use the appliance start time as the decision variable and other parameters such as power, length of timeslot, delay etc are the inputs which define the problem instance. Load power profiles and other parameters are assigned to appliances based on data extracted from surveys, communications and various sheets for smart home appliances in the appendix [16]. Some modern electric vehicles contain a lower capacity charger and will charge and deliver 38 miles of range in 4 hours using 4KWh charging system [17]. In this research, we assign 2000W of power to electric vehicle with 2 hours length of timeslot. The peak power constraint is the maximum instantaneous power that may be drawn by the load. In this paper, we assume this limit to be 5500Wh.

A. Exact Algorithm

The algorithm starts with an initial start time for every appliance which is in the operation time window. A cost matrix is created for all timeslots of the schedulable appliances, and the start times are exhaustively searched to obtain the best possible combination (in terms of economic cost) which also satisfies the constraints. Power consumed is calculated such that if the power assigned to every phase of an appliance is less than the peak power demand in (3), the algorithm calculates the total cost of operation for the appliance. If total cost of the appliance is less than the initial calculated total cost, then the initial total cost is updated; followed by the updates of the optimal start times.

**Algorithm 1.** Exact method

```
for all appliances $i = 1$ to $N$  do
   for all timeslots $k = 1$ to $24$  do
      *Create cost matrix for all timeslots for each appliance
   end for
end for

for all timeslot for $i^{th}$ appliance  do
   for all timeslot for $2^{nd}$ appliance do
      .
      .
      .
      for all timeslot for $N^{th}$ appliance  do
*Calc. the total power for each timeslot
   if Power $\leq$ Peak  then
      *Calc. the total cost
      if total cost $< \text{initial total cost}$ then
         *Update the initial total cost
         *Update the optimal start time
      end if
   end if
end for
```

B. Heuristic algorithm

The heuristic algorithm decides the best time to schedule the appliances in an order specified by the user, which can allow for an implicit priority ordering amongst appliances in terms of constraint satisfaction (although this was not explicitly considered in this paper). The appliances are scheduled sequentially based on a greedy strategy without back-tracking. The start time which minimizes the total cost across the horizon is selected from the set of feasible start times (i.e. those which satisfy the given constraints). An appliance start time, once fixed, is not changed. The remaining appliances are scheduled on top of the current solution; the set of appliances already issued with fixed start times.

The algorithm basically performs three main control loop processes. The first loop in variable $i$ iterates over all the number $N$ of appliances and initializes the best time for the appliance scheduling. The second loop iterates over the timeslots $k$ and initializes the appliance execution time window. The third iterates over the number of phases $ni$ of appliance $i$. Delay is included in the phase loop and can be imposed in-between the execution start time of any phase of an appliance or between appliances such that $\text{StartTime} = \text{StartTime} + (\text{Delay}(i, j))$. The algorithm calculates the total cost and total power of individual appliances and ensures it doesn’t exceed the peak power demand. The optimized appliance solution is updated.

**Algorithm 2.** Heuristic Greedy Algorithm

```
U $\leftarrow$ Vector of sorted appliances by User preference
for all appliances $i \in U$  do
   *Initialize the best appliance scheduling solution
   for all time slots $k \in [1, m]$  do
```

We present numerical studies to evaluate the performance of the proposed algorithms for the residential smart home appliances. All experiments are performed on a HP® PC with an Intel Core i5 CPU, 3.40GHz speed and 6GB of memory.

A. Optimal cost solution using the proposed algorithms based on hourly system price

In this experiment, we investigate our algorithms using the system price of electricity in fig 1 above, for the Scandinavian electricity market from the 29th September 2014 to the 5th October 2014 [18]. The hourly system price is used for one week of optimization to obtain the cost effective scheduling for the exact and heuristic algorithms. Optimization was carried out once every 24 hours in these tests. The scheduling system is composed of four controllable smart home appliances: washing machine, tumble dryer, dish washer & electric vehicle. In addition, 2 uncontrollable appliances: electric heater and television are also included such that the user imposes an adjustable constraint (Adjustable = 0) if and only if none of the uncontrollable appliances can be processed during the timeslot k. The execution period of the appliance operation is within the time window for every appliance. Time preference constraint in (8) is imposed by the household consumer such that dish washer and electric vehicle are to run any time between 5pm to 11pm and 1am to 5am respectively. The washing machine and tumble dryer can operate anytime between 10am to 11pm. However, Appliance operation constraint is applied such that washing machines phases must finish before tumble dryer phases start.

After extensive simulation, the periodic optimized cost solutions for exact and heuristic algorithms are plotted against each other as shown in fig 2. Percentage cost difference in comparison of heuristic and exact algorithms are as follows: (0.2116 – 0.2100)/0.2116 = 0.0076%. The result verifies that while the exact algorithm achieves an optimal cost solution to the problem, the percentage cost difference with the heuristic is near to optimality.

B. Testing the proposed algorithms based on spot prices for New York City (NYC) and Denmark (DNK)

Our proposed algorithms are further tested to determine the cost of energy consumption for a typical day in the fall (autumn) period using the spot prices of electricity for New York City (NYC) and Denmark (DNK)
York City and Denmark on 5th November 2014. See fig. 3 and 4 for the respective NYC and DNK spot prices.

The results of the total cost obtained for both instances of spot prices using both algorithms are shown in the table 1 below:

### TABLE I. RESULTS OF TOTAL ENERGY CONSUMPTION FOR HEURISTIC AND EXACT WITH SPOT PRICES FOR NEW YORK CITY (NYC) AND DENMARK (DNK)

<table>
<thead>
<tr>
<th>Algorithms with spot prices</th>
<th>Cost (USD)</th>
<th>Cost (EUR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic: with NYC</td>
<td>0.2841</td>
<td>[ 8.523 per month ]</td>
</tr>
<tr>
<td>Exact: with NYC</td>
<td>0.2751</td>
<td>[ 8.253 per month ]</td>
</tr>
<tr>
<td>Heuristic: with DNK</td>
<td>0.2670</td>
<td>[ 8.01 per month ]</td>
</tr>
<tr>
<td>Exact: with DNK</td>
<td>0.2565</td>
<td>[ 7.695 per month ]</td>
</tr>
</tbody>
</table>

### C. CPU Time Scalability

In testing the solving time for this scheduling problem, the number of timeslots over the horizon and number of controllable appliances are the major variables dictating the problem size and hence algorithm execution time. Table II shows the one week periodic re-optimized average solving time for the heuristic and exact algorithms against increases in the number of controllable appliances. The controllable appliances are increased from 4 to 10 with random (but representative) energy and operating time requirements. There is a significant difference in the average solving times for both algorithms; clearly the heuristic solving time grows linearly with the increasing appliance number, where the exact algorithm grows exponentially. For 10 controllable appliances, no result was returned by the exact algorithm in 30 minutes computation time. Extrapolating the growth rate from the data obtained, we predict that approximately 12,700 seconds would be required. For 12 controllable appliances, this would rise to over 611 hours. However at this stage, it must be cautioned that the exact algorithm we employed did not utilize pruning techniques (such as branch-and-bound) which may help to reduce the average-case run-time. The extent to which such techniques would impact is an area of future investigation.

### TABLE II. TESTING THE HERURITIC AND EXACT SOLUTIONS FOR DIFFERENT NUMBER OF APPLIANCES

<table>
<thead>
<tr>
<th>No of appliances</th>
<th>Average CPU solving time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heuristic Algorithm</td>
</tr>
<tr>
<td>4</td>
<td>0.000704</td>
</tr>
<tr>
<td>6</td>
<td>0.00123</td>
</tr>
<tr>
<td>8</td>
<td>0.00157</td>
</tr>
<tr>
<td>10</td>
<td>0.00201</td>
</tr>
</tbody>
</table>

Hence, the exact algorithm as presented is not scalable, and should be restricted to the case of a few smart home appliances (e.g. less than 9 appliances). However, the experiment demonstrates that the heuristic algorithm developed in this research is scalable, comparatively very efficient in terms of computational time complexity and could be applied as a core element in a decision support process for a real-time residential appliance scheduling.
V. CONCLUSION

Load scheduling is one major aspect of Demand Side Management. By scheduling loads in response to variable energy prices, consumers can affect economic savings whilst simultaneously reducing pressure on supply side generation at critical times, leading to potentially greener energy production. It has been generally proved in many previous works that load scheduling problems are computationally difficult, and no efficient polynomial time algorithms exist for their solution. This paper proposes exact and heuristic algorithms for load scheduling. Using real system prices for electricity obtained from the Scandinavian and North American energy countries, and representative appliance data, it was found that the proposed heuristic algorithm could achieve within 5% of the optimal cost obtained by the proposed exact algorithm. The results demonstrate that the heuristic algorithm displaced scalable computational behavior, with computational growth linear in the number of controllable appliances. In conclusion, the heuristic approach seems a promising candidate for integration into a rolling horizon optimization approach for smart home appliance management in the presence of variable energy tariffs.

REFERENCES


APPENDIX

A. Smart home appliances – Technical specifications

Table III - VI are the lists of data for the technical specifications of the scheduled smart home appliances: washing machine, tumble dryer, dish washer and electric vehicle. Appliance phases, power assigned to the respective phases, and the length of timeslots are contained in the tables.

TABLE III

TECHNICAL SPECIFICATIONS OF WASHING MACHINE [16]

<table>
<thead>
<tr>
<th>Appliance energy phase</th>
<th>Energy (Wh)</th>
<th>Power (W)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>movement</td>
<td>51.99</td>
<td>119.97</td>
<td>26</td>
</tr>
<tr>
<td>heating</td>
<td>600</td>
<td>600</td>
<td>60</td>
</tr>
<tr>
<td>maintenance</td>
<td>52.4</td>
<td>120</td>
<td>26.2</td>
</tr>
<tr>
<td>Cooling</td>
<td>21.67</td>
<td>130</td>
<td>10</td>
</tr>
<tr>
<td>1st rinse</td>
<td>86.4</td>
<td>480</td>
<td>10.8</td>
</tr>
<tr>
<td>2nd rinse</td>
<td>396</td>
<td>800</td>
<td>29.7</td>
</tr>
</tbody>
</table>

TABLE IV

TECHNICAL SPECIFICATIONS OF TUMBLE DRYER [16]

<table>
<thead>
<tr>
<th>Appliance energy phase</th>
<th>Energy (Wh)</th>
<th>Power (W)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>drying</td>
<td>2400</td>
<td>1200</td>
<td>120</td>
</tr>
</tbody>
</table>

TABLE V

TECHNICAL SPECIFICATIONS OF DISH WASHER [16]

<table>
<thead>
<tr>
<th>Appliance energy phase</th>
<th>Energy (Wh)</th>
<th>Power (W)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-wash</td>
<td>116.72</td>
<td>470</td>
<td>14.9</td>
</tr>
<tr>
<td>Wash</td>
<td>428.16</td>
<td>800.3</td>
<td>32.1</td>
</tr>
<tr>
<td>1st rinse</td>
<td>26.93</td>
<td>160</td>
<td>10.1</td>
</tr>
<tr>
<td>Drain</td>
<td>7.17</td>
<td>100</td>
<td>4.3</td>
</tr>
<tr>
<td>2nd rinse</td>
<td>94.75</td>
<td>310.67</td>
<td>18.3</td>
</tr>
<tr>
<td>Drain &amp; dry</td>
<td>34.87</td>
<td>40</td>
<td>52.3</td>
</tr>
</tbody>
</table>

TABLE VI

TECHNICAL SPECIFICATIONS OF ELECTRIC VEHICLE [17]

<table>
<thead>
<tr>
<th>Appliance energy phase</th>
<th>Energy (Wh)</th>
<th>Power (W)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>charging</td>
<td>2000</td>
<td>1000</td>
<td>120</td>
</tr>
</tbody>
</table>