The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Reducing the cost using Battery Energy Storage System with Fluctuating Electricity Price based on Optimization

Submitted April 2023, in partial fulfillment of the conditions for the award of the degree BSc

20215708
School of Computer Science
The University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature: Chengyang Zhong

Date: 2023/04/20

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive

Abstract

Contents

1 Introduction

The wholesale electricity price has been rapidly increasing, for example, the electricity price in the first half of 2022 was twice the average from 2016 to 2021 during the first half of the year in Europe [1] multiple factors could lead to this. First, since Covid lockdown and trade restrictions, the demand for fossil fuel energy surpassed the supply and further caused an increase in oil prices. Moreover, the Russia-Ukraine war worsens the situation due to the sanctions imposed on Russia. This sanction resulted in a disruption in the energy supply to European countries and make the gas price rise. Both of them contributed to the energy crisis in Europe, consequently, leading to electricity prices increasing [18].

Additionally, people became more and more aware of the problem of air pollution and global warming caused by traditional energy sources. Indoor and outdoor air pollution is responsible for more than two million death cases and illnesses cases hard to estimate, which may have a terrible impact on the economy and security of nations. Furthermore, global warming both creates disease problems like heat stress and exacerbates the severity of environmental problems like tropical storms. Worse still, the location of the suitable agricultural place will be changed, and the ecosystem will be destructed as well as animal habitats [24]. Hence, to avoid the appalling consequences caused by the usage of conventional energy sources, people tried to replace them with clean energy. However, clean energy like wind and solar power is intermittent, which means its power output instability makes it hard to match electricity demand [22]. Hence, Battery Energy Storage System(BESS) can be used to overcome this shortage by collecting energy during low-demand periods and injecting the energy when needed [19].

BESS is composed of batteries, a control and power conditioning system, and the rest of the plant. The rest of the plant is aimed at protecting batteries and control and power conditioning systems. The batteries are made of stacked cells. The most important features of the battery are its energy capacity and power rate. Furthermore, the batteries used in BESS also have different types with different materials and working modes. Another component Controls and Power conditioning systems(C-PCS) are critical in BESS, which is also the field that this paper will focus on. C-PCS interfaces the batteries to the loads and controls the behavior of the batteries-charge or discharge and the charging rate. This component can be used to enhance power system reliability and power quality [10].

The price of the battery and the electricity price tariff structure also make BESS more advantageous. The lithium-ion battery price has been reduced by 97% since 1991, which enables the cost of renewable technologies to be cheaper than fossil fuels [2]. Besides the battery price, the tariff structure is another reason to deploy BESS. Many countries including the UK [23] adopt a time-of-use(TOU) tariff policy due to the shortage of energy. TOU is a kind of structure that stipulate peak hours with higher electricity price and off-peak hours with cheap price during the day. This gives BESS room to maneuver, which means BESS can charge at periods when the price is cheap and discharge at periods with higher prices. As a consequence, the entire load is still met with lower expenses [12].

The BESS prediction control problem becomes a significant problem in many aspects. As illustrated above, because of the rapidly increasing energy price, millions of people have been impacted by the price, especially those with lower-income households. For instance, it will cost the poorest 20% of families an extra £1,000-1,100 [6]. Apart from reducing the electricity cost, BESS can also be applied in many other different applications. They can act as an uninterruptible power supply and reduce the negative impact of downtime during an electricity grid failure which is vital to equipment like servers that require a high level of continuous power supply. BESS can also be used to build independent microgrids in remote areas which can help people live in such places and avoid heavy costs [9].

Furthermore, current power systems face technical limitations, such as unpredictable renewable energy sources and high load demands, which create economic inefficiencies in the deregulated electric market [8]. As a result, energy storage technology presents promising opportunities for various stakeholders [3]. To address these challenges, the battery energy storage system (BESS) has been developed and implemented. This system is composed of batteries, control and power conditioning systems, and protective devices [11]. The BESS is connected to the AC distribution grid through a converter and can serve as an end-user load to reduce energy costs during periods of low prices and enable load shifting during peak demand periods, thereby improving grid stability [21]. Using this system, low-income households can also reduce their energy burden and contribute to overall power system improvement by taking advantage of predicted electricity prices and managing their loads more efficiently.

2   Motivation

Due to the practical datasets and the well-structure of the BESS prediction control problem, correct answers can be generated through the application of an appropriate algorithm. Hence, comparing different approaches and strategies and applying novel algorithms become easier work.

Currently, there are several existing approaches and strategies, such as Mixed Integer Linear Programming, in the previous papers that can be applied to optimize the BESS prediction control problem which means providing sequential control operations to manipulate BESS for the purpose of exploring the global optima. However, some of them only focus on the performance of one exact approach, they have emphasized the work mode of the algorithm and how close their models come to reality. Consequently, people are confused when there are several different approaches to the same problem with no idea which one performs better, although these algorithms have been studied in detail and comprehensively. Other existing algorithms have been studied and compared in other papers under the same circumstance of the problem. Nevertheless, these papers either didn't compare them in-depth, or the methods they have put together are few. That's one of the motivations for this paper, to do some research on the deep comparison between those existing algorithms.

Additionally, some approaches that have not been applied to this problem but are famous in other fields will also be implemented and compared in this paper. Some of these algorithms such as rule-based heuristics have proven their efficiency and effect on other problems which are similar in structure to the problems this paper is going to study. Other algorithms like rolling horizon algorithms is suitable for time series problem including the BESS control model. Applying these novel approaches to the BESS prediction control model is another motivation for this paper.

The final motivation for studying this topic is the practical and economic factors mentioned in the introduction. After finding the most feasible algorithm that combines all aspects, BESS can be better deployed in commercial or civil scenarios. Hence, they can help reduce the electricity cost for those who will use BESS in a shorter time and with less memory. With a completed, professional model, customers no longer need to operate or set the parameters and instructions themselves, the model will help them to deal with the complicated data of loads and various electricity tariff structures, which may make BESS better promoted. Moreover, the grid can ease the burden of imbalanced loads between peak and off-peak times and the waste of energy when the demand is far away from the generated energy during off-peak times. All of these improvements could further reduce the nation's financial expenditure on electricity problems. In the future case, BESS can even combine with clean energy and avoid their disadvantages to improve the situation of energy crisis and global warming.

3 Related Work

Comparing different methods and applying some novelty algorithms that have not been applied to the BESS prediction problem are two of the motivations in this paper. Many algorithms have attempted to solve the BESS prediction control problem or some problems which are related or similar to it. Exact algorithms can always find the global optima of the problem instance, while heuristics may not guarantee the optimality but is more efficient.

3.1 Exact Algorithm

3.1.1 Mixed Integer Linear Programming

Eskil Sulen Gjerde [14] has implemented Mixed Integer Linear Programming (MILP) and Dynamic Programming (DP) for the BESS prediction control problem. The paper compared them with some traditional methods on different measures-Running time and battery utilization. After collecting the results, the paper concluded that MILP ran an average of 5 times faster than DP and was easier to understand than Dynamic Programming. However, the paper only focused on two approaches without considering other algorithms and didn't use more metrics to measure different approaches. Hence, an in-depth comparison would make the conclusion more convincing in this paper.

3.1.2 Dynamic Programming

Muhammad Farooq [13] discussed Dynamic Programming and DP with some pruning strategies. The time constraint was utilized to trim down the Dynamic Programming approach that was based on the Economy-7 tariff policy. This policy assumes that the tariff price would be kept low during some period in a day, hence, it is recommended that BESS charge and store energy during this period. In addition, there is another restriction called the Tariff Threshold Constraint that dictates that the BESS should discharge energy to power-consuming devices when the loads reach their peak limit, and import energy from the grid when the loads are at a low level. Furthermore, the BESS behavior is restricted by the load trend, which prohibits charging when the load is increasing and discharging when the load is decreasing. Another study conducted by Guillem Rigaill [20] concluded that pruning strategies would enhance the performance of the algorithm significantly by applying these strategies to DP on the K-change-points problem. However, in practice, the load trend constraint does not play a role in improving the quality of the solution. Therefore, the load trend constraint is dropped in pruned Dynamic Programming. However, Muhammad Farooq only focused on Dynamic Programming, though DP and pruning strategies have been studied in detail. In this paper, several approaches will be discussed and compared.

3.2 Heuristic

3.2.1 Rolling Horizon Approach

A heuristic that is suitable for time series problems like the BESS prediction control model has been studied by Glomb L.et al [15]. The idea of the paper is to divide a long period into several small periods, and then solve the sub-problem within each range of the small periods-the result is called the initial solution and becomes the initial local optima. Then, the windows based on these ranges will move one step each time, and solve a new sub-problem within the window. The new solution with more information about the future will be compared with the local optima, the new solution will be accepted if it is not too worse than the local optima. Aftering moving the windows to the end, all the solutions of the sub-problems have been generated and they will form a completed solution for the entire problem. With the knowledge of the rolling idea, MILP and DP will be used to solve the sub-problem of the BESS prediction control problem.

3.2.2 Rule Based Heuristic

Another heuristic called the rule-based heuristic has been designed for some related problems by Michael Alexander Campbell [7]. At each time interval, the model will follow a series of rules and make a predefined operation. Furthermore, he created rules based on peak period and photovoltaic(PV) and

customized high and low thresholds. However, since this paper doesn't consider the PV component, the rules, and their order has been changed, including the way to calculate two thresholds. Moreover, some prediction models of the loads will be applied to the rule-based heuristic which enables the heuristic to gather more information and make a more reasonable solution.

Farias O. et.al [17] has proposed another different series of rules. Though that research concentrated on scheduling and real-time operation in electric vehicle charging stations. One kind of set of rules that have been discussed in their work is that according to the different amounts of remaining energy in the battery, the model will distribute different probabilities to different charge modes (with different charging rates). Hence, The idea will be covered in this paper while some finer control on the charging rate will be designed.

3.3 Evaluation Method

One of the motivations of this paper is to compare different methods in-depth, thus, the way to compare them is important. The article [5] has introduced 9 criteria for three performance category-efficiency, reliability, and quality of the solution. In this paper, only part of these most suitable criteria will be used to measure various methods.

3.4 Extended

Abronzini et.al [4] collected the data from battery manufacturers to predict the achievable life cycles. After getting data on the depth of discharge at that time interval, with the predicted life cycles, the degradation cost can be calculated. Lee Jason [16] has verified that if the battery temperature is constant, then the rate of degradation is also constant. Moreover, he further provided the equation to calculate the degradation rate. Additionally, Zhou et.al [25] has figured out the relationship between the life cycle and depth of discharge for Lithium-Ion batteries. Hence, based on the depth of discharge, the life cycle is calculable. With the attempt to extend the model, the degradation component has been embedded in the rule-based heuristic in this paper.

4 Description of the work

4.1 Problem

The BESS prediction model problem is that given an electricity tariff structure and the electricity loads data, building a model that minimizes the total cost of electricity. The tariff structure varies from country to country, from time to time. However, in this project, all these values such as the peak and the electricity prices within the different periods are fixed.
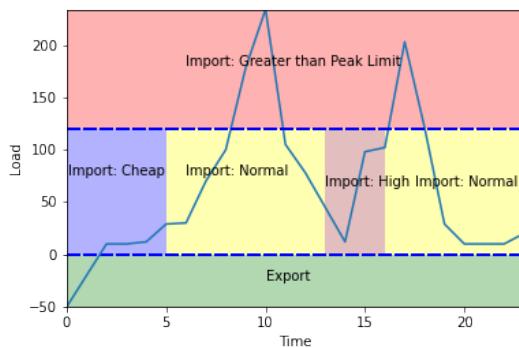

Figure 1: An example of tariff structure in a day [14]

Figure 1 above shows a typical Time of Use tariff structure that contains most of the possible situations existing in a tariff. If the entire loads exceed the preset peak limit, then the price will rise for the

excess. Otherwise, when the electrical load is maintained at normal levels, most of the time the price is also set at normal levels. Nonetheless, since the government or power companies attempt to prevent the excessive load from causing damage to the grid during peak times, the energy price will be set at a high level to encourage people to use electricity at other times. On the contrary, the demand decreases rapidly at some times like midnight, therefore, the price will be adjusted low to make full use of the power grid. Additionally, the structure in the figure also has a price for the export, which stands for the price when individual or business customers have some measures to generate energy or they want to sell the energy which is bought at a cheap price to make a profit. This project will not consider the export part, the introduction is only for completeness and objective.

When discussing the operations, continuous time is divided into time intervals of equal length because it's infeasible for most of the time series algorithms. Hence, at a certain time interval, the model will do only one action. The model will try to find a sequence of operations, which can meet the needs of users and reduce the cost to the minimum. The ideal result is shown below.
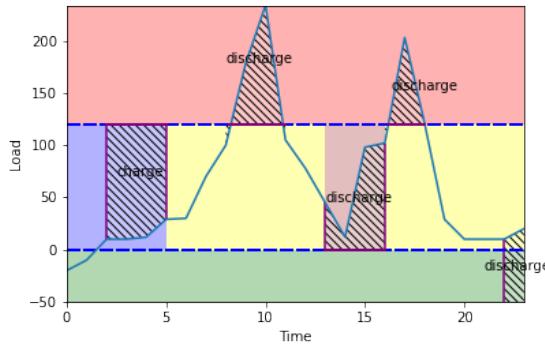

Figure 2: The ideal result after applying the BESS model

Under ideal circumstances the amount of charging controlled by the model when the price is cheap, can offset the loads exceeding peak limit or in peak period. Nevertheless, this ideal situation will not happen due to many constraints in real life. For instance, the efficiency of charge and discharge and the capacity of the battery will reduce as time goes by. In addition, the tariff structure is not always fixed in real life, which means the model has to receive the current price structure in real-time. The future loads' data is unknown, which increases the prediction difficulty of the model. This project only focuses on the basic problem, hence, the optimization approaches will try to find the best solution under some of these constraints.

4.2 Model

Each time interval has a load $L_i \in \mathbb{R}$ which stands for the consumption at that time. The battery in BESS has a maximum capacity of C_{max}, maximum input and output power of Po_{max}, the amount of charge or discharge of p_i at each time interval where $-Po_{max} \leq p_i \leq Po_{max}$. In the electricity tariff structure, P_{ij} stands for the price at certain time intervals and load levels. For $j = 0$ and $j = 1$, they stand for the situation when loads are at normal level and peak level PL. The price for time i is calculated by

$$P_i = \begin{cases} (L_i + p_i - PL) * P_{i1} + PL * P_{i0}, & \text{if } L_i + p_i > PL \\ (L_i + p_i) * P_{i0}, & \text{if } 0 \leq L_i + p_i \leq PL \end{cases} \tag{1}$$

Hence, the cost function is

$$P = \sum_{i=0}^{T-1}, \text{T is time interval} \tag{2}$$

7

4.3 Objective and Aim

The project aims to compare different methods in depth-success rate, percentage of global optimal found, running time, and memory usage. Such comparisons can rarely be found in other papers. Except for comparisons, this paper similarly focuses on novelty. some methods that have not been applied to the BESS prediction control model are also implemented in the project, to find out if there is a better method.

5 Methodology

5.1 MILP

MILP is a mathematical optimization algorithm with some linear constraints and one linear objective function. Apart from linear programming, MILP also allows the mixing of discrete decision variables which means limiting variables to integers and continuous decision variables together. The model built by MILP applies a combination of some linear programming techniques and branch-and-bound methods, which enables the model to search the solution space to find the optima. The branch-and-bound approach will break down a problem into smaller sub-problems and systematically explore the solution space of each sub-problem. Additionally, linear programming techniques are employed to establish bounds on the objective function. At first, MILP will apply linear programming to find the optimal solution. Nonetheless, if one or more integer decision variables that are necessary for the optimal solution take non-integer values, choose one of the variables, and separate the problem into two subproblems according to the integer variable. Each of the subproblems will generate a new constraint based on the integer closest to that variable. The figure below shows how the Branch and Bound method works in the MILP.
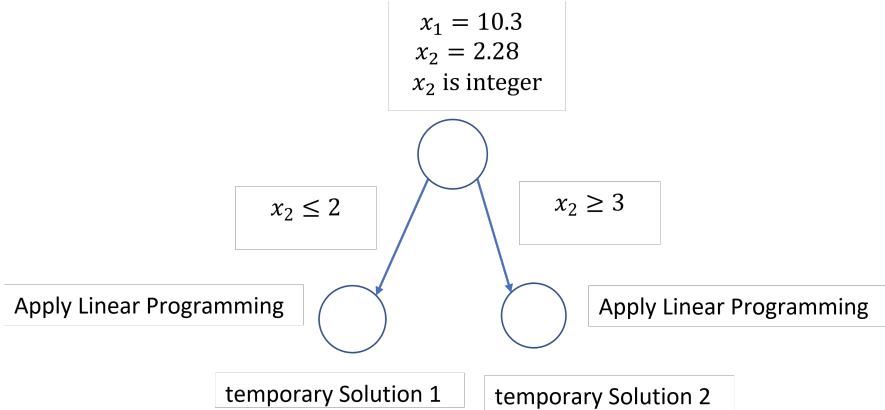

Figure 3: An example of Brand and Bound method

After dividing the problem into two subproblems, MILP will again use linear programming to solve these subproblems. If there are still integer variables whose values are not integers, repeat the division behavior of the branch and bound until all of the integer decision variables that reach the optimal solution have integer values. After traversing the entire solution space, the optimal solution of these subproblems is selected as the optimal solution of the whole problem. Due to the properties of the branch and bound method, many subsets of solution space that don't need to be evaluated can be discarded since the integer decision variables don't satisfy the integer property. Hence, by a combination of linear programming and the branch and bound method, the MILP's computational efficiency is greatly improved and the optimality of the solution is still guaranteed.

MILP is an exact approach which means it can find optimal solutions to complex optimization problems with discrete decisions. Moreover, according to Gjerde Eskil Sulen [14], the MILP runs faster than some algorithms like Dynamic Programming. Though understanding how MILP exactly work is difficult, it is easy to implement it with the help of some IDE or framework. The MILP will generate many constraints according to the limitations of various conditions in reality. For instance, the energy

inside the battery is always greater than 0 and less than the capacity of the battery for the different states of the battery located at all the time intervals. Additionally, Due to the tariff structure, the energy needed to be imported from the grid is segmented into several sections and each of which has a different price. Hence, the MILP has to generate additional constraints to ensure the equality between import energy and the sum of different energy sections.

5.2 Dynamic Programming

Dynamic Programming is a traditional algorithm to solve optimization problems. It can break the original problem into several sub-problems because the solution of the sub-problem can contribute to the final solution of the original problem. Additionally, the solution to the sub-problem will be stored since they will be reused in some way. Dynamic Programming has two different types of implementation-Top down and Bottom up. For Top down Dynamic programming, the problem is firstly broken into several sub-problems and then compute recursively with the result stored in a table. For Bottom up Dynamic programming, the sub-problem is solved iteratively and finally composes the final solution of the large problem. The bottom up method is adopted in this project because it is more suitable for the BESS prediction control problem with a sequential structure.

The most significant reason for applying Dynamic Programming is the feature that Dynamic Programming can help find optimal solutions. Hence, by combing the result of MILP and Dynamic Programming, verifying whether a global solution has been found is simple. Furthermore, dynamic programming is easy to understand and implement. All of the computational steps are iterative or recursive which can learn the structure of the problem by learning from the simple sub-problem. At each time interval, different states have been divided according to the amount of energy inside the battery to enable the searching in solution space. Hence, the sub-problem is defined as the minimum cost calculated from the start time interval with no energy in the battery to the current time interval with fixed battery energy. After solving all the subproblems, the answer to the original problem is solved because based on the solution of minimum cost from the start time interval to the previous time interval, it is easy to get a solution from the previous time interval to the current time interval.
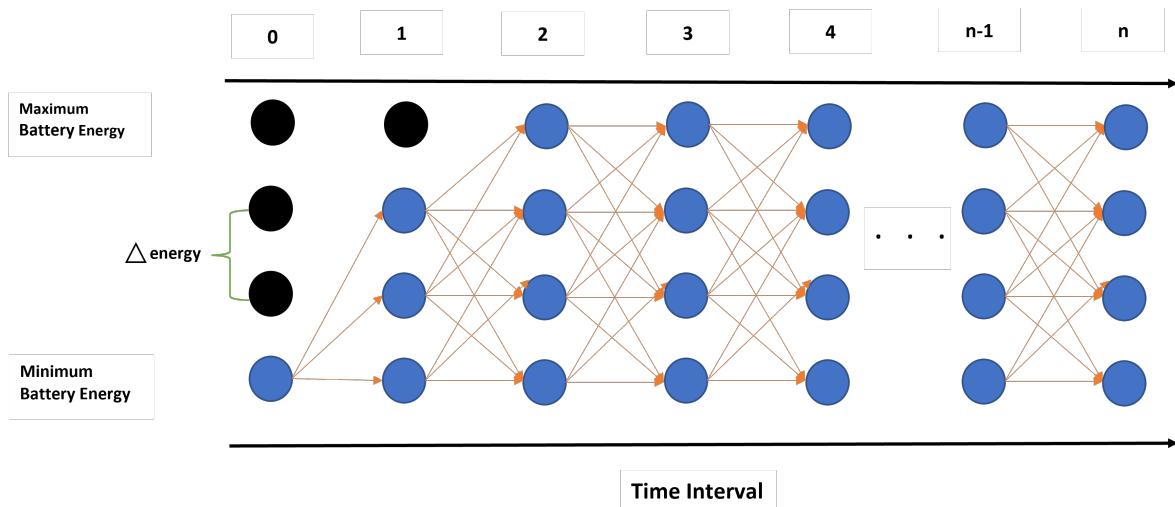
Figure 4: An example of a model built by Dynamic Programming

Figure 4 shows the structure of the dynamic programming and the nodes are the states of the battery (remaining energy in the battery) in different time intervals. The blue nodes stand for reachable states and the black nodes stand for unreachable nodes. The reachable node is defined as when starting from the beginning node, after making some operations of charge and discharge, the node which can

be reached. Moving from one state at a time interval to the state at the next interval represents the charge and discharge behavior-moving to a higher state means importing energy from the grid and charging the battery, inversely, moving to a lower state is discharge from the battery and using the energy for the loads. The difference between nodes and their neighborhood is shown as $\Delta energy$, which is the minimum change of energy of different states between nearby time intervals. On the contrary, the nodes' maximum change of energy is constrained by the maximum input and output power of energy, which is set to equal in this project. Not only the node (or state) contains information of the state of charge of the battery but also contains the energy cost from the beginning state to the current state.
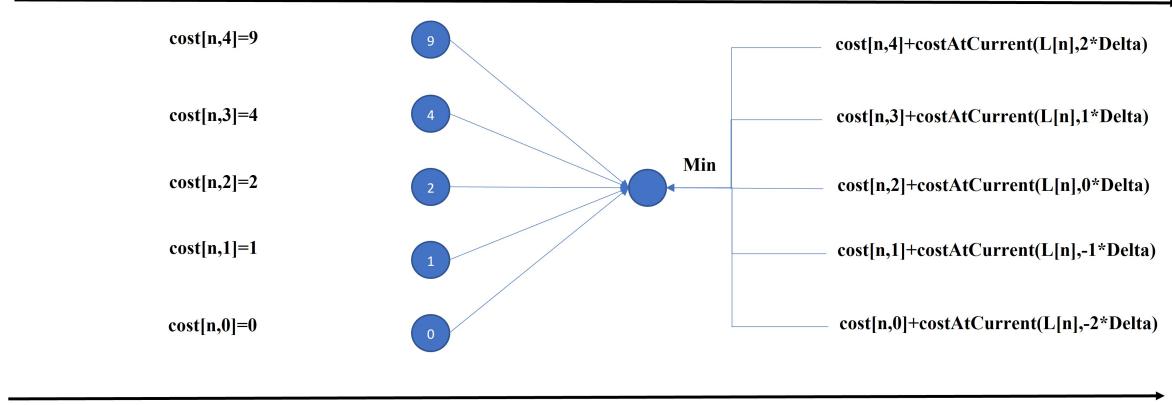

Figure 5: How DP works to solve the BESS prediction control problem

Figure 5 shows the reason why Dynamic Programming can find the optimal of the sub-problem. Due to the varying electricity price in the tariff structure, the same energy may have different prices. This feature is the key idea of dynamic programming. As shown in the figure above, there are different paths to reach that node by the act of charging and discharging a battery. After calculating the total cost given the current electricity price, battery charging and discharging status, and load information, the cost at the current time interval and current state can be calculated. Hence, by comparing different costs and choosing the minimum one, the sub-problem at its current status can be solved.

5.2.1 Pruning Strategies

According to the work of Muhammad Farooq [13], several pruning strategies have been applied to improve the efficiency of dynamic programming.
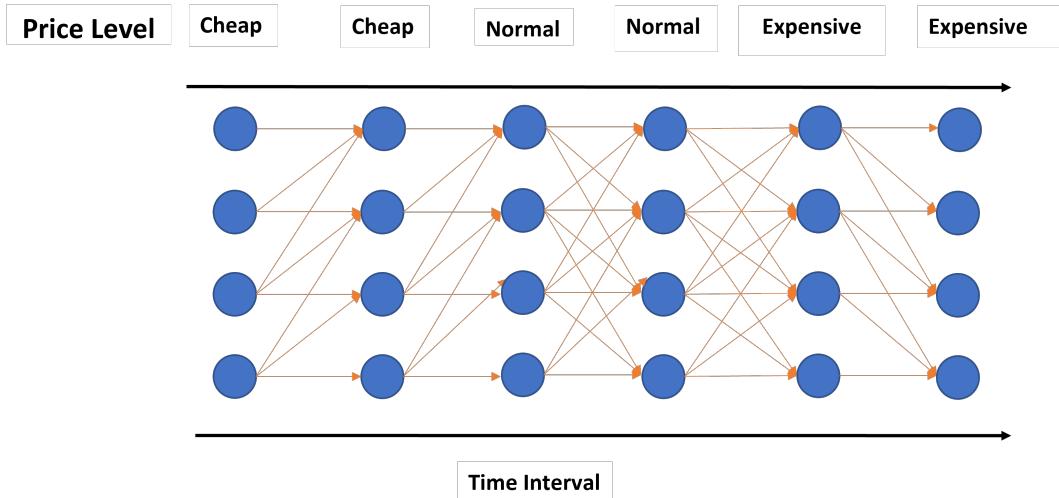

Figure 6: Time Constraint

Figure 6 shows how the time constraint pruning strategy applies to dynamic programming. When the price is at a normal level, no pruning is applied. However, since the government or electric power companies will adjust the price to perform peak shaving, the price will become very cheap or expensive for a certain period of time. Hence, to buy energy at a cheap price to avoid electricity consumption at a high price, under no circumstance should dynamic programming perform discharge when the price is cheap, likewise when the price is high. Another constraint called tariff threshold constraint works in a way similar to time constraint. When the load exceeds the peak limit, charging behavior is disabled because the price is high under such circumstances. The behavior is exactly the opposite when the load doesn't exceed.

5.3 Rolling Horizon Approach

Due to the fact that the BESS prediction control problem requires making complex decisions over a large time-span search space and the data in the problem repeats in a similar identical way, the rolling horizon approach can take advantage of such properties. The method will focus on solving the subsequence of the problem. An example of the structure of the rolling horizon can be found in Figure 7.
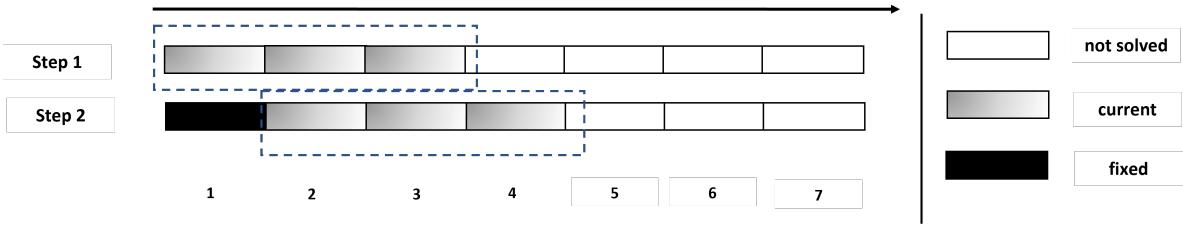

Figure 7: Rolling Horizon Approach

The approach will create a fixed-size window within a sub-problem. Then another problem-specific algorithm will be applied to this sub-problem and get an optimal solution with a sequence of operations for this sub-problem. As a result, the operations and the cost at each time interval are stored. After that, move the window forward by 1 time interval, solve the new sub-problem, and overwrite the operations and cost for each time interval within the window. The distance the window moved can be set, nevertheless, to improve the optimality of the final solution, it is moved only one time interval at each step. By iteratively moving the window and solving the problem, all the operations will be found and fixed. As a consequence, the complete sequence of operations is then solved.

Rolling horizon has been applied to large quantities of different problems and for some of those problems, the rolling approach's efficiency has been proved [15]. The efficient calculation is vital in this BESS prediction control problem due to its large time span. Moreover, the rolling horizon approach can always find a solution that is almost identical to the global optimal solution. Another key factor for applying the rolling horizon approach is that the algorithm doesn't need global knowledge which makes the model more realistic. In reality, it's impractical to know the future information about the tariff structure and the loads in-home or in companies, thus only partial information is known which is suitable for the rolling approach. Last but not least, unlike MILP and dynamic programming, the rolling horizon approach is designed for such time series problems like the BESS prediction control problem and it has not been applied to this problem in previous work, hence, this will add novelty to the project.

However, the rolling approach is something like the hyper-heuristic method, the problem-specific algorithm is required to solve the subproblem generated by the rolling approach. Hence, that's where MILP and dynamic programming come in. Within each window, apply MILP or dynamic programming to find the sequences of operations that minimize the cost of the sub-problem.

5.3.1 Improved Rolling Horizon Approach

However, though the rolling horizon method guarantees sub-optimal in the current period, this information will never be used in future periods when making optimal decisions. As a result, the simple rolling method is not able to satisfy any quality guarantee. Hence, an improved rolling horizon approach is designed by L. Glomb et.al[15]. On the basis of the rolling horizon approach, it generates a new constraint to guarantee the quality of the solution. The improved approach divides the entire problem into several fixed sub-problem according to the window size. Then when the window in the simple rolling horizon approach moves to the location where these subproblems start, at first use the simple rolling horizon to get the optimal solution of the fixed subproblem and store it. After that, as long as the window intersects with the subproblem's time range, the influence of new operations on the optimality of the solution of the fixed subproblem will be considered. If the solution based on more future information is only slightly worse than optimal, accept and overwrite the existing operations, otherwise discard the new sequence of operations.
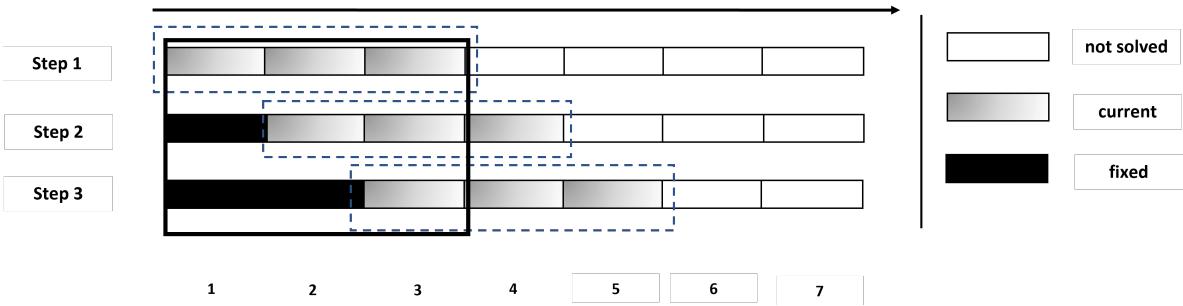

Figure 8: Improved Rolling Horizon Approach

Figure 8 shows how the improved rolling horizon approach works. The black frame stands for the fixed sub-problem whose solution is known and all the windows at each step above intersect with the black frame (fixed sub-problem). Hence, when getting a sequence of operations by applying a problem-specific algorithm, these operations will generate a result that contains another different solution to the original fixed sub-problem. Since the solution in the current window with more future information is more likely to improve the quality of the final solution, hence a slightly worse solution would be acceptable.

5.4 Rule-based Heuristic

Rule-based heuristics always take the current load and price as input and then follow a series of rules. After the traversal of the entire rule chain, the only result operation will be picked and performed.

Compared to other algorithms, it is simple to implement the rule-based heuristic for the BESS prediction control problem since only several rules need to be set. Moreover, unlike all the previous approaches, the whole approach has no complex structure and doesn't need an additional data structure to store other information. Hence, the amount of memory required is very small. For the same reason, running a rule-based heuristic takes only a short period of time to get the result. Additionally, since the rule-based heuristic doesn't require any future information and only depends on the information at the current time interval, it is more consistent with the lack of information in the real world. Hence, with the fast running speed and current information, the approach can react quickly to the change in price or load which makes the rule-based heuristic a real-time method. Meanwhile, on account of such property, the optimality of the solution will not be guaranteed with no future information. Consequently, as opposed to MILP being able to find the global optimal, rule-based heuristic puts a lower bound on the minimum cost [7].
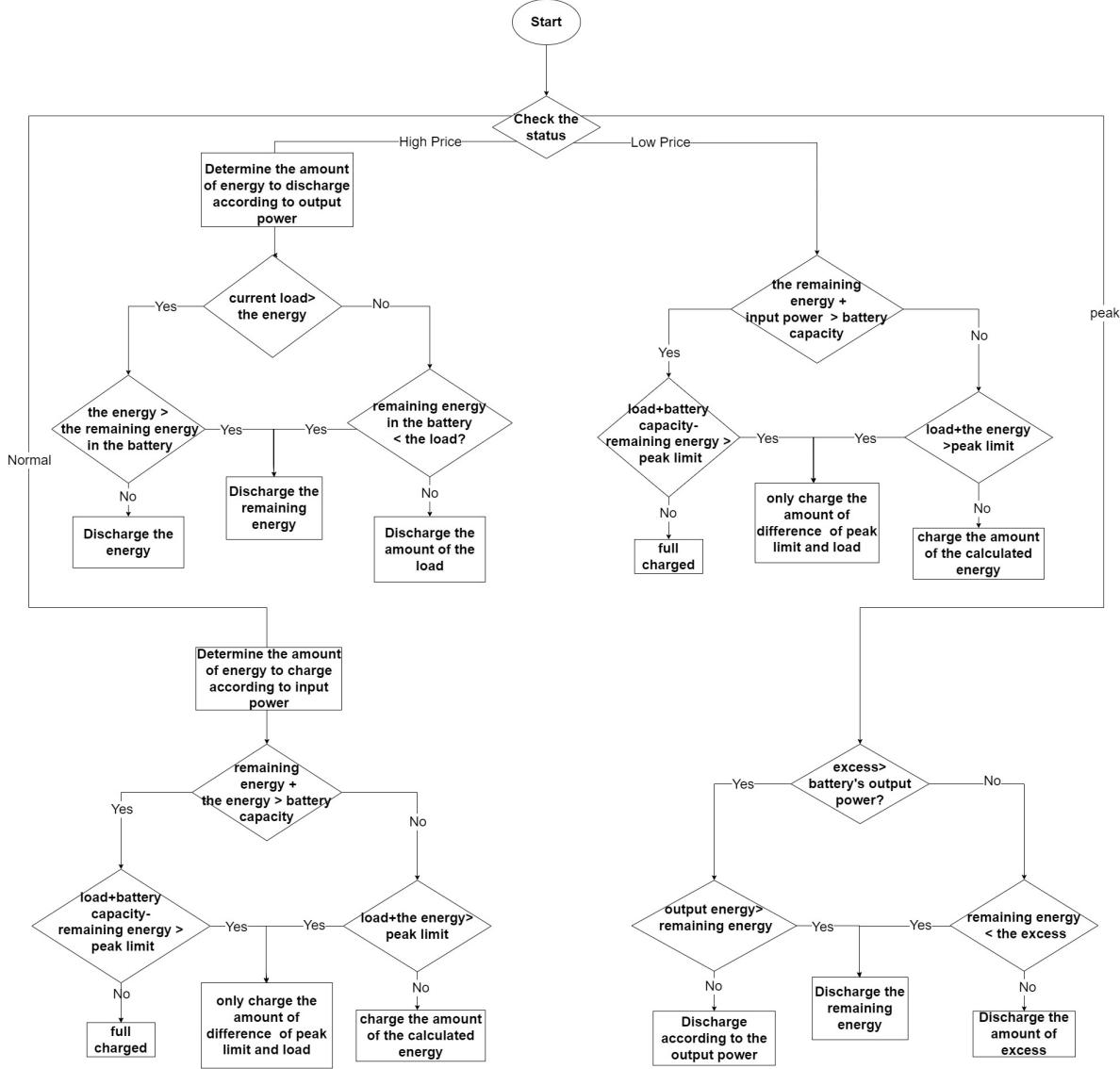
Figure 9: Simple rule-based heuristic

As shown in Figure 9, there are four main rules for different situations: the load exceeds the peak limit, the price is high, normal, or cheap. When the load surpasses the peak limit or the price is at a high level, only discharge behavior is allowed. However, when the price is at a high level, the amount of discharge depends on the ratio of remaining energy in the battery to the battery capacity. Inversely, when the price is at a normal level or cheap level, the algorithm will perform a discharge operation. Since it is more profitable to charge the battery at a cheap level, the approach will only charge limitedly at a normal price and charge as much as the battery can when the price is cheap. The amount of energy to charge is calculated based on the ratio of the current load to the peak limit. If the load is closer to the peak limit, the loads in the future are more likely to be greater than the peak limit. Hence, the battery should charge as much possible as and that's the reason to use the ratio. Additionally, the remaining energy in the battery and the maximum output power of the battery must be considered when discharging, because under no circumstance should a battery discharge efficiency be greater than the maximum output power or discharge the amount of the energy be greater than the remaining energy. Similarly, when charging the battery, the amount of energy at the current time should not be greater than the maximum input power and capacity of the battery. Furthermore, if the sum of the current load and the energy needed to be imported into the battery exceeds the peak limit, then the excess portion will be charged a high fee which will increase unnecessary expenses. Hence, this secondary rule should be added to the discharging rule chain to improve the quality of the

13

solution.

5.4.1 Rule-based Heuristic with Prediction

The rule-based heuristic only makes predictions based on current information, which may lower the quality of the solution. Furthermore, this problem is based on data that is cyclical: the load data this week or today may be similar to the previous week or yesterday. Hence, with previous information, a load prediction model can be built to forecast the future load which can help to control the current operation based on this information. This paper only focuses on the previous day's load information.
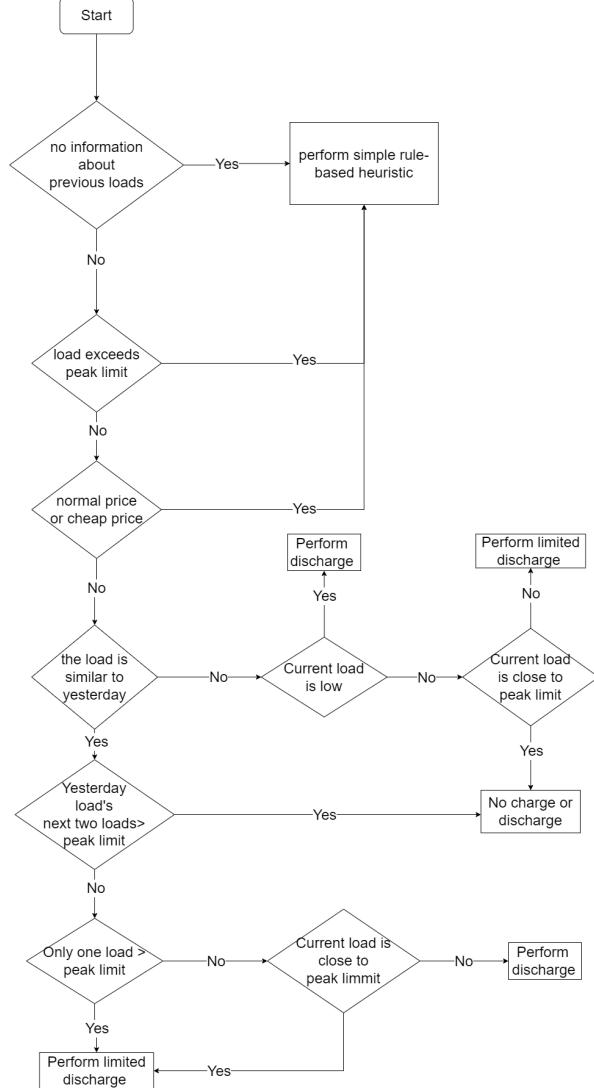
Figure 10: Improved Rolling Horizon Approach

The load prediction model predicts the best operation based on yesterday's data and then the rule-based heuristic follows the result generated by that model. As shown in Figure 10, if the prediction model doesn't have any previous information, the current load is greater than the peak limit, or the current price is at a normal or cheap level, run the previous simple rule-based heuristic. Otherwise, if the price is at a high level and the load doesn't surpass the peak limit, then the load at the current time will be compared with the load yesterday at the same time position. Assuming the difference between the two loads is not significant, the trend of the loads at the same time yesterday may repeat at the current time. Hence, previous information can be used to measure what operation should be performed now. Provided that most of the previous loads have exceeded the peak limit, it is likely that future loads will mostly exceed the peak limit. Hence, given the price when the peak limit is surpassed

is more expensive than the price at a high level, it isn't cost-effective if the battery discharges at the current time interval. Therefore, the model will tell the rule-based heuristic not to perform any charge or discharge operations. Moreover, if only some of the loads of yesterday exceed the peak limit, to make full use of the energy in the battery, do limited discharge which is calculated in a similar way to the limited charge introduced above. Furthermore, if none of the previous loads were greater than the peak limit, then discharge as much as the battery can.

Another case is that the current load is different from yesterday's load at the same time, which implies that the previous information has lost its meaning and becomes useless. Hence, prediction is made only depending on the current load information. Providing that the current load is far away from the peak limit, then it is possible that there are fewer loads which is greater than the peak limit in the future. Hence, it is unnecessary to save energy in the battery and then the battery can perform the discharging operation without artificial limitation. Otherwise, assuming the load is approaching the peak limit, thus it is more likely that many loads will surpass the peak limit in the future. Hence, for the last case, if the load is neither close to the peak limit nor far away from it, then only limited discharge is carried out.

6 Implementation

In this project, most of the hyperparameters-battery capacity, maximum input and output power, and tariff structure-are fixed to simplify the comparison between different models. Only one hyperparameter, the window size, in the improved rolling horizon will be tuned and tested.

6.1 MILP

The MILP model was implemented in IBM ILOG CPLEX Studio using Optimization Problem Language (OPL), which provides a rich set of tools for model development. The MILP method requires an objective function consisting of decision variables to minimize or maximize. Additionally, the value of decision variables shouldn't break any constraints and no slack is allowed. The symbols of the MILP method are shown below.

Symbol	Description
E_{ij}	the energy at time interval i for price level j
P_{ij}	the price at time interval i for price level j
Po	the maximum input and output power of the battery
p_i	the amount of energy that battery charges or discharges at time interval i
L_i	the load at time interval i
B_{start}	the energy in the battery at the beginning
B_{max}	the capacity of the battery
B_i	the energy remained in the battery at time interval i
PL	the peak limit in the tariff structure
N_1	the number of time intervals
N_2	the number of different price levels
C	the total cost of the problem

Table 1: The symbols appear in the model

The decision variable x_{ij} stands for the energy at time interval i and price level j. The price is split into two levels-one price level for energy usage below the peak limit, and another for over the peak limit. The MILP model is defined as below which has one objective function to minimize and several constraints

$$\text{Minimize} \quad C = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} E_{ij} * P_{ij} \qquad\qquad i = 1, ..., N_1 \quad j = 1, ..., N_2$$

$$\text{subject to} \quad -Po \le B_i - B_{i-1} \le Po \qquad\qquad i = 1, ..., N_1$$

$$0 \le B_i \le B_{max} \qquad\qquad i = 1, ..., N_1$$

$$B_0 = B_{start}$$

$$E_{i1} + E_{i2} = L_i + (B_i - B_{i-1}) \qquad i = 1, ..., N_1 \tag{3}$$

$$E_{i1} \le PL \qquad\qquad i = 1, ..., N_1$$

$$0 \le E_{ij} \qquad\qquad i = 1, ..., N_1 \quad j = 1, ..., N_2$$

The objective function calculates the sum of the cost at all the time intervals and the cost at each time interval is the sum of the multiplication of the energy E_{ij} at each price level j and its corresponding price P_{ij}.

The first constraint means that the difference between the remaining energy in the battery at the current time interval and the previous time interval-the amount of charge or discharge $(B_i - B_{i-1})$-shouldn't exceed the maximum input and output power of the battery. $-Po$ stands for the maximum output energy of the battery at each time interval, inversely, Po is the maximum input energy.

The second constraint restricts the status of the battery at all time intervals. Although it is natural that the energy in the battery can never be negative and never exceed the capacity of the battery, it is still necessary to add this constraint to the model. The next constraint set the battery at the beginning empty which is in line with reality when the model is first applied to the BESS prediction control problem.

The remaining three constraints are all for energy in each time interval. The first in these constraints calculates the sum of the load at the current time interval i and the amount of charge or discharge, which represents the total energy imported from the grid at that time interval. Moreover, the energy will be split into two parts based on the next constraint. Since the energy E_{i1} is the energy at the first price level, by the definition E_{i1} should never exceed the peak limit. The last constraint means different parts of the energy at all time intervals E_{ij} should always be greater than zero and in this project, the export component is not considered.

6.2 Dynamic Programming

Since Java language is popular-with many useful library-and robust, Dynamic Programming is implemented using Java. Although Java itself doesn't carry some CSV file readers, importing a package with this capability from a third party can solve this problem quickly and efficiently.

Dynamic programming will divide the battery into different states and each state stores the information about whether the state is available, the remaining energy in the battery B_i, and the total cost incurred when all the loads and the sequence of operations performed from the start to the present time interval i reach this state.

Symbol	Description
N_3	the number of the states at each time interval
ND_{ij}	the state at time interval i and state position j
D	the difference between two neighbourhood state
N_4	the maximum number of the states in one direction can be reached by current state

Table 2: The additional symbols appear in the DP

The user sets the number of states N_3 and the difference D between two neighboring states can be calculated by dividing the battery capacity B_{max} by the number of states N_3. After calculating the difference D, the maximum number of the states N_4 is calculated by rounding off the decimal part of the result of dividing the maximum input and output power Po by the difference D.

Algorithm 1 Dynamic Programming's Core Function

1: **for** $i = 0 \rightarrow N_1 - 1$ **do**
2:     **for** $j = 0 \rightarrow N_3 - 1$ **do**
3:         **if** $ND_{i,j}$ is not Available **then**
4:             Skip to next iteration
5:         **end if**
6:         **for** $action \in ActionSpace$ **do**
7:             **if** $action ==$ NoAction **then**
8:                 $price =$ caculatePrice$(i, 0)$
9:                 **if** $ND_{i,j}.cost + price < ND_{i+1,j}.cost$ **then**
10:                     $ND_{i+1,j}.cost \leftarrow ND_{i,j}.cost + price$
11:                     $ND_{i+1,j}$ is now Available
12:                 **end if**
13:             **end if**
14:             **if** $action ==$ Charge **then**
15:                 **for** $k = 1 \rightarrow N_4$ **do**
16:                     **if** $j + k > N_4 - 1$ **then**
17:                       Skip to next iteration
18:                     **end if**
19:                     $price =$ caculatePrice$(i, k * D)$
20:                     **if** $ND_{i,j}.cost + price < ND_{i+1,j+k}.cost$ **then**
21:                         $ND_{i+1,j+k}.cost \leftarrow ND_{i,j}.cost + price$
22:                         $ND_{i+1,j+k}$ is now Available
23:                     **end if**
24:                 **end for**
25:             **end if**
26:             **if** $action ==$ Discharge **then**
27:                 **for** $k = 1 \rightarrow N_4$ **do**
28:                     **if** $j - k < 0$ **then**
29:                       Skip to next iteration
30:                     **end if**
31:                     $price =$ caculatePrice$(i, -k * D)$
32:                     **if** $ND_{i,j}.cost + price < ND_{i+1,j-k}.cost$ **then**
33:                         $ND_{i+1,j-k}.cost \leftarrow ND_{i,j}.cost + price$
34:                         $ND_{i+1,j-k}$ is now Available
35:                     **end if**
36:                 **end for**
37:             **end if**
38:         **end for**
39:     **end for**
40: **end for**

The pseudocode above shows the core function of dynamic programming. The state j at each time interval i stands for a sub-problem $ND_{i,j}$-look for the minimum cost at the current time interval with fixed remaining energy in the battery at the current state. Since this paper uses bottom up dynamic programming, the model will traverse the sub-problems in all states from the beginning time interval to the final time interval and each sub-problem's solution contributes to the next optimal solution of the sub-problem in the next time interval. As long as the state at that time interval is available, which means the sub-problem at that state has the optimal solution and can form a global optimal to the sub-problems at the next time interval. Hence, three actions can be taken by the model-charge, discharge, and no operation. The maximum number of the states N_4 in one direction has been calculated, which stands for the different amounts of energy that can be charged or discharged from the battery. At each state traverse all the possible actions and calculate the cost of the solution, compare it with the current solution's cost stored in the target state, choose the best one, and store it for the next calculation.

6.2.1 Pruning Strategies

The pruning strategies can be applied to dynamic programming to improve the efficiency of the model. At the beginning of traversing the states at each time interval, a flag is used to describe the status at the current time. When the price is cheap at the time interval, under no circumstance should any discharge operation be performed because it's not cost-effective. Therefore, the flag will be set to inform the model not to consider the related calculation. On the contrary, if the price is expensive at that time or the load has exceeded the peak limit which drives up the cost (two constraints have been combined when setting the value of the flag), don't consider any charge behavior which greatly increases the cost. For the same reason, the flag will tell the model not to do the calculations.

6.3 Rolling Horizon Approach

Due to differences in performance between platforms and languages, in order to avoid the impact of using too many different platforms and languages on the difference in calculation results, the rolling horizon approach is implemented using the Java language like Dynamic Programming.

Symbol	Description
μ	the size of the window (or fixed subproblem)
ε	The portion of relaxation which is allowed for the new solution

Table 3: The additional symbols appear in the DP

μ is a hyperparameter that will be tuned in the project because the size of the window and subproblem has an impact on the quality of the solution. ε is another hyperparameter but it is fixed due to the lack of time to tune and compare different values' influence.

Algorithm 2 Improved Rolling Horizon approaches [15]

INPUT: A sequence of minimization problems $P_{0,\mathbf{T}}, \mu \leq \mathbf{T}, \varepsilon > 0$
OUTPUT: A solution $(x_t)_{t\in[\mathbf{T}]}$ of $P_{\mathbf{T}}$
1: **procedure** CAREFUL FINITE FORESIGHT OPTIMIZATION $(P_{\mathbf{T}}, \mu, \varepsilon)$
2:      $\beta \leftarrow N_1 - \lfloor N_1 - 1 \rfloor_\mu$
3:      $(x_t)_{0\leq t<\mu} \leftarrow \operatorname{argmin} P_{\mu-1}$
4:      s.t. $P_{\beta-1} \leq (1+\varepsilon)z_{\beta-1}$
5:      **for** $t \in \{1,...,\beta\}$ **do**
6:          $(x_j)_{t\leq j<t+\mu} \leftarrow \operatorname{argmin} P_{t+\mu-1}$
7:          s.t. $P_{\beta-1}^{new} \leq (1+\varepsilon)z_{\beta-1}$
8:      **end for**
9:      **for** $t \in \{0,...,\lfloor \frac{N_1-1}{\mu} \rfloor - 2\}$ **do**
10:          **for** $j \in \{\beta+1,...,\mu+\beta\}$ **do**
11:              $(x_j)_{\mu t+j\leq l<\mu(t+1)+j} \leftarrow \operatorname{argmin} P_{\mu(t+1)+\beta-1}$
12:              s.t. $P_{\mu(t+1)+\beta-1}^{new} \leq (1+\varepsilon)z_{\mu(t+1)+\beta-1}$
13:          **end for**

14:     **end for**
15:     **return** $(x_t)_{t \in [\mathbf{T}]}$
16: **end procedure**

$\lfloor N_1\text{-}\mathbf{1} \rfloor_\mu$ represents the maximum size of the number of the fixed problems generated by the model according to the window size mu. Hence, β is the number of remaining time intervals which is not enough to satisfy the window size μ but has to be considered. The algorithm will first determine the sequence of operations x_t of the subproblem within window size μ using some future information that contains the operations of the fixed subproblem $P_{\beta-1}$. Afterward, start a loop to traverse all the time intervals in the previous fixed subproblem to compare the existing solution and solution with a combination of previous information and future information at each time interval. After the loop, all the operations in the previous fixed subproblem have been determined. Hence, the remaining elements divided into fixed subproblems with the same size of the window size μ can use a nested loop to determine and change the operations in the solution of each fixed subproblem. The reason for the nested loop is the same as the previous one.

6.3.1 Dynamic Programming

When implementing the rolling horizon approach, a problem-specific algorithm needs to be determined to solve the subproblem within the window. At first, dynamic programming is taken as that algorithm in the model, but the overall running time of the model is lengthy. Although the paper [15] describes the rolling method as an efficient algorithm, in practice, dynamic programming actually makes the running time longer.
The running time of the dynamic programming:

$$f(n) = O(N_1 N_3 N_4) \tag{4}$$

When calculating the complexity, the details of the price calculation, comparison, and judgment are treated as $O(1)$. The total number of states the model will traverse is $O(N_1 N_3)$ and the number of the states N_4 in one direction which can be reached by each state (the actual number of the states can be reached by one state is $2N_3+1$, and the factors and constant can be ignored). Hence, the time complexity is $O(N_1 N_3 N_4)$
The running time of the rolling horizon approach based on dynamic programming:

$$f(n) = O(\mu N_1 N_3 N_4) \tag{5}$$

N_1 represents the number of subproblems to be solved. Since at each time the window only forward by 1 step, hence the number of subproblems only differs from the number of time intervals by a small constant which can be ignored in time complexity. It is important that the number of fixed subproblems is ignored since it is just a constant. μ represents the number of time intervals of the subproblem, N_3 stands for the number of states in the subproblems and N_4 stands for the number of the states which can be reached by each state. Hence, the total complexity can be calculated by multiplying the number of subproblems N_1 and the time complexity of the subproblems $O(\mu N_3 N_4)$.

Consequently, comparing dynamic programming and the rolling horizon approach based on dynamic programming, it is clear that applying the rolling horizon approach to dynamic programming can't improve efficiency.

6.3.2 MILP

Since the rolling horizon approach based on dynamic programming lost its efficiency of what it should be, this project changed into another problem-specific algorithm MILP to increase the speed of running. Additionally, when implementing this rolling approach, a CPLEX jar package from IBM is imported into the project to assist the model. The worst time complexity of the MILP is usually exponential. However, the actual time complexity highly depends on other factors-the problem structure and the solver algorithm.

The details of the rolling horizon approach implementation have also been changed, but the

6.4 Rule-based Heuristic

The total cost is calculated by summing up the cost by following the rules at each time interval. The load and current tariff structure will be considered in the rules and the model will attempt to generate a reasonable decision at each time interval. When the load doesn't exceed the peak limit and the price level is normal, the amount of energy a battery can charge when other constraints are not considered is calculated:

$$RE_i = Po\frac{L_i}{PL} \tag{6}$$

RE_i stands for the maximum amount of energy the battery can charge at that time interval. If the load at time interval i is approaching the peak limit, it is more likely that the load will rise in the future and exceed the peak limit. Hence, the battery should charge as much as it can when the price is at a normal level and discharge for the excessive in the future to reduce the cost. Consequently, $\frac{L_i}{PL}$ is used to adjust the amount of energy to charge. Apparently, the load L_i plus the energy to charge should not surpass the peak limit PL and the amount of charge should not exceed the battery capacity B_{max}.

When the load exceeds the peak limit, discharge for the excessive part. The rules are expressed below:

$$DE_i = -\min(L_i - PL, Po, B_i) \tag{7}$$

First, the amount of energy to discharge DE_I shouldn't be greater than the maximum output power and the remaining energy in the battery. Afterward, discharge as much as the portion of the load that exceeds the peak limit.

When the price is at a high level, the amount of energy to discharge is:

$$RE_i = -Po\frac{B_i}{B_{max}} \tag{8}$$

If the remaining energy in the battery is little, it is more cost-effective to keep the energy and discharge when the load exceeds the peak limit since the price for the peak limit is higher. Consequently, $\frac{B_i}{B_{max}}$ is used to adjust the amount of energy to discharge. Additionally, this energy should also be compared with the remaining energy in the battery and the load.

When the price is cheap, the amount of energy to charge is:

$$CE_I = min(PL - L_i, Po, B_{max} - B_i) \tag{9}$$

The amount of energy to charge should be as much as possible when the price is cheap. However, there are some constraints to obey. Initially, $PL - L_i$ means that the energy of charge plus the load at the current time interval L_i should not exceed the peak limit which will increase the cost. Moreover, $B_{max} - B_i$ represents that the sum of the amount of the charge and the remaining energy in the battery should not exceed the battery capacity.

Though the rule-based heuristic is more efficient and uses less memory, the quality of the solution needs to be improved. Hence, a prediction model is introduced in the heuristic.

6.4.1 Rule-based Heuristic with Prediction

In the prediction component, in addition to the information on the current load, the previous loads' information from the same position yesterday is used to predict the trend of the future loads and improve the quality of the solution. The prediction model is composed of many conditional expressions, and each of the expressions will return a flag that can instruct the model to proceed to the next step.

At first, the prediction model will examine whether there is any previously available information that can be utilized. If such information doesn't exist, then return a flag to inform the heuristic to

run a simple rule-based heuristic using only current load information to make an operation. On the contrary, if the prediction model finds out that the current load has surpassed the peak limit or the price is at a normal or cheap level, the prediction model will return a flag to instruct the heuristic to perform as the simple rule-based heuristic.

For a special case when the price is set to a high level, a dilemma arises: if the loads in the future will exceed the peak limit, the battery energy should be saved to use in the future; Contrarily, if the loads won't be greater than the peak, the battery should discharge to cut the cost. Hence, it is vital to predict the situation of the future using previous information.

However, information is time-sensitive which means the information may have lost its original meaning at the present time. For instance, if a customer uses two air conditioners for a week due to the high temperature, which keeps the load high for the previous week. However, after the week the customer decides not to turn on the air conditioners, the load levels will return to normal. Hence, the previous information loses its usage to predict the future. Under such circumstances, the prediction model has to judge whether the information is still valid. Therefore, the first step is to estimate the validity of the information. If the difference between the current load and the previous load at the same time yesterday is greater than a threshold ($0.4\ PL$), the previous information is probably no longer usable. Hence, make a prediction based on the current load information.

$$Predict(t) = \begin{cases} \text{Fully Discharge,} & \text{if } L_i < 0.5PL \\ \text{No Charge or Discharge,} & \text{if } L_i > 0.8PL \\ \text{Limited Discharge,} & \text{otherwise} \end{cases}$$

If the current load is far away from the peak limit, then assume the load will keep low in the future for a time. Hence, return a flag that tells the heuristic discharge as much as possible. Similarly, if the current load is close to the peak limit, it is possible that in the future some loads may surpass the peak limit, thus no operation for the battery is allowed. Otherwise, it is hard to predict, consequently, a limited discharge flag is returned to the heuristic.

For the other, the difference between the current load and the previous load is less than this threshold, the information is more likely to be useful to help with prediction. Hence, the next two loads of the previous load are picked.

$$Predict(t) = \begin{cases} \text{No Charge or Discharge,} & \text{if } L_{i-23} > PL \text{ and } L_{i-22} > PL \\ \text{Fully Discharge,} & \text{else if } L_{i-23} < PL \text{ and } L_{i-22} < PL \\ \quad \text{and } L_i < 0.8PL \\ \text{Limited Discharge,} & \text{else} \end{cases}$$

7 Evaluation

References

[1] Executive summary – Electricity Market Report - July 2022 – Analysis. https://www.iea.org/reports/electricity-market-report-july-2022/executive-summary, January 1970.

[2] The price of batteries has declined by 97% in the last three decades. https://ourworldindata.org/battery-price-decline, January 1970.

[3] F Abou Chacra, P Bastard, G Fleury, and R Clavreul. Optimisation multiobjectifs du stockage d'energie dans un poste source htb-hta. In *eletronic proceedings of EF2003 conference-SUPELEC-Gif sur Yvette-France*, 2003.

[4] Umberto Abronzini, Ciro Attaianese, Matilde D'Arpino, Mauro Di Monaco, and Giuseppe Tomasso. Cost minimization energy control including battery aging for multi-source ev charging station. *Electronics*, 8(1), 2019.

[5] Vahid Beiranvand, Warren Hare, and Yves Lucet. Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18, 12 2017.

[6] Paul Bolton and Lona Stewart. Domestic energy prices - house of commons library.

[7] Michael Alexander Campbell. Impacts of pricing structures on home battery storage operation and economics. 2021.

[8] F.A. Chacra, P. Bastard, G. Fleury, and R. Clavreul. Impact of energy storage costs on economical performance in a distribution substation. *IEEE Transactions on Power Systems*, 20(2):684–691, 2005.

[9] Nigel Charig. BESS: These factors are driving the growing demand. https://www.power-and-beyond.com/bess-these-factors-are-driving-the-growing-demand-a-bba2519b07b900034a5720811d44da60/, April 2022.

[10] K. C. Divya and Jacob Østergaard. Battery energy storage technology for power systems—An overview. *Electric Power Systems Research*, 79(4):511–520, April 2009.

[11] KC Divya and Jacob Østergaard. Battery energy storage technology for power systems—an overview. *Electric power systems research*, 79(4):511–520, 2009.

[12] Melike Erol-Kantarci and Hussein T. Mouftah. TOU-Aware Energy Management and Wireless Sensor Networks for Reducing Peak Load in Smart Grids. In *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1–5, September 2010.

[13] Muhammad Farooq. The model predictive controls for battery energy storage systems based upon dynamic programming. dissertation, 2022.

[14] Eskil Sulen Gjerde. *Novel Approaches for Optimal Control of Battery Energy Storage Systems*. dissertation, 2021.

[15] Lukas Glomb, Frauke Liers, and Florian Rösel. A rolling-horizon approach for multi-period optimization. *European Journal of Operational Research*, 300(1):189–206, 2022.

[16] Jason Lee. Linear estimation of degradation of Lithium-Ion battery base on constant temperature approach.

[17] Héricles Eduardo Oliveira Farias, Camilo Alberto Sepulveda Rangel, Leonardo Weber Stringini, Luciane Neves Canha, Daniel Pegoraro Bertineti, Wagner da Silva Brignol, and Zeno Iensen Nadal. Combined Framework with Heuristic Programming and Rule-Based Strategies for Scheduling and Real Time Operation in Electric Vehicle Charging Stations. *Energies*, 14(5):1370, January 2021.

[18] Peterson Ozili and Ercan Ozen. Global Energy Crisis: Impact on the Global Economy. December 2022.

[19] Ehsan Reihani, Saeed Sepasi, Leon R. Roose, and Marc Matsuura. Energy management at the distribution grid using a Battery Energy Storage System (BESS). *International Journal of Electrical Power & Energy Systems*, 77:337–344, May 2016.

[20] Guillem Rigaill. Pruned dynamic programming for optimal multiple change-point detection. *arXiv*, 04 2010.

[21] Marco Stecca, Laura Ramirez Elizondo, Thiago Batista Soeiro, Pavol Bauer, and Peter Palensky. A comprehensive review of the integration of battery energy storage systems into distribution networks. *IEEE Open Journal of the Industrial Electronics Society*, 1:46–65, 2020.

[22] Faraz Syed, Michael Fowler, David Wan, and Yaser Maniyali. An energy demand model for a fleet of plug-in fuel cell vehicles and commercial building interfaced with a clean energy hub. *International Journal of Hydrogen Energy*, 35(10):5154–5163, May 2010.

[23] Daniel Teplan. Time of Use Tariff UK 2022 - Overview & Provider Comparison. https://justwe-gpi.com/energy-economy/time-of-use-tariff-uk/, December 2021.

[24] Mark Z. Jacobson. Review of solutions to global warming, air pollution, and energy security. *Energy & Environmental Science*, 2(2):148–173, 2009.

[25] Chengke Zhou, Kejun Qian, Malcolm Allan, and Wenjun Zhou. Modeling of the cost of ev battery wear due to v2g application in power systems. *IEEE Transactions on Energy Conversion*, 26(4):1041–1050, 2011.