Appendix C

# Electrical appliance scheduling for a residential neighborhood

A neighborhood of fifteen households with three appliances each is given. Each home is equipped with an EMC which schedules the run time of the appliances. The EMCs can communicate with each other and borrow power at run time. The schematic of the neighborhood problem is shown in Fig. 1.

EMC

EMC

EMC

EMC

EMC

EMC

EMC

EMC

Home

Set of appliances

Fig 1: Schematic of residential neighborhood with 15 homes

## Problem

Schedule the 45 appliances in a 48 hour time period such that

1. The total power consumption is below a certain threshold (Pmax) set by the Electric Company.
2. The electricity cost is optimal.
3. Homes demanding more than the maximum power limit set per home (Pmax/15) compete for extra power amongst each other as long as total power consumed is below Pmax.

## Data provided

bnt: probability that, if an appliance is ‘on’ at period t¸ it will be off in period ‘t+1’

ant: probability that, if an appliance is ‘off’ at period t¸ it will be on in period ‘t+1’

pn: amount of power consumed by appliance n

mn: maximum delay time allowed for appliance n

cn1: cost for each period of delay within mn periods

cn1: cost for each period of delay beyond mn periods

et: cost of electricity in time period t

T: length of the planning horizon (48 hours)

The values of ant, bnt, pn andet for home 1are shown in Figures 2 through 5 respectively.



Fig 2: Appliance Off to On probabilities for Home 1



Fig 3: Appliance On to Off probabilities for Home 1



Fig 4: Power consumption of appliances in 15 homes



Fig 5: Cost of electricity per time period

## Mathematical formulation of the problem

### 3.1 Determine appliance ON request and run time (duration)

The appliance ON request time and duration are calculated using the same technique as the base case (random number generation with value less than ant). It is assumed that the appliance is turned ON only once in the planning horizon of 48 hours. The run time of each appliance is considered as a task to be scheduled on a processor. The processor is the EMC. As all EMCs are identical, the problem reduces to scheduling N tasks on 15 processors. Each task can be represented pictorially as Fig. 6 [].



Fig 6: Graphical representation of appliance run time (task) on a single processor (EMC)

The mathematical terms in Fig. 6 are described below:

rj : Appliance request time. It is the time at which a task becomes ready for execution. This is obtained through the random number generation per time slot.

pj (duration): Appliance run time or duration. This is the time necessary to execute task Tj on the processor without interruption (preemption is not allowed).

dj : The time limit by which the task has to be completed, otherwise the scheduling is assumed

to fail. This is a hard deadline and for the given problem, it is set to infinity because once an ON request is granted, the appliance can run forever.

d˜j : The time limit by which the task should be completed. This is the due date for completion of the tasks and is set to the end of the planning horizon (48 hours).

sj : The task starts time. This is obtained from the scheduler.

### 3.2 Problem definition

Minimize total electricity cost:

*C = max (0, (duration-mn) Cn2)*+ *min(duration, mn)Cn2*

Subject to the total power constraint:

∑*duration \* pn* ≤ *Pmax* (1)

(Sum of all running appliances in the planning horizon)

### 3.3 Scheduling Algorithm

Multiple scheduling algorithms from the TORSCHE [11] scheduling toolbox in MATLAB were analyzed. Two algorithms were found suitable to solve the problem: algorithm for problem 1|rj|Cmax which is similar to first come first serve and earliest starting time first (EST). The description of the algorithms is available in [11]. The usage of the scheduling algorithm is given in section 4 and the flowchart is shown in Fig. 7.

It is observed that there is no significant variation in the electric power consumption cost including the delay penalty. This is unlike the base case where the cost of power consumption is significantly high in certain time periods. As seen in Fig. 5, the maximum power consumption cost variation is USD 0.03. Therefore, the appliance scheduling is optimized to meet the power constraint in (1).

[11] Torsche scheduling toolbox for MATLAB, “http://rtime.felk.cvut.cz/scheduling-toolbox/”

Set task priority as inverse of power consumed by the task in time T

Fig. 7: Appliance scheduling flow chart

No

Yes

The appliance scheduling obtained from the algorithm is shown as a Gantt chart in Fig. 8.

### 

rj = sj

Fig 8: Appliance scheduling on 15 EMCs (parallel processors) using EST algorithm

### 3.4 Determination of Pmax

From the given data set, an optimum value of Pmax is determined such that allthe appliances requesting to be turned ON in the 48 hour time period are granted access. This ensures a quality of service (Qu’s) level of 100%. The appliances were scheduled using the EST algorithm for 1000 randomly generated scenarios of appliance run time and duration. Table 1 provides the total power consumption statistics. Pmax is thus set to the mean value of the power consumed by the 45 appliances (103 KW).

|  |  |  |  |
| --- | --- | --- | --- |
| **Mean (KW)** | **Mode (KW)** | **Median (KW)** | **Max (KW)** |
| 102.97 | 94.9 | 104.7 | 126.9 |

Table 1: Statistical data of total power consumed by appliances running in 48 hour time period

### 3.5 Trade-off analysis between Pmax and Qu’s

To obtain further power savings for the electric company and the neighborhood association, the value of Pmax is lowered below 103 KW. The appliances are now scheduled with a modified version of FCFS algorithm. A priority is assigned to each task which is inversely proportional to the power consumed by the task. This is a greedy heuristic which schedules all the tasks which arrive early and have least processing time. Lowering Pmax below70 KW requires more than one iteration of the algorithm shown in Fig 7 and the Qu’s degrades to 80%. This is shown pictorially in Fig. 9. 15 appliances crossed the due date of 48 hours.



Fig 9: Appliance scheduling by assigning weights to the tasks as per their power consumption.

## Matlab code for the neighborhood appliance scheduling with power constraint

### Declare variables and pre-allocate matrices

num\_house = 15;  
% Number of appliances  
num\_app = 45;  
% Total time slots of length one hour  
time\_slots = 48;  
% Appliance stop time  
stop\_time = zeros(1,num\_app);  
% Appliance run time  
duration = zeros(1,num\_app);  
% Appliance delay cost  
Cd = zeros(1,num\_app);  
% Appliance power consumption cost when ON request granted immediately  
Cpr = zeros(1,num\_app);  
% Appliance scheduling task  
app\_task = zeros(1,num\_app);  
% Power consumed per appliance  
pow\_app = zeros(1,num\_app);  
% Power limit per home (kW) per time slot of one hour  
% Obtained through sensitivity analysis done on neigborhood level max power  
% consumption  
Pmax\_home = 7; %Not used currently  
% Power limit per neigborhood (kW) per day (24 time slots)  
Pmax\_neighbor = 105; % This the average value obtained from the data given

### Data for one neighborhood with 15 households

Static data, per appliance

load m\_n;  
load c\_n1;  
load c\_n2;  
load p\_n;  
  
% Dynamic data per time slot  
% The electricity cost is almost same for all 48 hours,  
% thus schedule the appliance as soon as requested  
load e\_t;  
load a\_nt;  
load b\_nt;  
n = 1;  
j = 1;  
  
  
% split up the given probability data into matrics (time\_slotsX num\_app)  
for k = 1:num\_app  
 for i = n:n+time\_slots-1  
 a\_t(i-n+1,k) = a\_nt(i);  
 b\_t(i-n+1,k) = b\_nt(i);  
  
 end  
 n = n+time\_slots;  
end  
  
% name string for appliances: House#\_app#  
app\_name = cell(45,1);  
n = 1;  
for i = 1:num\_house  
 for j = 1:3  
 house = int2str(i);  
 app = int2str(j);  
 app\_name{n} = strcat('app\_',house,'\_', app);  
 n = n+1;  
 end  
end  
app\_name = app\_name';

### Generate random scenarios for appliance ON time and OFF time

Assumption: Consider only the first instance of appliance turned ON in the 48 hour period

Ton = zeros(time\_slots,num\_app);  
Toff = ones(time\_slots,num\_app);  
  
for j = 1:num\_app  
 for i= 1:time\_slots-1  
 if (Toff(i,j))  
 if(rand(1,1) < a\_t(i,j))  
 Ton(i+1,j) = 1;  
 Toff(i+1:24,j) = 0;  
 break;  
 end  
 end  
 end  
end  
% request\_time = index of first 1 in Ton matrix  
[C, request\_time] = max(Ton);  
% stop\_time = index of first 1 in OFF array  
for j = 1:num\_app  
 for i= request\_time(1,j):time\_slots  
 if(rand(1,1) < b\_t(i,j))  
 stop\_time(1,j) = i+1;  
 break;  
 end  
 end  
end

### Cost formulation

for i = 1:num\_app  
 % Obtain the equipment ON duration  
 duration(i) = stop\_time(i) - request\_time(i);  
 %Power Consumption cost if appliance is not delayed (no scheduling)  
 Cpr(i) = sum(e\_t(request\_time(i):stop\_time(i)));  
end  
% Total cost per house  
for i = 1:num\_house  
 cost\_house\_initial(i) = Cpr(1,i)+Cpr(1,i+15)+Cpr(1,i+30);  
end

### Schedule each appliance to obtain feasible start times

#### Step 1: No power constraint per home in scheduling the appliances

A hard deadline of inf is set for the completion of run-time of each appliances A due date equal to 48 hours is set for all appliances

for j = 1:num\_app  
 app1((j)) = task('',duration(1,j), request\_time(1,j), inf, time\_slots);  
end  
%app\_task1 = taskset(app1());  
app\_task1 = taskset([app1(1,1),app1(1,2),app1(1,3),app1(1,4),app1(1,5),app1(1,6),app1(1,7),app1(1,8),app1(1,9),app1(1,10),app1(1,11),app1(1,12),app1(1,13),app1(1,14),app1(1,15),app1(1,16),app1(1,17),app1(1,18),app1(1,19),app1(1,20),app1(1,21),app1(1,22),app1(1,23),app1(1,24),app1(1,25),app1(1,26),app1(1,27),app1(1,28),app1(1,29),app1(1,30),app1(1,31),app1(1,32),app1(1,33),app1(1,34),app1(1,35),app1(1,36), app1(1,37),app1(1,38),app1(1,39),app1(1,40),app1(1,41),app1(1,42),app1(1,43),app1(1,44),app1(1,45)]);  
% Two scheduling algorithms are tried for optimum scheduling:  
% Earliest Starting Time first (EST)  
% First Come First Serve (FCFS)  
% EST offers the advantage of parallelizing the scheduled tasks on  
% available processors (EMCs). Since all EMCs are identical and can share  
% power with maximum power limit set per neigborhood and not per EMC, EST  
% provides the optimal solution  
prob = problem('P|rj|sumCj');  
% Assign all appliances to 15 EMCs (processors)  
TS = listsch(app\_task1,prob,15,'EST');  
figure();  
plot(TS);  
% schedule the appliances on first come first serve basis  
prob1=problem('1|rj|Cmax');  
TS1=alg1rjcmax(app\_task1,prob1);  
figure();  
plot(TS1,'proc',0);  
% Get the scheduled start times for all appliances  
[start\_time, length, processor, is\_schedule] = get\_schedule(TS);  
% Delay time for each appliance  
delay = start\_time - request\_time;  
  
% Number of appliances that could complete their runtime in 48 hours  
num\_scheduled = 0;  
for i = 1:num\_app  
 if (start\_time(i) + duration(i) < time\_slots)  
 pow\_app(i) = p\_n(i)\*duration(i);  
 % Assign priority to appliances based on their total power consumption  
 % as per original schedule  
 priority(i) = 1/pow\_app(i);  
 num\_scheduled = num\_scheduled+1;  
 else  
 % Appliances which did not start in 48 hour period or did not  
 % complete are assigned least priority  
 priority(i) = 0;  
 end  
end  
% Total power consumed till run time of all appliances  
total\_power = sum(pow\_app);  
% Since obtained priorities are fractions  
priority = ceil(tiedrank(priority));

### Utility Perspective

Schedule appliances such that QoS is not impacted while the power constraint is met Calculate the power consumption for all scheduled appliances in 48 hours

#### Reschedule appliances if total power consumed in original schedule exceeds neighborhood level power limit (Trade off analysis)

if (total\_power > Pmax\_neighbor)

for j = 1:num\_app  
 app2((j)) = task('',duration(1,j), request\_time(1,j), inf, time\_slots, priority(1,j));  
 end  
 app\_task2 = taskset([app2(1,1),app2(1,2),app2(1,3),app2(1,4),app2(1,5),app2(1,6),app2(1,7),app2(1,8),app2(1,9),app2(1,10),app2(1,11),app2(1,12),app2(1,13),app2(1,14),app2(1,15),app2(1,16),app2(1,17),app2(1,18),app2(1,19),app2(1,20),app2(1,21),app2(1,22),app2(1,23),app2(1,24),app2(1,25),app2(1,26),app2(1,27),app2(1,28),app2(1,29),app2(1,30),app2(1,31),app2(1,32),app2(1,33),app2(1,34),app2(1,35),app2(1,36), app2(1,37),app2(1,38),app2(1,39),app2(1,40),app2(1,41),app2(1,42),app2(1,43),app2(1,44),app2(1,45)]);  
% Modified the first come first serve algorithm to consider scheduling along with assigned priorities while scheduling the tasks  
 prob2=problem('1|rj|Cmax');  
 TS2=alg1wjcmax(app\_task2,prob2,15);  
 [start\_time2, length, processor2, is\_schedule] = get\_schedule(TS2);  
 figure();  
 plot(TS2,'proc',0);  
 % Calculate the power consumption for all scheduled appliances in 48 hours  
 % Number of appliances that could complete their runtime in 48 hours  
 num\_scheduled2 = 0;  
 for i = 1:num\_app  
 if (start\_time2(1,i) + duration(i) < time\_slots)  
 pow\_app2(i) = p\_n(i)\*duration(i);  
 num\_scheduled2 = num\_scheduled2+1;  
 end  
 end  
  
% Total power consumed in 48 hour period  
total\_power2 = sum(pow\_app2);

### For home user, calculate the cost of power consumed from the start time.

#### For cost optimization: Rerun scheduling by assigning weights as per cost

Calculate the cost of power consumed for all scheduled appliances Delay time for each appliance

delay = start\_time2 - request\_time;  
for i = 1:num\_app  
 if (start\_time2(i) < time\_slots)  
 if (start\_time2(i)+duration(i) < time\_slots)  
 Cp(i) = sum(e\_t(start\_time2(i):(duration(i)+start\_time2(i))));  
 else  
 Cp(i) = sum(e\_t(start\_time2(i):time\_slots));  
 end  
 Cd(i) = max(0,(delay(i)-m\_n(i)))\*c\_n2(i) + min(delay(i),m\_n(i))\*c\_n1(i);  
 end  
end  
% Total cost per appliance  
C = Cp + Cd;  
% Total cost per house  
for i = 1:num\_house  
 cost\_house(i) = C(1,i)+C(1,i+15)+C(1,i+30);  
end  
  
% Cost saving per appliance  
C\_save = Cpr - C;  
%Cost savings per house  
c\_house\_save = cost\_house\_initial-cost\_house;  
% Power savings  
pow\_sav = total\_power - total\_power2;

end