Documentation for the Peak Demand Management Solution

Shan Dora He
dora.he3@monash.edu

Prof. Mark Wallace,
Dr. Frits de Nijs

September 11, 2021

List of Acronyms

LP linear programming.

PDMP peak demand management problem.

SOC state-of-charge.

List of Symbols

$p_{b,n}^{+}$ the amount of power charged to the battery at time step n.

$p_{b,n}^{-}$ the amount of power discharged from the battery at time step n.

e_b^{init} the initial battery energy level.

e_b^{max} the maximum battery capacity.

$\bar{p}_b$ the maximum battery power rate.

e_b^{min} the minimum battery capacity.

η_b the battery round-trip efficiency.

$soc_{b,n}$ the state of charge at time step n.

c^{anual} the annual peak demand charge.

c^{summer} the summer peak demand charge.

l_n the forecast load at time step n.

f^{cost} the peak demand cost.

h^{eod} the end of day energy level penalty.

g^{health} the battery health cost.

N the total number of time steps.

T total minutes per time step.

n^{eod} the index of the end day time step.

n the index of a time step.

1 Introduction

This document presents the solution for the peak demand management for the Net Zero project at Monash Clayton campus, including the models of the optimisation problem for peak demand management, the solving method and the detailed implementation in Python.

The scope of this work is limited to scheduling batteries given load forecasts and rates for peak demand to minimising the peak demand costs. We assume that the load forecasts are given and updated during the day. A rolling horizon control is used to incorporate changes in load forecasts in real time. The detailed implementation of this work is available on BitBucket `https://bitbucket.org/dorahee2/battery-scheduling/src/master/`. Please email `dora.he3@monash.edu` for access.

2 Battery Scheduling Problem Model

The peak demand management problem (PDMP) is concerned with scheduling batteries to minimise the peak demand charges for the yearly maximum demand and the summer peak demand. This section presents the problem model including the parameters, variables, constraints and objective functions.

2.1 Scheduling Horizon

A scheduling horizon (or a day) is divided into multiple time steps. Each time step has the same length:

- T: total minutes per time step
- $N = 60/T$: the total number of time steps per day
- n: the index of a time step

2.2 Load Data

The load data required for this work is the load forecast l_n for each time step n (in kWh).

2.3 Peak Demand Charge

Two peak demand charges are considered in this work:

- c^{anual}: the annual peak demand charge:
  - on a rolling 12-month basis (11 full months + the current month ignoring the future days in the current month),
  - applied to 30-minute intervals from 7am to 7pm on work days.
- c^{summer}: the summer peak demand charge:
  - for each month from November to March each year,
  - applied to 30-minute intervals from 3pm to 6pm on work days.

2.4 Battery Model

Each battery b is represented by:

- e_b^{init}: the initial energy level at the beginning of the day (in kWh)
- e_b^{min}: the minimum allowed energy capacity (in kWh)
- e_b^{max}: the maximum allowed energy capacity (in kWh)
- $\bar{p}_b$: the maximum power rate (in kW)
- $p_{b,n}^{+}$: the amount of power charged to the battery per time step (in kW):
- $p_{b,n}^{-}$: the amount of power discharged from the battery per time step (in kW)
- η_b: the efficiency (between 0 and 1)
- $soc_{b,n}$: a state-of-charge (SOC) profile — the amount of energy remaining in the battery per time step (in kWh)

The $p_{b,n}^{+}$ and $p_{b,n}^{-}$ are the solutions we seek for the battery scheduling problem.

2.5 Battery Constraint

Each battery b is constrained by the followings:

- at each time step n, a battery cannot charge or discharge at a rate higher than the maximum power rate:

$$\forall n \in [1, N],\ 0 \leq p_{b,n}^{+} \leq \bar{p}_b \tag{1}$$

$$\forall n \in [1, N],\ 0 \leq -p^{-}_{b,n} \leq \bar{p}_b \tag{2}$$

- at each time step n, a battery cannot have more (or less) than the maximum (or the minimum) allowed energy:

$$\forall n \in [1, N],\ e^{min}_b \leq soc_{b,n} \leq e^{max}_b \tag{3}$$

- at the first time step of the scheduling horizon, the battery must have satisfy the given initial energy level:

$$soc_{b,1} = e^{init}_b \tag{4}$$

- at each time step n, the SOC depends on the SOC, charge and discharge at time step $n - 1$:

$$\forall n \in [2, N],\ (soc_{b,n} - soc_{b,n-1}) \times (60/15) = p^{+}_{b,n-1} + p^{-}_{b,n-1} \tag{5}$$

2.6 Objective Function

The objective is to minimise the total peak demand charges each (financial year) while keeping the battery operates in an healthy manner. Two objectives are considered:

- peak demand cost:

$$l'_n = l_n \times (60/T) + \sum_b (p^{+}_{b,n}\ \eta_b + p^{-}_{b,n} \times \eta_b) \tag{6}$$

$$f^{cost} = max([l'_n\ | n \in [1, N]]) \times (\alpha \times c^{anual} + \beta \times c^{summer}) \tag{7}$$

$$\alpha = \begin{cases} 1, & \text{if the current time is between 7am to 7pm} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\beta = \begin{cases} 1, & \text{if the current month is Nov/Dec/Jan/Feb/March,} \\ & \text{and the current time is between 3pm to 6pm} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

- battery health cost that prevents batteries from charging or discharging more frequently than needed:

$$g^{health} = \sum_{b} \sum_{n=1}^{N} p^{+}_{b,n} \tag{10}$$

- end of day energy level penalty that matches the energy level at the end of the day to the maximum capacity as much as possible:

$$h^{eod} = \sum_{b} (e^{max}_{b} - soc_{b,n^{eod}}) \tag{11}$$

Note that we have added the battery health cost to avoid charging and discharging the battery too frequently.

2.7 Formal Problem Formulation

This problem seeks the best values for the charge/discharge per time step: $p^{+}_{b,n}$ and $p^{-}_{b,n}$ that solves the following problem:

$$\begin{aligned} &\textsf{minimise} && f^{cost} + g^{health} + h^{eod} \\ &\textsf{subject to} && (1),\ (2),\ (3),\ (4),\ (5) \end{aligned}$$

3 Method

The main solution for solving the peak demand management problem (PDMP) include a linear programming (LP) model and the rolling horizon control. The rolling horizon control is illustrated by Figure 1 — the load forecast is refreshed regularly during the day, for example, every 15 minutes or half an hour. Whenever the load forecast is refreshed, the battery scheduling problem is solved again using the updated forecast data, and the charge and discharge decisions from the new schedule will be implemented in practise until the load forecast is refreshed again.

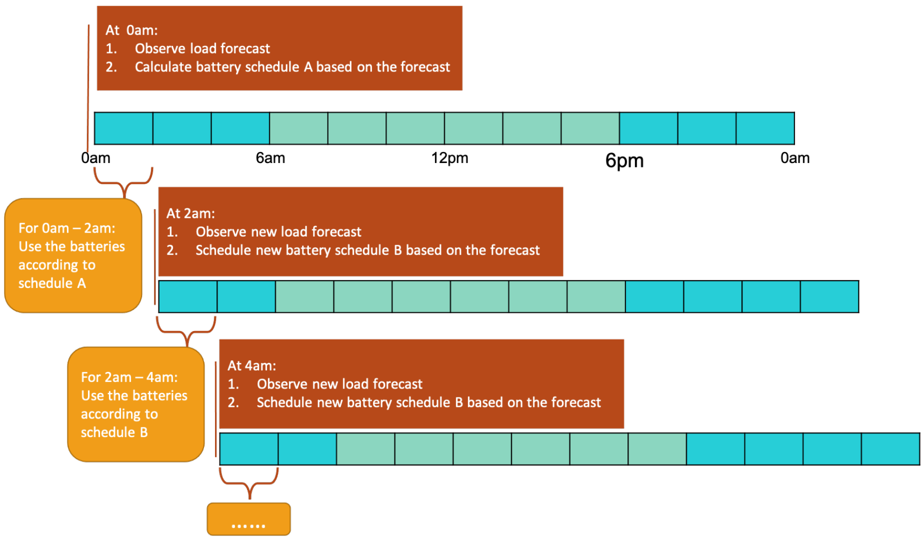

Figure 1: Rolling Horizon

Let us write $\bar{d}^{ann}$ as the demand threshold for the annual peak demand charge and $\bar{d}^{sum}$ as the demand threshold for each month from November to March. The rolling horizon process can be described as follows:

1. set the demand thresholds for both charges $\bar{d}^{ann} = zero$ and $\bar{d}^{sum} = 0$.
2. whenever the load forecast is refreshed:
    (a) check if the forecast maximum demand exceeds the current minimum demand threshold:
    (b) if yes:
        i. solve the battery scheduling problem,
        ii. implement the charge and discharge decisions in practise until the load forecast is updated again,
        iii. update the demand thresholds based on the actual demand and the implemented battery schedule.
    (c) if no, do nothing and wait until the load forecast is updated again.
3. repeat Step 2.

4 Detailed Implementation

The algorithm is implemented in Python, and the LP model was written in MiniZinc and solved by a mixed-integer programming solver, such as COIN-BC or Gurobi. The code is available on BitBucket `https://bitbucket.org/dorahee2/battery-scheduling/src/master/`.