Politecnico di Torino

Politecnico di Torino

Master's Degree in ICT for Smart Societies
a.a. 2021/2022

Heuristic Optimization Approaches for the Smart Home Appliance Scheduling Problem

Supervisors

Prof. Emilio LEONARDI

Prof. Edoardo FADDA

Candidate

Andrea MINARDI

Abstract

With the emerging of smart grids as an answer to the need of sustainability, it grows in relevance the ability to manage the energy at our disposal. This is of utmost importance in residential settings where the demand is dependant on the family behaviors and the possibility of having other distributed energy sources (i.e. Solar Panels, Plug-In Electric Vehicles). Reliable energy Management Systems help in this task by scheduling the appliances in order to optimize a specific aspect of the energy demand. This thesis studies the effect of different optimization approaches for the power scheduling problem in residential settings. We start from a Mixed-Integer Linear Program as a base and build an heuristic approach based on the Tabu Search algorithm, improved with an diversification techniques. Moreover, a Markov Decision Process formulation of the problem is developed. The different approaches are then compared and analyzed with different pricing schemes and different Renewable Energy Sources settings.

Summary

This thesis presents a study in the Smart Grid field aimed at understanding the effects of different optimization methodologies on the Smart Home Appliance Scheduling Problem. The problems goal is to find the best possible starting times for each of the appliances within the residential environment, in order to optimize a predefined objective function. It is a complex problem since it tries to outline the best possible actions to take in order to minimize the overall impact of electricity cost on the household without impacting family behavior. This is a challenging task due to various reasons: nowadays the number of devices that draw power in the household grows continuously and each of them has its own usage patterns, power rating and time needed to complete its task. A possible help comes from the usage of distributed energy resources (DER) such as renewable energy sources (RES) like photovoltaic panels and wind turbine or plug-in electric vehicles (PHEV), which behave as batteries once plugged in. These DER help the customer to amortize the electricity consumption and do not incur in power outages. Nevertheless, the main issue is their stochasticity, which is inherent in their generation methodologies: for example, RES such as Solar Panels are highly dependent on weather conditions, or PHEV depends on consumers' driving patterns. Due to the large scope of possible cases, in this thesis we consider only scheduling of shiftable appliances without preemption and an array of Solar Panels in support of the power drawn from the grid that generates power ideally, with no losses. To address the aforementioned scheduling problem, there are many possible methodologies that can be used. The aim of this thesis is to study the state of the art methodologies to solve the Smart Home Appliance Scheduling Problem, with deeper attention to heuristic approaches such as MetaHeuristic, namely Tabu Search, and Reinforcement Learning.

Metaheuristic approaches gained a lot of traction in the optimization field since, for this kind of problems, speed of computation is more important than getting an exact solution. This is especially true if the pricing scheme used by the utility company is not day-ahead but in real time (i.e. if the price changes on an hourly basis). As a result, an enhanced Tabu Search has been developed to solve this issue: whenever the method finds itself in a local minima, diversification is used to help exploration, restarting the algorithm from a random solution in the search

space. Several analyses are then performed to understand the behavior of the metaheuristic algorithm with respect to the more common off the shelf MILP optimization technique.

A reinforcement learning approach is also proposed. RL is well suited for problems described in discrete time, where there is a well defined cost function that can be used as a reward signal for the agent to learn an optimal policy. Eventually, due to the high dimensionality of the problem under exam, a Markov Decision Process formulation for the smart home environment was non trivial to be achieved and the results are still suboptimal: in order to find the best policy for a particular set of working appliances, a lot of samples are needed and the inherent stochasticity of the problem makes this solution problematic.

Acknowledgements

First of all, I'd like to thank my supervisors, professor Emilio Leonardi and professor Edoardo Fadda, for giving me a chance to work on such a challenging thesis, for their support and advice during these months.

I want to thank also the Mu Nu Chapter of IEEE-HKN for helping me grow beyond my expectations.It truly has been an honor meeting you.

It's reductive to thank the Green House Family of the Study Room 3 as a whole but each and everyone of you has been of such an importance that I couldn't do otherwise. Thank you, from the bottom of my heart, for giving me love when I felt hatred, warmth when I felt cold and friendship when I felt lonely.

Thanks to my mother Amalia for listening me when I thought I no longer had a voice and advising me, even when I didn't want to listen.

Thanks to my father Martino for believing in me when I couldn't and showing me that dreams do come true when you really believe in them

At last, this thesis is dedicated to my brother Giuseppe who more than anyone knows how much blood sweat and tears I poured to get where I am now. You made me feel valued when I felt worthless and that is a favour that I can not repay. So thank you, I love you, now don't get carried away, I still have more hair than you.

"The only person who can sympathise
with you and understand you is you.
So be good to yourself"

Table of Contents

List of Tables

List of Figures

Acronyms

1TP One-Tier Pricing

2TP Two-Tier Pricing

3TP Two-Tier Pricing

AMI Advanced Metering Infrastructure

CPP Critical Peak Pricing

DER Distributed Energy Resources

DP Dynamic Programming

DR Demand Response

DSM Demand Side Management

EC Electricity Cost

EV Electric Vehicle

ICT Information and Communication Technology

ILP Integer Linear Programming

HEMS Home Energy Management System

LP Linear Programming

LS Local Search

MDP Markov Decision Process

MILP Mixed-Integer Linear Programming

NIST ational Institute of Standard and Technology

NLP Non Linear Programming

PAR Peak-To-Average Ratio

PHEV Plug-In Hybrid Electric Vehicle

PMU Phasor Measurement Unit

PV Photo Voltaic

RES Renewable Energy Sources

RL Reinforcement Learning

RT Real-Time Pricing

ToUP Time-Of-Use Pricing

TS Tabu Search

SHASP Smart Home Appliance Scheduling Problem

SG Smart Grid

WAMS Wide Area Monitoring System

US User Satisfaction

Chapter 1

Introduction

This thesis focuses on possible application of reinforcement learning methods in a smart grid framework.

1.1 Motivation

As the world tries to move on from carbon fuels, electricity and "green" fuels generation and utilization become of paramount importance in how fast this sustainable transition has to be. in this, the energy consumption from residential buildings accounts for a large portion of the global energy consumption. This means that understanding how the energy is used in our homes is the key to a smarter distribution of electricity within the grid.

The bidirectional data flow between each endpoint in the Smart Grid, allows the utility companies to optimize each consumer's electricity usage, in light of the fact that the addition of Renewable Energy Sources is now more and more common and each customer takes the role of a energy produces. This relives some of the pressure off of the Utility Companies that now can use smart planning to deliver the right amount of electricity when it's needed to each building, based on the customers behaviour. It is well known that, by varying the prices, it's possible to shift the demand, reduce the risk of power outages, reduce the emissions and minimize the cost of production of electricity.

A way to tackle this problem is the usage of automated scheduling systems which aims at minimizing the electricity consumption within a household by scheduling the

appliance usage during off-peak hours, while keeping track of operation constraints and consumer preferences.

1.2 Structure of the Thesis

This section defines the structures of the thesis:

Chapter 1 - Introduction

Current Chapter with the motivation behind the work here presented and

Chapter 2 - Thesis Framework

This chapter focuses on giving a detailed description of the main concepts behind the presented research. It's aim is to make this work as self-contained This chapter explains the methodologies that can be used to

as possible giving the information needed to a reader to understand why the problem is of such importance and why the field of research is so thriving. The chapter consists of two main parts:

- a section where the overall concept of Smart Grid is explained, focusing on why is important, its characteristics, the differences between with the traditional paradigm and the main techniques that electricity utilises use to adapt to different demand and supply.
- a detailed explanation of what a Smart Home is, its characteristics and how intelligent management systems can help organize all the resources and the characteristics of the elements that compose a Smart Home Environment.

Chapter 3 - Optimization Strategies for Smart Home Appliance Scheduling Problem

This chapter explains the methodologies that can be used to solve the Smart Home Appliance Scheduling Problem. Here is given an overview of the major optimization

techniques for resolving the problem, such as Mixed-Integer Linear Programming, heuristics and meta-heuristics and Markov Decision Process based approaches. The chapter is divided in four sections: at first the problem is presented with a description of the variables involved. The second subsection describes the Mixed-Integer Linear Programming of the SHASP, along with the formulation of the cost function to be minimized and the constraints the problem is subject to. Then the Local Search heuristics are introduced and is given a brief explanation of how they work and some of the notable algorithm are introduced. Given that the choice of algorithm fell on Tabu Search, an explanation is given of its workings, how it was adapted to the problem and what improvements were made to the basic algorithm in order to achieve a satisfactory result. Eventually also reinforcement learning has been studied as a possible approach to the problem. An introduction to the Markov Decision Process and reinforcement learning is detailed and a tentative design of a MDP referred to the SHASP is formulated.

Chapter 4 - Results

Second contribution of this thesis is the comparison between the MILP and the novel TS-based algorithm developed. In this chapter the results are presented by comparing the two models in two different settings, one where the Smart Home has auxiliary power generation, supported by solar power generation and one where no RES is attached to the house power system.

Chapter 5 - Conclusions

This chapter provides a summary of the research carried out. Moreover, the thesis is concluded with some insights on how future research can be done using this work as a foundation.

Chapter 2

Thesis Framework

2.1 Smart Grid

Smart Grids are a prevalent framework for the distribution of electricity in smart cities. The grid refers to the electric grid, the network of transmission lines that delivers electricity from a power plant to a consumer to be used for various purposes, be it industrial, residential or commercial. At the moment, begin smart grids still a new technology, there is not a definitive standard for their development and every nation is pushing to impose its own reference model as international standard

The National Institute of Standard and technology (NIST) [1] defines a Smart Grid as:

> a modernization of the electricity delivery system so it monitors, protects and automatically optimizes the operation of its interconnected elements – from the central and distributed generator through the high-voltage network and distribution system, to industrial users and building automation systems, to energy storage installations and to end-use consumers and their thermostats, electric vehicles, appliances and other household devices.

2.1.1 Characteristics

The smart grid is the modern way of building the electric infrastructure of the cities. Equipping ICT devices, enables the grid to smoothly include renewable energy source (RES) allows to sense with high accuracy, monitor and manage the stability of the power system. The usage of ICTs and intelligent controllers enable the automation of the power network [2]. This is crucial since the addition of distributed energy generation with the use of RES leads to a more reliable, safer and manageable energy generation network [3].

The usage of a distributed energy generation system allows for a more widespread usage of greener electricity. The residential sector, even more than the industrial one, is the one that contributes more to the load on the smart grid: each consumer has little awareness of how much their action impact both the stability of the grid and the overall cost of the electricity. It becomes important how to manage the appliances in order to effectively reduce the gap between the demand asked by the end-users and the supply generated by the elements in the grid. [4]. The management of the appliances within each home connected to the grid is done through a smart scheduling based on the power supplied by the utility.

2.1.2 Traditional vs Emerging Paradigm

A comparison between the traditional grid and the smart grid scan be shown in table 2.1 where the two paradigms are compared in different aspects such as technology, metering infrastructure, generation of electricity, monitoring, fault sensing and management.

The main elements can make a traditional grid *"smart"* are:

Distributed Energy Resources (DER) the usage of small-scale power generators (between 3 kW and 10 MW) that add power to the public distribution grid and help balancing the load;

Advanced Metering Infrastructures (AMI) smart meters, data management systems, communication networks help provide the bidirectional communication between the consumer and the provider to ensure a better distribution of electricity.

Phasor Measurement Units (PMU) devices that measure the electrical waves on an electricity grid, enhancing the measurements on the power lines to detect

Table 2.1: Comparison between smart grid and traditional grid

Characteristics	Traditional Grid	Smart Grid
Power Generation	Centralized at the power plants	Distributed, with RES, PHEV and the plants
Information flow	One way: from utility to consumer	Two way: utility and consumer effectively communicate both power and information
Monitoring	Manual	Self-monitoring using digital technology
Topology	Radial	Network
Fault management	Failures break the network, power is cut off	The power network is rerouted in order to avoid the failure and ensure power distribution
e Recovery	The failure is restored manually	Self-healing techniques are used to recover from failures without human intervention
Sensor	There is little equipment on the power lines	Multiple sensors used over the power lines to sense possible failures and prevent them
Metering Infrastructure	Meter Readings done manually	Smart Meters make advances readings both for the user and the utility (Advanced Metering Infrastructure)

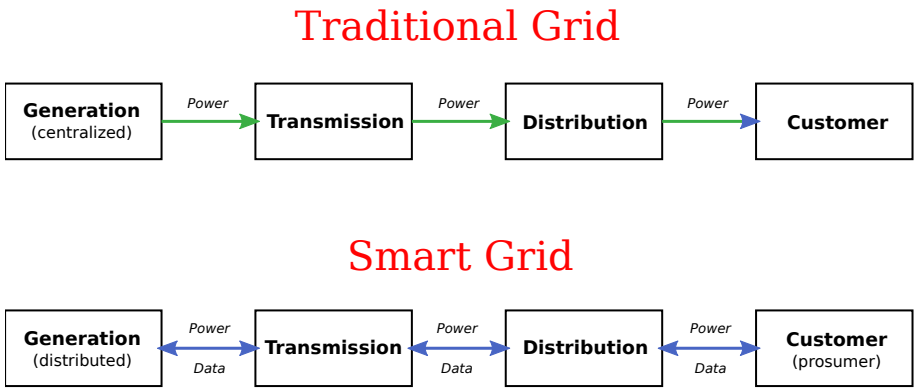

Figure 2.1: Comparison between Traditional and Smart Grid Flow of information and power

failures. the measurement done by this devices are then sent to a **Wide Area Monitoring System** (WAMS) in order to be evaluated;

demand Response (DR) Incentives the customers initiate a change in their consumption behaviour by modifying cost patterns, giving ecological information or simply favouring a different way of consuming electricity. Demand Response belongs to a wide variety of policies called Demand Side Management (DSM);

2.1.3 Demand Side Management

Demand Side Managemnet (DSM) consist in a series of initiatives that the electricity utilities implement to lead customers in taking smarter practices that are advantageous for both parties. These practices aim at changing the load shapes by influencing the behaviour of the user. A drawback of these mechanisms is the rapid increase in complexity of the power system since the presence of a capillar sensing networks is of paramount importance in the choice of the correct policies and techniques to implement. Monitoring and then controlling the whole grid at different levels becomes now a more challenging task than ever. The evolution of the grid helps the implementation of these mechanisms since the bi–directional flow of data and the increase in information shared helps in choosing the correct plans for each consumer.

In figure 2.2 are represented six different DSM techniques in which the load curves of the residential demand are altered between on-peak and off-peak duration [5].

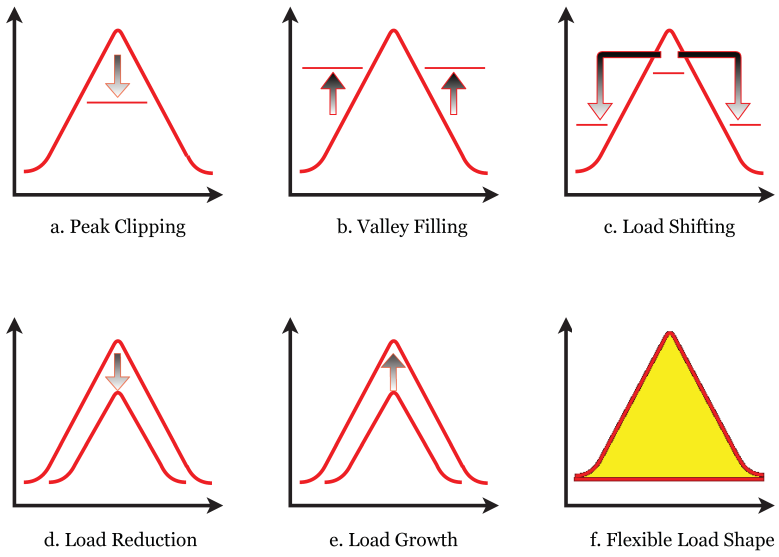

Figure 2.2: Demand Side Management Techniques, image elaborated from Shewale et al. [5]

a **Peak Clipping**: Direct Load technique that aims at reducing the peak in the demand curve.

b **Valley Filling** Opposed to the peak clipping technique, valley filling focuses on increasing energy consumption during off-hours.

c **Load Shifting**: Trade-off between Valley Filling and Peak Clipping, main aim is to shift the load from peak to off-peak periods. This is done by introducing tariffs that encourage using appliances in specific time frames. Most common DSM technique.

d **Load Reduction** Also called Strategic Conservation, the desired effect is for the demand curve to be reduced. This is done though the use of efficient appliances or less use of electricity overall.

e **Load Growth** Also called Load Building, the idea is that the user is encouraged into consuming more energy within a certain limit. This is done in order to maintain power system capacities and a smoother operation of the power system.

f **Flexible Load Shape**: The consumers are incentivised, through specific contracts and tariffs, in redistribute the loads to various time slots.

A main issue with the growing population is that the demand is difficult to be satisfied, while also taking into account the stress on the grid during peak hours and the global warming and green house emissions. The usage of more RES in residential settings, electric vehicles and demand response program, the demands is well balances by the supply.

The multiple DSM techniques listed above can be divided in **energy efficiency** programs and **demand response** programs [6].

Energy Efficiency The aim is to minimize the electricity usage through the usage of energy efficient house-appliances and building envelopes. This type of approach can decrease the demand during any time of the day, not only in peak hours. In this category also belong maintenance of commonly used electrical equipment so as to reduce waste of energy due to malfunctions

Demand Response it is the usage of particular pricing policies that shape the behaviour of the users, influencing them into consuming electricity in off-peak periods. DR can be defined as "the changes in electric usage by end-use consumers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale prices or when system reliability is jeopardized" [7]. Demand Response are the most common DSM techniques do to their ability to affect the load directly.

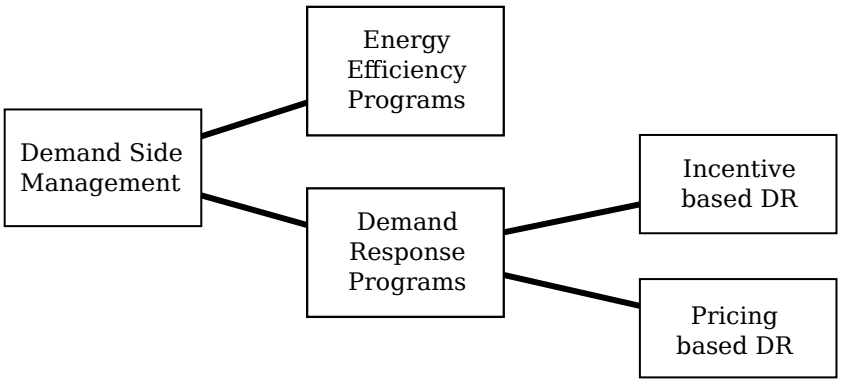

Figure 2.3: Demand Side Management (DSM) techniques classification

Demand Response

Demand Response programs are classified into **Incentive-based DR** and **Price-Based DR** [8].

In *Incentive-based DR*, the utility is the one in control of the loads and the consumer allows it in return of incentives from the provider. Among these can be seen:

Direct Load Control The utility directly interrupts or reduces the users power supply during peak demand times. The consumer is still notified before it happens and in return it receives a compensation

Interruptible Load usually seen in industrial and commercial settings, the provider can turn off for a short period of time their load. The consumer can either follow through with the interruption and it is compensated or it's penalized if it decides to not shut down it's load.

Emergency Reduction During system contingencies the user is asked to reduce it's demand in order to alleviate the pressure on the grid. There is no penalty here if the costumer does not follow through with the request.

Price-Based DR instead use time-varying tariffs to incentives the user to follow virtuous energy consumption plans. The responsibility is all on the user. The pricing schemes for a price-based DR program include:

Time-Of-Use Pricing (ToUP) The day is divided in time blocks (example eight-hour block). Typically the blocks are either two or three, dividing the day

in *peak*, *off-peak* and optionally *mid-peak*. The price differs from one period to the other with higher price in the peak period and a lower price in the off-peak period. The consumer will aim at minimizing the energy cost and will minimize its energy usage in peak hours, shifting it into lower rate hours

Critical Peak pricing (CPP) This type of plan is specific for high demand homes (more than 20 kW). Periods where the electricity consumption exceeds that thresholds are called *critical*. Such periods are forecasted the day before and during said periods the price is increased, much like in Time-Of-Use. In this way, the consumer should shift the load outside of the critical period.

Real-Time Pricing (RT) More complex than the previous ones. Here the tariff can vary daily or even hourly. There are two types of RTP schemes: *day-ahead pricing* and *hourly pricing*. In day-ahead the price details are disclosed the day before for the day after, whereas in the hourly pricing scheme the price is revealed the hour before.

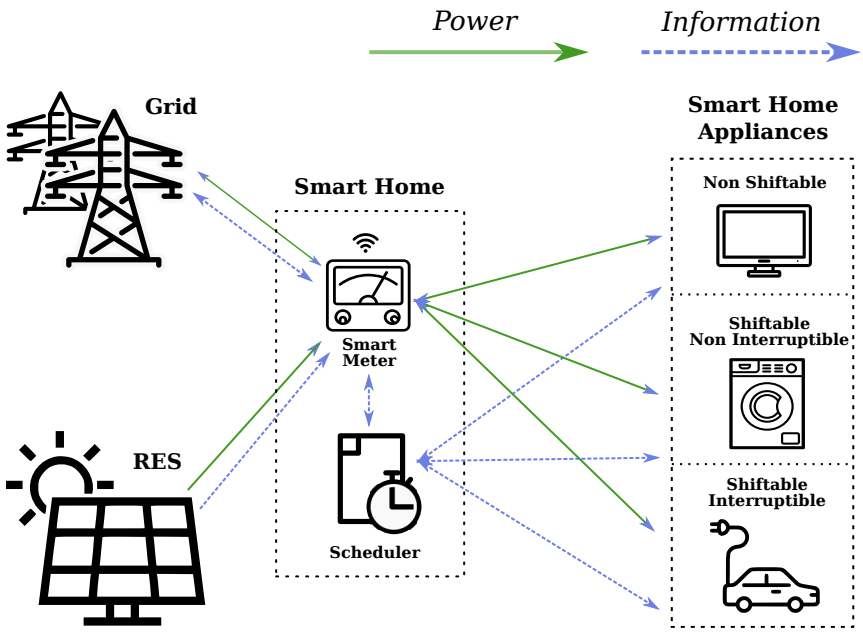

Figure 2.4: Home Energy Management System

2.1.4 Home Energy Management System (HEMS)

The electricity consumed in the residential sector is increasing due to today's lifestyle where more an more devices are used to have a better Quality of Life.

This is at the expanse of the electricity costs on the consumers and the burden that houses have on the overall grid load. In fact more than 30% of the electricity consumption is due to residential activities [9, 10]. Demand side management of smart homes becomes one of the main ways in which it is possible to meet the demand without increasing the supply where it is no more manageable. DR programs are used to manage the energy in the home settings. This is done through advanced metering infrastructure that helps both the user and the utility provider to understand the consumption patterns of the users and meet their demand through tailored plans. A Home Energy Management System uses the data provided by the AMI to help consumers schedule their appliances in order to minimize electricity consumption and maximize their comfort. a HEMS work consist of sending signals to the smart home controllers so that appliances starting times are shifted and power outages are avoided. Advanced programs also minimize the cost consumption, scheduling the appliances in off peak periods, maximizing the usage of Renewable Energy Sources or efficiently introducing more complex elements in the environment such as the recharge of electric vehicles or the control of air conditioning systems. Figure 2.4 depicts how the HEMS functions and their main components. In that model we considered the utility grid, a renewable energy source in the photo voltaic panel, a smart meter, a scheduler and three classes of smart home appliances (a detailed explanation on the subdivision is given in section 2.1.4). Being the System "smart", all the elements in it exchange both power and information. Through the smart meter, all the information that comes from the utility grid is met by the information coming from the distributed energy resources and the running smart appliances. By analyzing the data that comes from all the different points, the scheduler communicates with the smart appliances, shifting their power over the time horizon to accommodate for the behaviour of the user and the needs of the utility provider.

Appliance Characteristics

The household appliances and various devices that make up the smart home system can be modeled based on their consumption characteristics and time of use. There are multiple ways in which appliances can be classified. In Wang and Zheng [11] the appliances are divides in three categories based on heir main power consumption unit and working styles. A subdivision can also be done based on device operation characteristic like in Kim and Poor [12] where devices are divided into interruptible and non-interruptible appliances. A further distinction of appliances can be done following the works of Chen et al. [13] or Shewale et al. [5] where appliances are divided in three categories based on their ability to be shifted in time and/or interrupted during their task completion. In this work, the latter classification is

followed and the appliances are classified as *non-shiftable*, *shiftable non-interruptible* and *shiftable interruptible*. How each appliance is classified is based on the consumer usage of said device and the "smartness" of it: a dishwasher, based on the year of production, can be interrupted during its task without impacting the overall job or not.

Non Shiftable Appliances that cannot be shifted to other time slots other than the prefixed ones. In this category belong both the appliances that have to stay ON no matter what (i.e. the refrigerator) and the appliances whose job cannot be either programmer or stopped by the Smart management System (i.e. the Television, the Fan, the Oven).

Shiftable and Non Interruptible These are the appliances whose starting time can be moved within the available time slots but once it starts, it cannot be stopped. Such appliances are the washing machine or the water heater

Shiftable and Interruptible Flexible in their usage and can be both shifted in the available time horizon and interrupted during their work and resumed anytime. Dishwashers, vacuum cleaners are part of these category. If batteries or plug-in electric vehicles are connected to the smart home energy management system, those devices fit in these category since their charging/discharging cycles can be started and interrupted anytime as long as there is enough power available when the car is needed, for example.

Chapter 3

Optimization Strategies for Smart Home Appliance Scheduling Problem

The main goal of a scheduling algorithm is to find the best possible starting times of a series of jobs so as to maximize profit or minimize losses. There are various types of scheduling problems such as job scheduling [14], flow shop scheduling [15] and power scheduling problems [16].

A sizable increase in electricity demand can be seen in the emerging smart grid setting, due to the addition of more and more devices in our daily life. To reduce the gap between demand and supply in the residential sector that are many approaches but the most effective is to be more efficient with the utilization of energy sources.

The bilateral flow of information in a smart grid helps both users and utility provider in understanding when to reduce the power demand during peak periods of time, reducing the cost of the power generation. This is useful for both parties, the consumer, that can lower the costs on the electricity bill and the provider that can soften the energy production, minimizing the resource usage for the same result.

Home Energy Management Systems faces the difficult task of managing multiple appliances and their task, allowing them to be completed, minimizing the overall

cost, without compromising user comfort. This is the aforementioned power scheduling. The appliances, as mentioned in Section 2.1.4, have different characteristics, which adds a layer of complexity to the problem: some appliances can be scheduled while others are subject to user demand or have to be always on, which leaves them outside of the HEMS control.

The Smart Home Appliance scheduling Problem (SHASP) formulation can aim at minimizing/maximizing different objective function. Most notably:

- **Minimize electricity cost (EC)**
- Minimize peak-to-average ratio (PAR)
- Maximize user satisfaction (US)

It can solve one or more of these at the same time. The thesis here presented revolves around the minimization of the electricity costs by shifting the controllable appliances to off-peak hours, while also considering the amount of outages that the user might experience

3.1 Description of the problem

The SHASP (or Smart Home Appliances Scheduling Problem), can be described as follows:

Let A be a set of n independent appliances to be scheduled. The time horizon chosen for the study is 24 hours and it is divided into time slots. The resulting time horizon H is defined as $H = 1, \ldots, T$ with T being the deadline for all the appliances in set A, that varies depending on the chosen sampling of the day's time slots.

Each appliance $a_i \in A$ has a processing time d_i. Preemption is not allowed, meaning that once it is switched on, the job must be completed before the day ends, without interruption. Each appliance consumes power based on the power rating p_i assigned my the manufacturer, measured in kWh.

Assuming that the home is connected to the power grid and has stipulated a contract with the utility provider, a maximum energy consumption P is allowed for each time slot of the time horizon. This is used as a threshold for the energy

consumption of the appliances, over which, power outages happen. The latter is an undesirable occurrence that lowers the user satisfaction.

The utility provider, with the aforementioned agreement, defines a unit energy cost for the day, denoted by c_t, which specifies the cost of the electricity in each time slot. There can be different tariffs, based on the contract and it usually follows a Time-Of-Use Pricing scheme, where the cost per kWh increases during the peak hours so has to incentivize the consumer to shift their consumption behaviours to *off-peak* hours.

Additionally, for each time slot $t \in H$, an amount S_t of solar energy is produced and can be used as an aid to the need of energy during peak hours. The addition of solar energy increases the allowed peak energy consumption at each time slot from P to $P + S_t$.

The SHASP is solved once all the appliances requested are scheduled within the selected time horizon, minimizing the overall cost and minimizing the number of power outages.

3.2 Mixed-Integer Linear Programming

A possible mathematical programming formulation of the problem is presented by Della Croce et al. [17]. The main decision variable is $x_{i,t}$, for each $a_i \in A$ and $t \in H$, such that:

$$x_{i,t} = \begin{cases} 1 & \text{if the appliance } a_i \text{ starts at time t} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

In it's paper, Della Croce et al. [17] pre–calculates the overall cost for each appliance starting in every possible time–slot of the time horizon. This results in a cumulative coefficient $C_{i,t}$ which represents the total cost of time t the $i-th$ appliance, considering the unit cost c_t and the consumption of each appliance given by $p_{i,\tau}$ with $\tau \in \{1, \ldots, d_i\}$.

$$C_{i,t} = \sum_{\tau=1}^{d_i} p_{i,\tau} \tag{3.2}$$

The complete Mixed Integer Linear Programming formulation is as follows:

$$\min \quad f(x_{i_t}, z_t) = \sum_{a_i \in A} \sum_{t \in H} C_{i,t} \cdot x_{i,t} - \sum_{t \in H} c_t \cdot z_t \tag{3.3}$$

$$\text{s.t.} \quad \sum_{t \in H | t \leq T - d_i + 1} x_{i,t} = 1 \qquad \forall a_i \in A \tag{3.4}$$

$$\sum_{a_i \in A} \sum_{\tau=1}^{d_i} p_i \cdot x_{i,t-\tau+1} - P - S_t \leq 0 \qquad \forall t \in H \tag{3.5}$$

$$z_t \leq S_t \qquad \forall t \in H \tag{3.6}$$

$$z_t \leq \sum_{a_i \in A} \sum_{\tau=1}^{d_i} p_i \cdot x_{i,t-\tau+1} \qquad \forall t \in H \tag{3.7}$$

$$x_{i,t} \in \{0, 1\} \qquad \forall a_i \in A \quad \forall t \in H \tag{3.8}$$

$$z_t \geq 0 \qquad \forall t \in H \tag{3.9}$$

$$\tag{3.10}$$

The objective function is defined in 3.3 and it aims at the minimization of the overall cost of scheduling all the appliances, also taking into account the solar panel generation.

Constraints 3.4 limits to one the possible starting times of each appliance, meaning that same appliance cannot be run multiple times. Constraint 3.5 refers to the energy availability of the smart home. The power consumed by the scheduled appliances at each time slot must be lower than the power drawn from the grid, with the help of the power generated by the solar panel. Constraint 3.6 limits the solar panel energy usage to the actual generation and 3.7 limits it to the power consumed in the home, so as to not waste any more energy than the needed one. Lastly, constraints 3.8 and 3.9 define the domain of the decision variables.

3.3 Local Search Heuristics

A local search algorithm can be seen as a iterative search procedure that improves a initial solution via a series of little modifications until it reaches a local optimum. At each iteration, all the possible modifications (or "moves") to the current solution are looked at and the one that improves more the current objective function is taken as the new solution.

An important issue with the local search method is that, the solution obtained is the best with respect to its neighborhood: this means that said solution is not

necessarily the global optimum of the problem and in most of the cases it is a mediocre solution at best. How good the solution ends up being is a matter of the richness of the move set considered at each iteration of the method. Of course, the greater the number of solutions considered in the neighborhood, the higher the accuracy of the solution but also higher the computing times.

Local Search heuristics, called also *Meta-heuristics* (coined by Glover in 1986), come in many shapes and forms. A first example of this is Simulated Annealing (SA), described for the first time in 1983 by Kirkpatrick et al. [18]. SA is based on an analogy with metallurgy science methodologies where annealing part is interpreted as a slow decrease in the probability of accepting solutions that are worse as the search dives into the search space. This allows for a more extensive exploration. From there, other approaches where formulated starting from natural phenomena, such as Tabu Search, Ant Systems or even Bee Colonies behaviour. The preferred meta-heuristic for this thesis is the Tabu Search. The following section details an introduction to its workings and how TS has been adapted to solve the SHASP.

3.3.1 Tabu Search

Tabu Search is a Meta-Heuristic developed by Glover [19] and Glover [20]. This algorithm tries to overcome the problem of local optima of traditional Local Search methods by allowing non–improving moves whenever a local optimum is found. It memorizes solutions that have been already found by the algorithm in a list (so called *tabu list*) for a short period of time (*tenure*). Whenever no improving solutions can be found, the algorithm will cycle back to a previously visited solution using the aforementioned memory and continue to another path so as to find a better solution.

Search Space and Neighborhood Structure

Of critical importance in this type of meta heuristic are both the *search space* and the *neighborhood structure*.

Adapting the Tabu Search to the Appliance Scheduling problem can be challenging due to the size of the search space. The search space is defined as the space of solutions that can be visited by the Meta-heuristic. For instance, let us take in consideration a slightly different formulation of the SHASP problem. The decision variables $x_i \in [0, T)$ where T is the time horizon that the problem focuses on, for

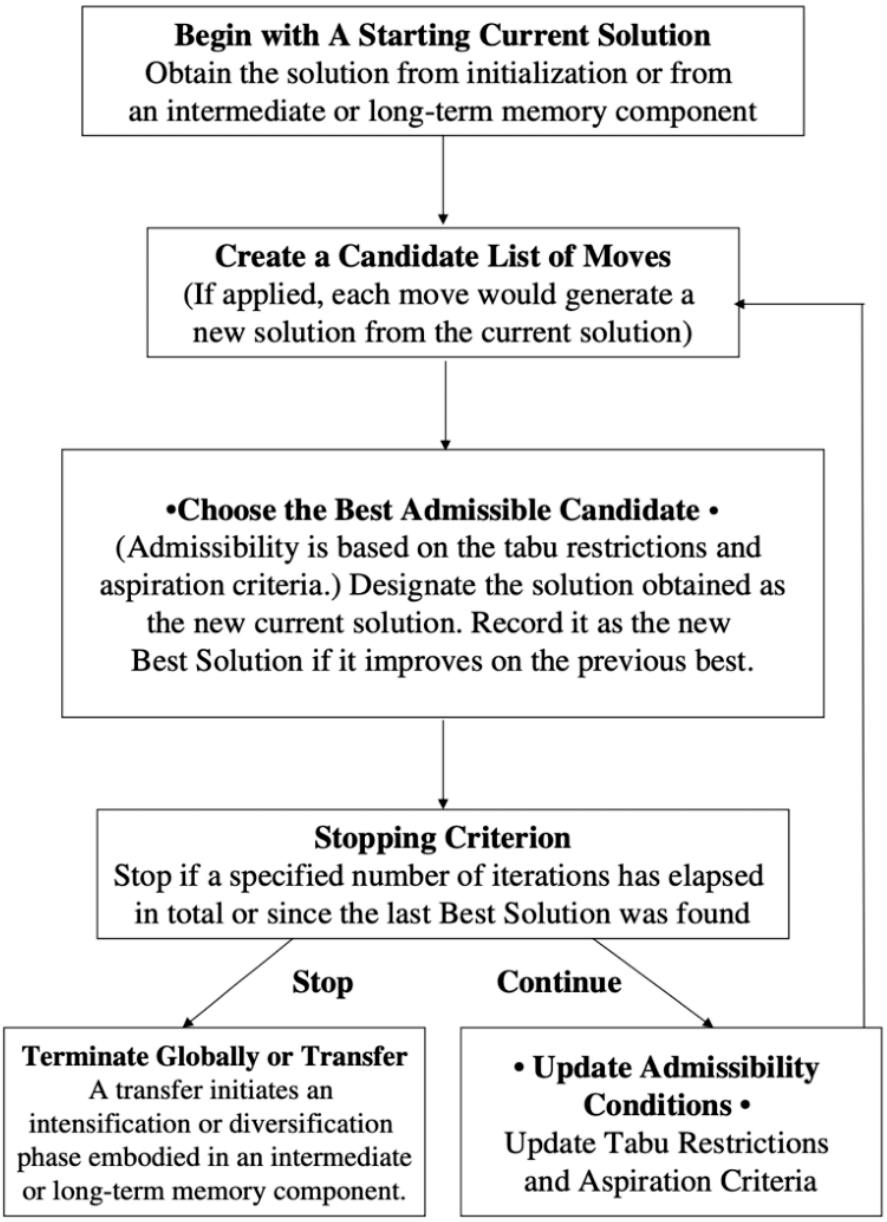

Figure 3.1: Tabu search memory components. Glover [19]

every $a_i \in A$ with A being the set of shiftable and non interruptible appliances. This leaves us with a solution set $x = [x, x_1, \ldots, x_n]$ with n number of appliances in set A. The resulting **search space** would be of size T^n.

The **feasible** search space is a subset of the search space where all the feasible solution are. A solution is feasible if it abides to some strict conditions i.e. the appliances must start in an interval between 0 and $T - d_i$ where d_i is the duration of time slots that the $i - th$ appliance takes to complete its job.

The way in which the algorithm looks for the solution is by comparing its current solution, denoted S, to its neighbors. In order to do that, a neighborhood structure must be defined. The *neighborhood*, denoted $N(S)$ (neighborhood of S), is defined as a subset of the search space which contains the *solutions obtained by applying a single local transformation to S*.

Usually, the neighborhood structure is problem specific so there is not a "one fits all" definition. The type of structure the neighborhood gets depends on the computational time constraints that the modeller might have, to the search space size or shape.

For the problem under exam different types of neighborhood structures have been considered. Assuming as a local move a shift of one slot ahead or behind of one appliance or more appliances (2^n possible combinations), the research of the best solution in the neighborhood becomes exponentially difficult, and it defeats the purpose of the heuristic of being faster then the MILP.

Another approach was then to consider only a fraction of the whole neighborhood of solutions, namely, a random sample of $\sqrt{N(S)}$. This allowed the algorithm to search the neighborhood while still having diversification. This is at the expanse of a more thorough research. This was the preferred approach since it could keep the computational times of the heuristic lower than its exact counterpart while still finding optimal solutions (results of this analysis are shown in 4).

Tabus

What separates a Tabu Search from any other Local Search is the possibility of making solutions "taboo". Their main aim is to prevent the method to cycle back to a local optima when exploring the search space, looking for a different solution that do not necessarily improve the objective functions value. When this phenomenon occurs, something must tell the algorithm to avoid going back to recently visited solutions and instead, explore different solutions that might not be the best right away but lead to better solutions globally. This is the key idea behind this method: recently visited solutions are declared *tabu* (prohibited or restricted). An example might be, for 3 appliances in our Smart Home, if solution $S_0 = [10, 1200, 100]$ has

been visited, the next step when a solution S_1 is chosen to be the most improving solution, if S_0 is present in the full neighborhood of solution S_1, that is not taken into consideration as one of the possible solutions, since going back to S_0 has become *tabu*.

For how long a solution cannot be taken in consideration is given by a hyper parameter called *tabu tenure*. The impact of the tenure value on the proposed method has been analyzed through a grid search over different instances of the problem with different settings. It is shown that the tabu length itself doesn't impact that much the computational time of the solution.

Tabu solutions are stored in what is called a *short-term memory* or, usually, the *tabu list*. Different information can be stored in the list in order to keep the memory expanse light. In our case, what is stored in the memory is the complete solutions that have been considered beforehand.

Cost Function

An important aspect of how the search is conducted is based on the choice of the Objective Function and the resulting values that each solution has. Being the minimization of costs the main objective of the formulated problem, it seems reasonable to use as fitness value of the candidate solutions, the actual cost that would be paid if said solutions were to be used in a real setting.

$$C = \sum_{a_i \in A} \sum_{t \in T} U(x_i, t) * p_i * c_t \tag{3.11}$$

where: c_t is the electricity consumption at time t given by the utility provider, measured in €, p_i is the power rating of the $i - th$ appliance in the appliance set and $U(x_i, t)$ is given by:

$$U(x_i, t) = \begin{cases} 1, & \text{if } t \in [x_i, x_i + d_i] \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

To this cost is then added a penalty term, in order to take into account of power outages. An outage occurs when the power needed to run the appliances in a specific time slot t is higher then the energy availability of the house. The energy availability is given by the power given by the utility provider plus the power generated by additional renewable energy sources. For every outage that

occurs, a penalty is added to the cost so as to take into account it and reduce it as much as possible.

The resulting Objective Function that the tabu search uses to find the solution is:

$$\text{Objective Function} = C + n_{bo}$$

where n_{bo} is the number of outages that happened in the whole time horizon and C is the electricity cost consumption calculated in equation 3.3.1

Finally, a general basic approach to the tabu search algorithm is given in Figure 3.1

Termination Criteria

Being the Tabu Search a iterative process, the search itself could go on for as long as we like, even forever. It is important to define some stopping condition ourselves in order to avoid an infinite loop. The most commonly used stopping conditions are:

- a maximum number of iterations (or maximum time) is reached
- objective value reached a fixed threshold, usually a percentage of the exact solution
- if after a number of iteration there is no improvement in the value of objective function

Diversification

As explained before, one of the main issues of the Tabu Search, along with the Local Search methods, is the myopic evaluation of the search space. Most of the time, if not tweaked correctly, the method will be stuck in a small portion of the entire search space. This is an issue when the dimentionality of the problem (the number of appliances considered within the residential setting) becomes very high. In order to avoid missing on better solutions, *diversification* is employed. It is an algorithmic mechanism that forces the search in unexplored areas of the search space.

In our specific case, the diversification technique used is called *restart diversification* where, when the algorithm gets stuck for a fixed amount of iteration on a solution, it will move to a randomly generated solution within the search space. This is also used as a stopping criteria: when 10 restarts have been done, the algorithm stops as it has looked enough around the search space to find a solution that should satisfy the criteria.

How many steps the methods stays on a solution before restarting and how many restarts to do before terminating the search have been found empirically.

3.4 Reinforcement Learning

Another approach that can be taken to solve this particular scheduling problem is knows as reinforcement learning. In this kind of approach, the method learns how to behave by interacting with external stimulus and it can be very powerful in solving scheduling problems.

Reinforcement learning (RL) is one of the three main machine learning paradigms alongside *supervised learning* and *unsupervised learning*. It approaches the learning problem by looking at how people or animals learn and evolve: we seek a goal and make choices to reach it while the world around changes based on our choices and more.

Much like in nature, an *agent*, main actor in the RL algorithm, will discover its best behaviour though a series of **trial and error**, tuning its policy so that it will reach its final goal. How does the agent know which is the best actions to take depends on the *reward signal* that the environment around it will generate at each step in its learning process. This is done by keeping in mind that most of the time, the action that will yield the maximum reward in the long run, isn't always the one that is best right now: eating vegetables might be worse than eating sweets in the moment, but your health will improve day by day.

We'll explore the theoretical elements of the methodology and how it can be applied to the problem at hand. The next section is heavily based on the works of Silver [21] and Sutton and Barto [22].

3.4.1 Theoretical Background

As anticipated, the main goal of an RL algorithm is to learn how to maximize the delayed reward by balancing how the the *agent* interacts with the *environment*.

Agent and Environment

The *agent* is the element that is capable of making decision and is the one that interacts with the environment. It observes the state of its surroundings and acts differently based on it. The decisions that an agent makes are defined as *actions* (a_t). An important distinction is that the agent itself has no control on the environment but through its actions it can influence it in different ways based on the type of problem. What actions an agent can make during its journey are based on a set of actions called *actions space*. The action space depends on the type of agent and how the modeller want the agent to behave with respect to the environment. It can be *discrete*, where only a finite number of moves are possible (e.g. $\mathbf{A} = \{0,1\}$ as the motor of a car can be either on or off) or *continuous* where the actions take on real values (e.g. $\mathbf{A} = [0,130]$ as in the speed in km/h of a car). While this seems like a trivial concept, in reality it is fundamental as different algorithms fit different types of actions spaces and there is not a one-fits-all approach.

The *environment* interface represents what the agents interacts with. Whenever the agents does something, the environment will react to it, changing its state and emitting a *reward* and an *observation* that the agent will then analyse and base its next action on.

State and Observations

The *observation* is a representation of the environment *state* (S_t). The observation helps the agent understand how its previous action affected the environment and what should be its next action in order to achieve its goal. The observation, which is what the agent sees, might not always be the same as the environment state. In fact an environment can be either *partially observable* or *fully observable*. If the environment is partially observable, only a fraction of the information is at the agent disposal whereas if the environment is fully observable, the information that the environment state carries is for complete use by the agent. Going forward, we'll focus on fully observable environments as they are the most commonly studies in literature and well suited for the problem under study.

Rewards

Whenever an action is done, the environment will respond with a *reward signal* or simply, *reward* r_t. It is a scalar feedback that allows the agent to understand the value of being in a particular state is. It is critical to the learning process as it allows the agent to distinguish between actions that are to encourage and actions that are to ignore. This feedback is given at each time step t, which makes it local. The agent's goal is to maximize the total reward obtained during its "life". This means that what is crucial is not the immediate reward but the cumulative reward in the long run. This idea is formalized by Sutton and Barto [22] as the *reward hypothesis*:

> All of what we mean by goals and purposes can be well though of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward)

An important result of this type of learning is that the agent will learn to act in order to maximize the reward. So it is important to create a reward function that represents what the agent has to accomplish. The reward signal is a way of communicating to the agent what we want to achieve and not how we want it to be achieved.

What ends up happening is that the agent does an action, waits for a response by the environment and then takes another action based on the previous information obtained. This is repeated cyclically, building a *trajectory*, a sequence of states, rewards and actions:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots \tag{3.13}$$

Delayed Return

Given the reward definition, the *expected cumulative return* is defined as the sum of the rewards. We define as G_t the sum of the rewards starting from time–step t:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T \tag{3.14}$$

with T being the final time step. This formulation makes sense if there exist a terminal state for each of the trials done by the algorithm. In this case, these "trials" are called *episodes*. There exist also cases where the agent-environment

cycle can never be stopped and can go on forever. We call these *continuing tasks*. In these cases, the return cannot be computed directly as it would be infinite, but each time–step needs to be *discounted*.

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad \gamma \in [0,1) \tag{3.15}$$

where γ is a parameter called the *discount rate*. Of course, the value of γ influences how much the agent values rewards further away from its time–step:

- $y = 0$: the agent is "myopic", meaning that it is only interested in maximizing the immediate rewards (R_{t+1});
- $y \longrightarrow 1$, the agent becomes more farsighted and weight the future rewards more;

Markov Decision Process

In order to make use of Reinforcement Learning techniques, a problem must be formalized as a *Markov decision Process* (MDP), a classical formalization of sequential decision making. A decision process is defined as an MDP if the agent, to make a decision at time t only needs the information carried by the state S_t. This means that the states have the memoryless property, meaning that the information that they carry is always sufficient to understand the environment history and there is not need to look at the states before:

$$\mathbb{P}[s_{t+1}|s_t] = \mathbb{P}[s_{t+1}|s_1, \ldots, s_t] \tag{3.16}$$

A visual representation of how a Markov Decision Process behaves can be seen in 3.2. The MDP is defined as a tuple of four elements:

$$< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} >$$

where:

$\mathcal{S}$ is the state space;

$\mathcal{A}$ is the action space;

$\mathcal{P}$ is the transition probability matrix where each element is:

$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a) \tag{3.17}$$

it being the probability of going from state s at time t to state s' at time $t+1$ after having taken action a;

$\mathcal{R}$ is the reward function, defined, for a state s and an action a, as:

$$R(s,a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a] \tag{3.18}$$

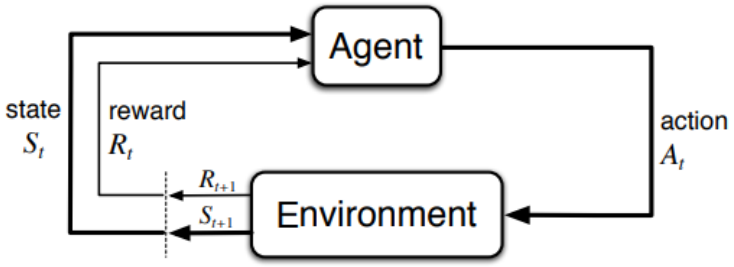

Figure 3.2: Agent-Environment Interaction. The reward and the observation of the environment, triggered by the action done by the agent, are then fed beck to the agent in order to take another action at the following iteration. Sutton and Barto [22]

Polices and Value Functions

All RL algorithms involve the estimation of a *value function* which is a function of the state (or pairs of state and actions) that allows the algorithm to know the "goodness" of a specific state (or the goodness of performing a specific action while being in a specific state). This "goodness" is referred to the return that is expected in the future, starting form that specific state. The way that the agent behaves when presented with a state is called *policy*, denoted by π which is a mapping from states to probabilities of selecting an action from a set of feasible choices ($\pi(a_t|s_t) = \mathbb{P}[a_t|s_t]$).

The main goal of Reinforcement Learning is to learn the optimal policy $\pi*$. The optimal policy is defined as a policy that leads the agent into taking action that will yield the maximum return in the long run. To learn the best policy, RL algorithms use the aforementioned value functions:

- A *State Value Function* $v_\pi(s)$ of a state s under a policy π is the expected return starting from s while following π:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s\Big] \quad \forall s \in \mathcal{S} \tag{3.19}$$

- Similarly, the Action-Value Function $q_\pi(s, a)$ of a state s and action a under a policy π is the expected return after taking action a, starting from s while following π:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|S_t = s, A_t = a\Big] \quad \forall s \in \mathcal{S} \tag{3.20}$$

Both equations 3.4.1 and 3.4.1 satisfy recursive relationship between the value at a given state and it's successor states. In fact, they can be rewritten, respectively as:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} P(s', r|s, a)\Big[r + \gamma\mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']\Big] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big], \quad \forall s \in \mathcal{S} \end{aligned} \tag{3.21}$$

The result is what is called in literature *Bellman Equations* [23]. The value function v_π is the unique solution to its Bellman equation.

The same problem can be approached with different policies that will yield different value functions. The best policy $\pi+$ among all of the possible choices is the one that is better than or equal to any other policy π' meaning that its expected return is greater than or equal to that of the other π' policies for all states. That is defined as the *optimal policy*. The resulting *optimal state-value function*, denoted by $v*$ is the value function that follows the optimal policy:

$$v*(s) = \max_\pi v_\pi(s) \tag{3.22}$$

Same for the *optimal action-value function* denoted $q*$, defined as:

$$q*(s, a) = \max_\pi q_\pi(s, a) \tag{3.23}$$

The Bellman equation for $v*$, also called *Bellman optimally equation* expresses that the value of a state under an optimal policy must equal the expected return for the best action from that state

$$v*(s) = \max_{a} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} P(s', r|s, a)[r + \gamma v*(s')] \tag{3.24}$$

and

$$q*(s, a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} P(s', r|s, a)[r + \gamma \max_{a'} q*(s', a')] \tag{3.25}$$

3.4.2 Adapting Reinforcement learning to SHASP

Since the Smart Home Appliance Scheduling Problem can be formulated into a decision making problem under uncertainty , Reinforcement Learning becomes a suitable solution for the problem. RL does not require explicit probabilistic models for either the Renewable Energy Sources or the controllable loads. It can learn from real data being sampled from actual PV panels in a home.

From the MILP formulation described in section 3.2 it is possible to derive a Reinforcement Learning formulation of the problem by adjusting its elements to fit the Markov Decision Process framework.

Markov Decision Process Formalization

Following the definition of a Markov Decision Process given in section 3.4.1, we investigated on how to transpose the original mathematical problem into the reinforcement learning formulation. This is not a trivial as adapting the constraints to RL turned out to be quite a challenge. In the following paragraphs, we'll go in a deeper look at the design process of the MDP, how the decisions made come out to be and the problem faced in the modelling.

State Space

The state space is the set of all the observations that the reinforcement learning agent sees during its interaction with the environment. Looking at the original formulation of the problem, the main state that the agent should be concerned

about is the actual energy consumption, meaning the energy consumed at each time step, taking into account both the power drawn from the grid and the renewable energy source. The energy availability at disposal of the agent alone is not enough to describe the Smart Home Environment. This is because, in order to turn on or off an appliance, the agents needs to know which tasks are currently running, which have been already completed and which still need to be started. The first approach to the issue was to make he state a four-tuple of values The state at time step t was described as $S_t = \{A_t, E_t, T_t, \bar{D}_t\}$, in which A_t is the active set of appliance jobs that are running at time step t, E_t is the energy availability which is given by the sum of the electricity consumption done by each of the active appliances during a time slot minus the maximum energy availability at disposal of the system at time t (Ψ).

$$E_t = \Psi - \sum_{a_i \in A} a_i^t * e_i \tag{3.26}$$

T_t is the completed set of jobs. $\bar{D}_t$ is the vector of activity down counter of the active appliances: Initialized at D_t where the d_i element is the running time of the $i-th$ appliance, once an appliance is activated, the respective element in vector $\bar{D}_t$ is decremented by one until it reaches 0. At that point the task is completed and T_t is updated.

While this kind of approach seems reasonable, it quickly became clear that a detailed description of all the elements is unfeasible due to the *curse of dimensionality*: E_t is continuous whereas E_t, T_t and $\bar{D}_t$ possible values grow exponentially with the increase of the number of appliances (2^N with N the number of appliances considered).

This posed the problem of favouring a computationally heavy approach with respect to a more "relaxed" one, with less elements in the state space. The latter option would come at the cost of losing the Markov properties of the decision process, since there are, in literature, studies centered around making Reinforcement Learning work even in non Markovian environments, showing that it is possible to work around this kind of issues [24, 25, 26]. The latter option was the preferred one since one of the main performance indicators for the goodness of a method instead of another is the speed of execution.

We ended up using only the energy availability E_t as state. This choice still carries the issues of having to work with a continuous state space, but with reduced complexity, due to the absence of the other three elements

Action Space

The action space is the set of all the actions that our HEMS can do. In this case the choice was straightforward and directly derived from the MILP formulation of the SHASP problem. Let $u_t(S_t)$ denote the set of activities modelled as an array of N binary values such that the i-th element of said array is 1 if the appliance has to be be switched on.

$$u_t = [u_1, u_2, \ldots, u_n] \tag{3.27}$$

This means that at each time step, the system looks at all the possible actions (2^N possible actions) and removes from them the schedulable appliances. The $i-th$ appliance has to be removed from the set of possible choices if:

- is already active ($a_i = 1$)
- is not active but, if activated, would bring $E_t < 0$
- has already finished its job ($\bar{t_i} = 1$)

Reward

The reward function is a crucial part of the formalization of the MDP as it describes what the agent should achieve. In our case, the main goal is the cost minimization of the energy consumption while still being in the energy boundaries set by the plan agreed with the utility provider. The first main component of the reward function is the actual cost of the electricity used during the day. This can be directly transposed from the MILP Objective Function with the expectation that, being RL all about maximizing the expected return of the episodes, the reward function has to be made negative.

$$R_t = -C_t \tag{3.28}$$

where C^t is sum of the costs (c_i) related to the energy consumption (e_i) of each active appliance (a_i) at time t.

$$C_t = \sum a_i^t * e_i * c_i \tag{3.29}$$

Since this is now a maximization problem but with a negative cost function, if no penalty is added to the reward, the natural course of action that the agent

would take is to keep all the appliances turned off for the entirety of the time horizon. This of course minimizes the energy cost but defeats the purpose of the scheduling problem: no jobs are scheduled, ever.

Many penalty have been tried in order to help the agent learn the best possible policy.

- Positive reward for completing a task (+100)
- Negative reward for failing a task due to power outage (-1e6)
- Negative reward if, at the and of the day, some tasks were not completed at all (-1e6)

The main issue with this kind of approach is that the reward function becomes non-Markov: the reward at time t, i.e. when the task of appliance i is completed, depends on an action done d_i time steps before and all the states in between. As stated before, even if there are in literature some works that tackle the problem of sparse reward in non-Markov environments, the results obtained with this kind of design choices where not satisfactory. For this reason, we decided to set aside the Reinforcement Learning methodology but, since there is potential in this kind of approach to the problem under exam, further studies could be done to design an adequate MDP.

Chapter 4

Results

4.1 Case Study

In this section, we are going to look at the numerical simulation done to test the proposed Tabu Search heuristic. Table 4.1 summarizes all the parameters and the characteristics of the appliances that can be controlled through a home management system in a smart household. The simulation where done for a 24 hour time horizon with a 1 minute scheduling resolution. One major assumption done for the computation was on the predicted PV generation energy shown in picture 4.1. Since the generation of solar energy is difficult to model and forecast due to the high variability of the energy source, in this thesis we considered it ideal, peaking at 1kW at mid day, where the solar irradiance is strongest. Figure 4.1 shows how the data in question is shaped.

Utility providers usually allow the consumers to choose a different plan for their consumption so as to adapt their consumption patterns in order to reduce their costs. The electric rate schedule can change depending on the Demand Response method used. In this thesis we analyse an household that makes use of Time-of-Use (TOU) Pricings. The type of tariff changes based on day of the week and consumption patterns but there can be different tiers:

One Tier Price (1TP) only one price is applied to the entire days' consumption;

Two Tier Price (2TP) two prices are applied to the day, a higher cost for *on-peak* periods, where load is usually heavier on the grid, and a lower cost during *off-peak* periods;

Table 4.1: Power ratings of different appliances of a single user with a length of operational time

Appliances	**Duration**	**Power Rating [kWh]**
Washing Machine	2 h	2
Dish Washer	2 h	1.2
Vacuum Cleaner	1 h	1.2
Cloth Dryer	1 h	1.8
Water Heater	2 h	3.5
Hair Dryer	1 h	2
Fan	4 h	1
Iron	2 h	0.25
Humidifier	4 h	0.15
Oven	45 min	2
Rice Cooker	45 min	0.8
Air Conditioning	2 h	3.5
Electric Car	4 h	3.5

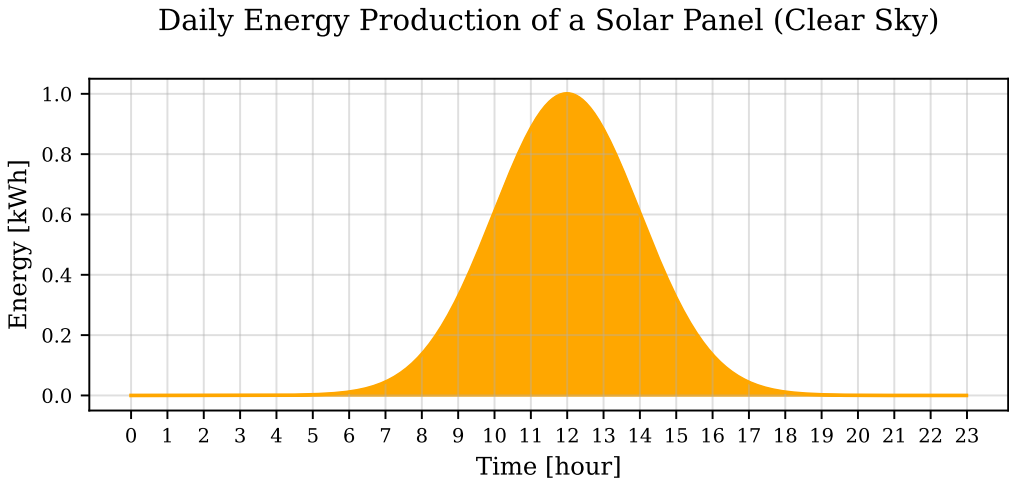

Figure 4.1: Ideal Power Generation of Solar Panel with 1 kWh at peak energy production

Three Tier Price (3TP) three price tiers are used. The day is divided into *on-peak*, *mid-peak* and *off-peak* periods.

Figures 4.2a, 4.2b and 4.2c, show different Time-Of-Use Pricing Schemes over the whole time horizon:

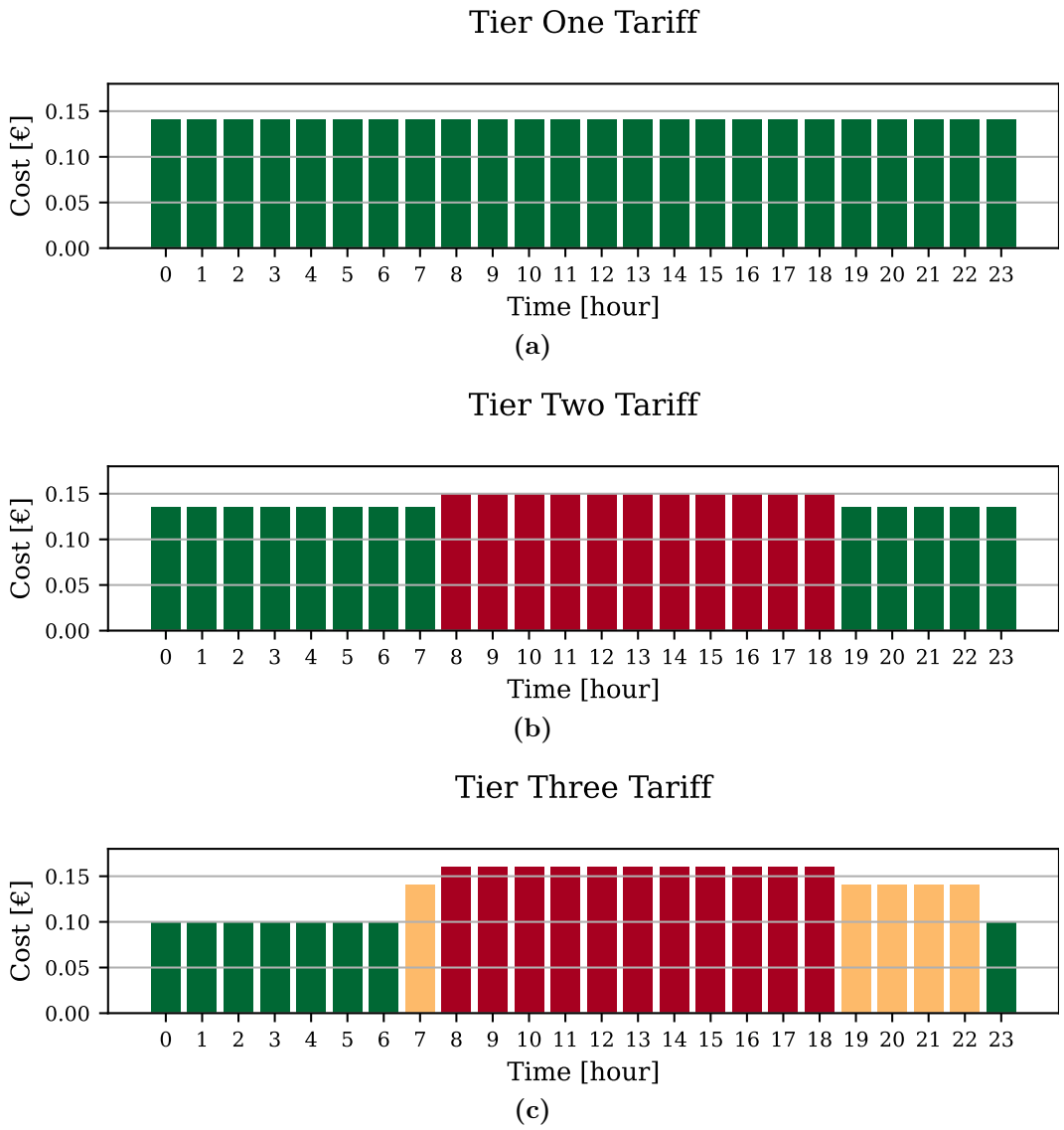

Figure 4.2: Time-of-Use pricing scheme with one (4.2a), two (4.2b) and three (4.2c) tier price

4.2 Scenarios

Here we present the scenarios that were considered for the simulation and comparison of the two methodologies. The first one considers the Home Energy Management System with only the power grid that supports the power demand. The grid is assumed to inject 3kWh at all times to the house, without interruption due to

failure of the power grid.

Second Scenario instead supposes the usage of Renewable Energy Sources in the form of a PV panel with ideal generation of power: weather conditions are ideal, with maximum possible solar irradiance and clear sky.

4.2.1 Scenario 1: HEMS with Only Grid

Table 4.2 compares the results between the solutions obtained by solving the Integer Linear Program of the SHASP and by the proposed Tabu Search heuristic. As for the Price Tiers considered, we look to the tier 2 and 3, mostly because we are interested in understanding how well the proposed heuristic shifts the different appliances in order to reduce the costs. A graphical representation of the data obtained can be seen in figure 4.3 that shows how the costs and the computational time varies with respect to the number of appliances that are considered in the environment. For the sake of simplicity, the appliances where took as in the order shown in table 4.1.

The costs between TS and ILP are consistent with each other but diverge when the appliances become more than three due to the presence of the Water Heater that has a power rating of 3.5 kWh. The presence of this appliance alone makes the ILP Problem infeasible since there is a fixed constraint on how much power can be drawn from the grid ($3kWh$) alone. As in a heuristic there are no hard constraints, but those are relaxed to a penalty term in the objective function, the higher cost for the TS algorithm is due to the actual increase in costs. It still finds a solution but said solution is heavily penalized: it cannot be scheduled in a meaningful way to reduce the costs.

As far as the computational costs go, the two algorithms are close for the first computations (1 and 3 appliances) but then the heuristic becomes slower. This is due to the more iteration that the TS does while looking for a better solution, whereas the solver stops before, after computing that there is no good solution based on the constraints applied.

A visual representation on how the appliances are scheduled with the two different methods can be seen in figure 4.4.

In both cases (3PT and 2PT) the appliances are scheduled in the off peak periods, avoiding any additional costs in the on-peak periods of the day. The ILP gets more “risky” by starting an appliance while another one is already running

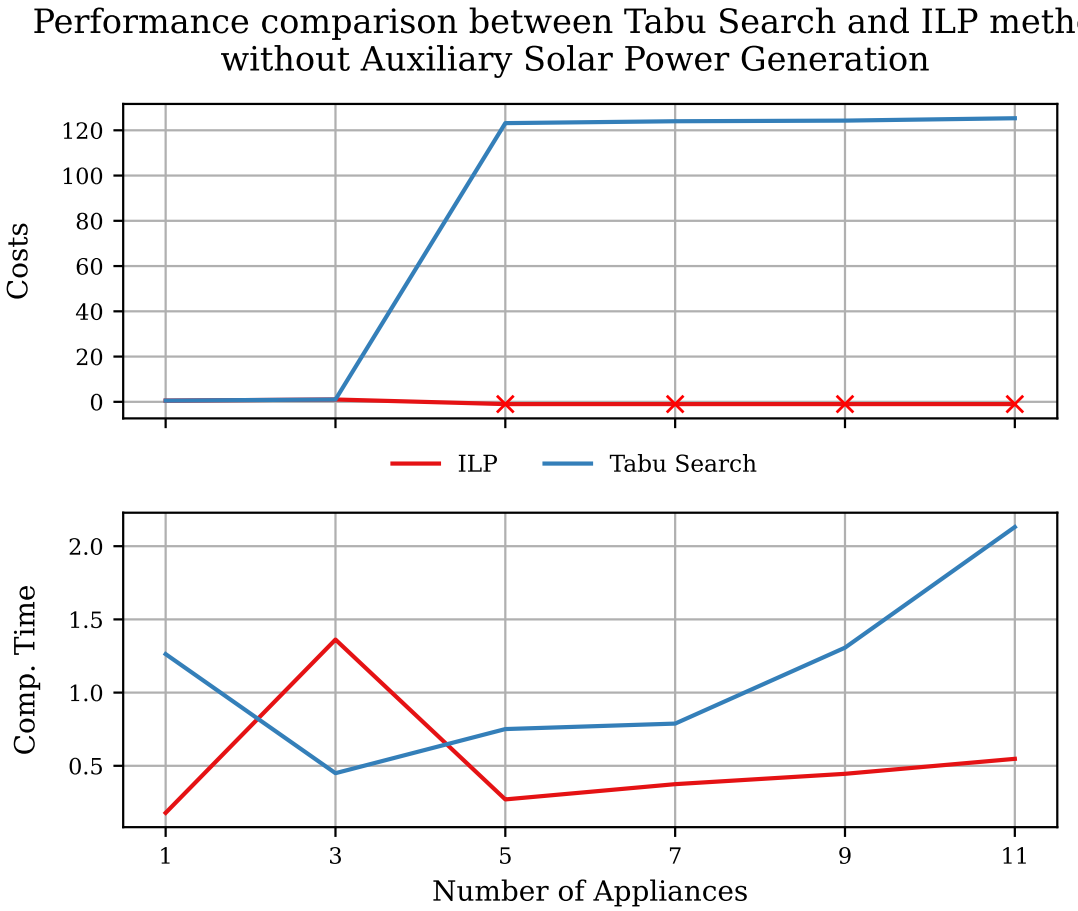

Figure 4.3: Numerical solutions for the Smart Home Appliance Scheduling Problem. Points marked with a red "x" are instances where outages occurred

(appliance 2 is scheduled at time 21:35 while appliance 3 was still running), whereas the Tabu Seach spreads them evenly. A non trivial result of our algorithm can be found in figure 4.5. Here is described the scheduling of 5 appliances with respect to a Tier 2 Tariff. Of course, being the instance infeasible for the ILP formulation given, no solution can be considered valid and no scheduling is performed. What the proposed heuristic does instead is to schedule the appliances at the end of the day, where the costs are lower without considering the burden on the power grid generated by accumulating all the power consumption in one time slot. This might be due to the design of the penalty term: the additional cost is the same whether the outages are spread along the time horizon or stacked in one.

Table 4.2: Result table of the computation between MILP method and Tabu Search without Panel Generation. If an ILP solution yielded a -1 Cost, that means the instance considered made the problem infeasible and no solution was found

N. Appliances	Price Tier	Algorithm	Cost	Comp. Time
1	2	ILP	0.54	0.17854
		Tabu Search	0.54	1.26293
	3	ILP	0.4	0.13583
		Tabu Search	0.4	1.06445
3	2	ILP	1.03	1.36222
		Tabu Search	1.03	0.44957
	3	ILP	0.76	1.38187
		Tabu Search	0.8	0.40936
5	2	ILP	-1.0	0.27043
		Tabu Search	123.17	0.75092
	3	ILP	-1.0	0.26852
		Tabu Search	122.45	0.67452
7	2	ILP	-1.0	0.37462
		Tabu Search	124.0	0.78831
	3	ILP	-1.0	0.4376
		Tabu Search	123.22	0.95903
9	2	ILP	-1.0	0.44527
		Tabu Search	124.28	1.30686
	3	ILP	-1.0	0.38552
		Tabu Search	123.31	1.29334
11	2	ILP	-1.0	0.54762
		Tabu Search	125.33	2.13101
	3	ILP	-1.0	0.54078
		Tabu Search	124.23	2.181

4.2.2 Scenario 2: HEMS with PV Panel

Table 4.5 shows instead the same appliances scheduled with the usage renewable energy sources, in our case a solar panel with 1 kW of peak power generation. Now the ILP method always finds a feasible solution for the selected appliances since the energy produces by the solar panel is enough to satisfy the required power

Table 4.3: Numerical results of both algorithm for 3 appliances in a 2 Tier and 3 Tier Price policy without auxiliary solar panel generation

Price Tier	Algorithm	Cost	Comp. Time	Solution
2	ILP	1.03	1.36222	[60, 1295, 1265]
2	Tabu Search	1.03	0.44957	[165, 1294, 1158]
3	ILP	0.76	1.38187	[51, 177, 127]
3	Tabu Search	0.8	0.40936	[331, 172, 154]

Table 4.4: Numerical results of both algorithm for 5 appliances in a 2 Tier and 3 Tier Price policy without auxiliary solar panel generation

Price Tier	Algorithm	Cost	Comp. Time	Solution
2	ILP	-1.0	0.27043	[0, 0, 0, 0, 0]
2	Tabu Search	123.17	0.75092	[1238, 1200, 87, 1209, 1188]
3	ILP	-1.0	0.26852	[0, 0, 0, 0, 0]
3	Tabu Search	122.45	0.67452	[21, 269, 321, 837, 90]

demand. The Tabu search behaves in a similar way, by finding the a reasonably good solution with respect to the ILP one in fewer time. There can be outliers that diverge from the desired solution, this might be due to the inherit problem of the local search algorithm of starting from a randomized solution and getting stuck in a local minima.

Figure 4.6 compares the two algorithms in terms of costs and computational times. The Tabu Search shows to be incredibly powerful as far as Computational Times go, keeping them as low as 1.5 seconds, compared to the 71.3 seconds needed for the ILP to find the same solution.

This is important since, by considering a scheduling resolution of 1 minute, keeping the computational times lower than that allows the user to add as many appliances within that period of time and the algorithm will find a way to schedule them before each time frame ends. This makes our Tabu Search-based heuristic powerful enough to be considered in a on-line setting, where the scheduling happens on the fly at each time slot.

As far as the actual scheduling of the appliances goes, we frame in figure 4.7 how the scheduling of three appliances is with the usage of a solar panel. Immediately

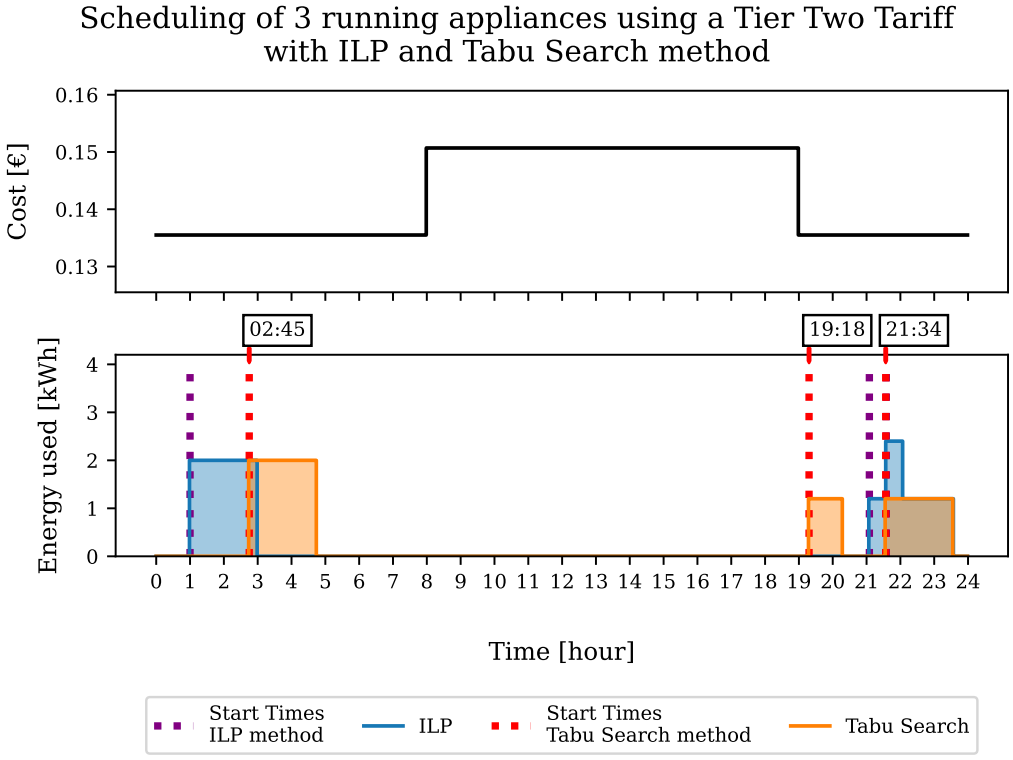

Figure 4.4: Power curve during the day while using three appliances scheduled by solving the Integer Linear Program (blue) and using the Tabu Search Heuristic algorithm (orange). The dashed lines mark the starting times for each appliance while using the ILP scheduler (purple dash line) or TS (red dash line). Scheduling done using a 2 Tier Price without the power generated by the photo voltaic panel

we see the difference between this schedule and the 4.4. The addition of a solar energy generation source (Figure 4.1) that is free to use, incentives the home energy management system to use said energy as much as possible to reduce costs and load off the grid. In fact, both methods (ILP and TS) look to use as much solar energy as possible shifting the energy to the mid-day period where the solar power is at it's maximum. This is particularly useful also in cases where appliances that are more power intensive are employed in the Smart Home environment. As shown in figure 4.8

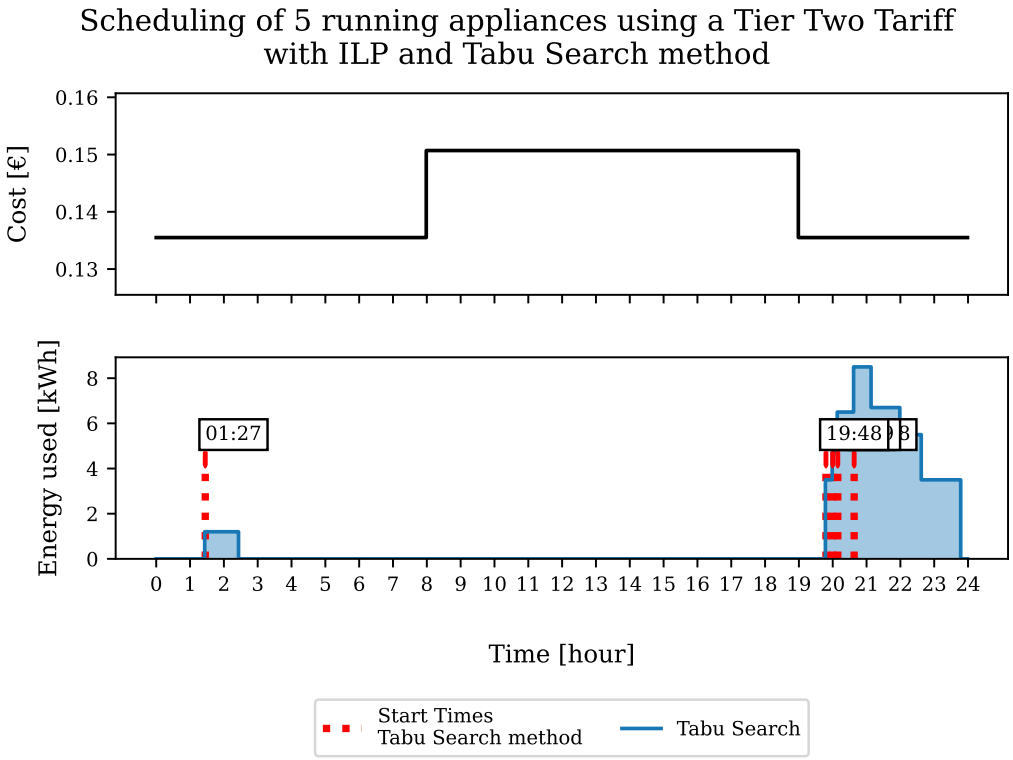

Figure 4.5: Power curve during the day while using five appliances scheduled by solving the Integer Linear Program (blue) and using the Tabu Search Heuristic algorithm (orange). ILP doesn't find a solution while TS-heuristic does

Table 4.5: Result table of the computation between MILP method and Tabu Search with PV panel generation

N. Appliances	Price Tier	Algorithm	Cost	Comp. Time
1	2	ILP	0.31	1.60427
		Tabu Search	0.31	0.99951
	3	ILP	0.33	0.74768
		Tabu Search	0.33	1.26947
3	2	ILP	0.54	2.20071
		Tabu Search	0.6	0.37956
	3	ILP	0.53	2.24028
		Tabu Search	0.54	0.58131
5	2	ILP	2.71	4.16266
		Tabu Search	2.79	0.46357
	3	ILP	2.61	4.20845
		Tabu Search	2.82	0.47825
7	2	ILP	3.47	8.31111
		Tabu Search	3.56	0.59414
	3	ILP	3.21	8.20338
		Tabu Search	3.54	0.61466
9	2	ILP	3.75	7.67445
		Tabu Search	3.92	0.96933
	3	ILP	3.4	9.42887
		Tabu Search	3.93	0.91321
11	2	ILP	-1.0	7.03815
		Tabu Search	49.46	2.07148
	3	ILP	-1.0	7.02396
		Tabu Search	37.25	2.17482

Table 4.6: Numerical results of both algorithm for 3 appliance in a 2 Tier Price and a 3 Tier Price setting with auxiliary solar panel generation

Price Tier	Algo	Comp. Time	Cost	Solution
2	ILP	2.20071	0.54	[840, 720, 900]
2	Tabu Search	0.37956	0.6	[701, 556, 805]
3	ILP	2.24028	0.53	[300, 840, 720]
3	Tabu Search	0.58131	0.54	[230, 711, 643]

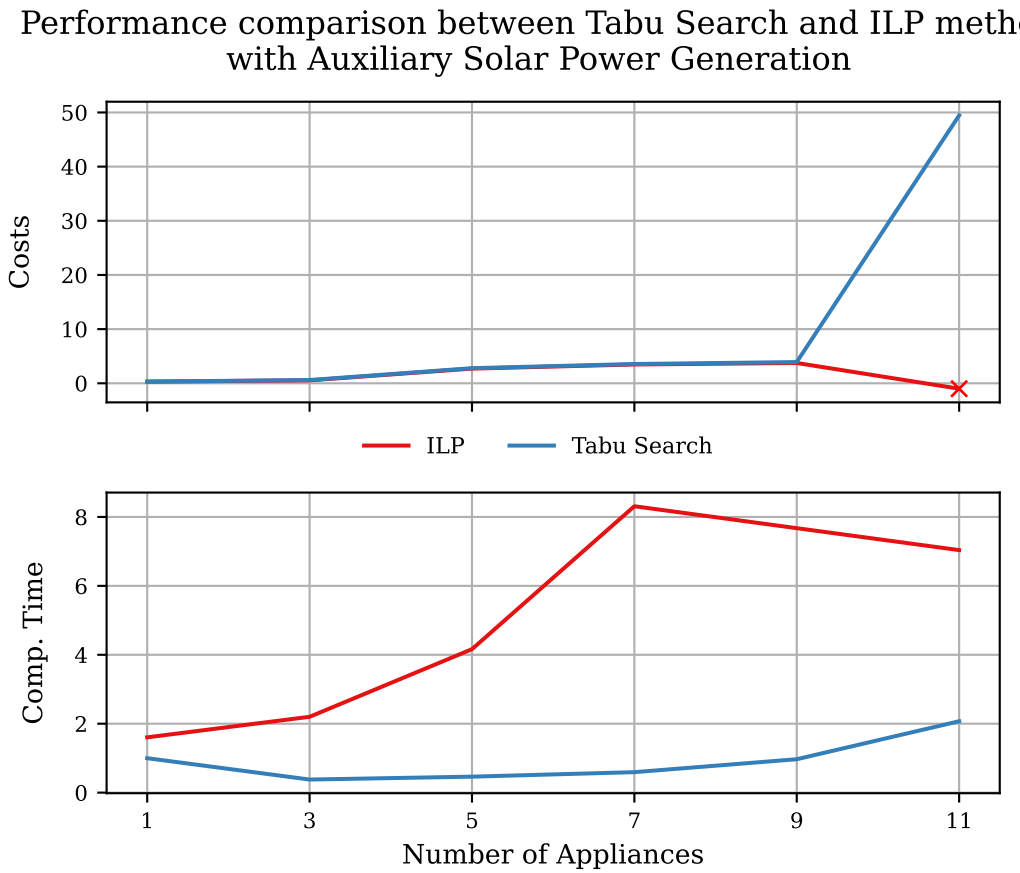

Figure 4.6: Numerical solutions for the Smart Home Appliance Scheduling Problem.

Table 4.7: Numerical results of both algorithm for 5 appliance in a 2 Tier Price and a 3 Tier Price setting with auxiliary solar panel generation

Price Tier	Algo	Comp. Time	Cost	Solution
2	ILP	4.16266	2.71	[1140. 630. 990. 930. 870.]
2	Tabu Search	0.46357	2.79	[497 917 895 22 637]
3	ILP	4.20845	2.61	[0. 300. 658. 360. 898.]
3	Tabu Search	0.47825	2.82	[188 1156 392 1211 638]

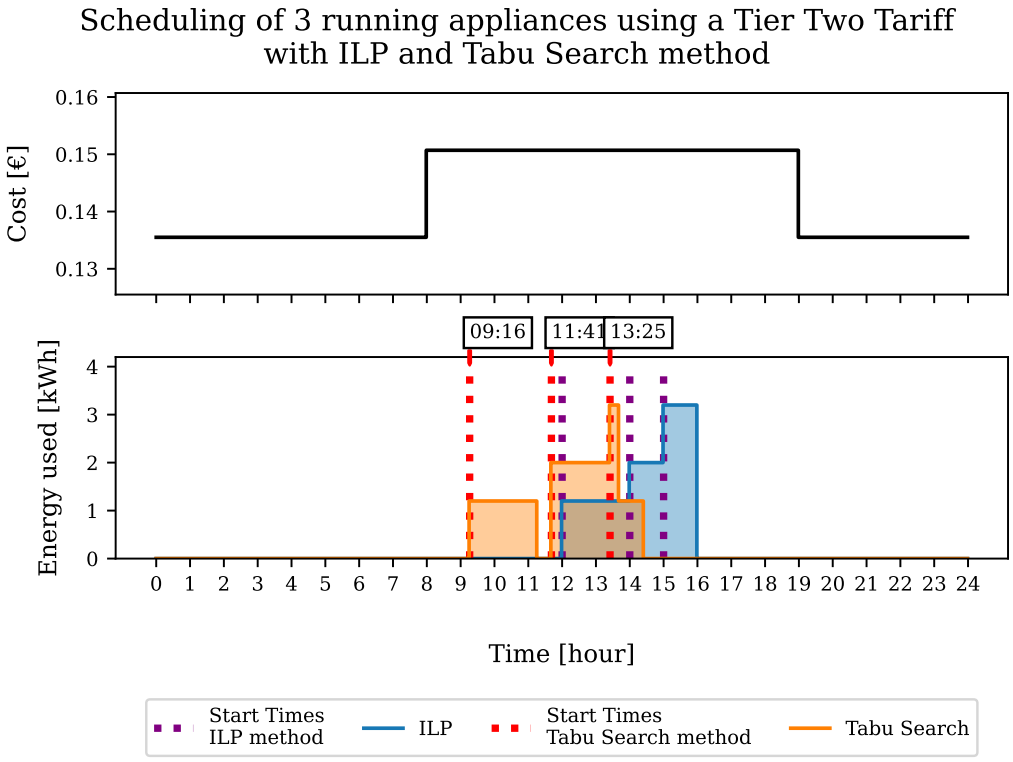

Figure 4.7: Power curve during the day while using three appliances scheduled by solving the Integer Linear Program (blue) and using the Tabu Search Heuristic algorithm (orange). The dashed lines mark the starting times for each appliance while using the ILP scheduler (purple dash line) or TS (red dash line). Scheduling done using a 2 Tier Price with the power generated by the photo voltaic panel

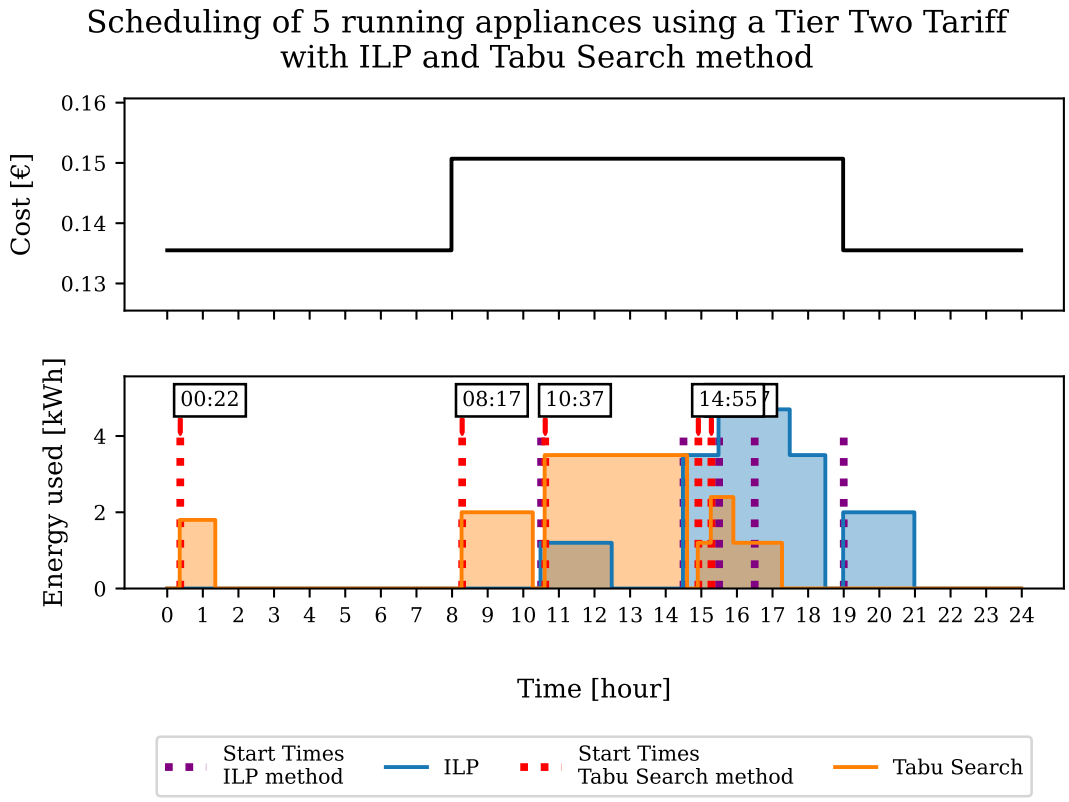

Figure 4.8: Power curve during the day while using five appliances. Scheduling done using both the ILP method and the proposed TS-based heuristic

Chapter 5

Conclusions

The growing interest in how we manage the energy is the main motivation behind this research. How we use the limited resource at our disposal is of critical importance in today's world. The research in this field is rich and thriving and many techniques have been proposed in order to intelligently adjust our consumption behaviour so as to fit the needs of our planet without losing comfort and personal satisfaction in our life. The aim of this thesis was to study, implement and apply novel approaches to the solution of the Smart Home Appliance Scheduling Problem (SHASP), comparing them to an already formulated Mixed-Integer Liner Programming formulation of the problem in terms of speed of resolution and accuracy.

Two main algorithms that were considered: Reinforcement Learning and Local Search Heuristics. Reinforcement Learning addresses the decision-making problem in a brand new way, achieving incredible results in simulated environments of zero-sum games (Chess, Go, StartCraft 2, etc.). In order to adapt the Appliance Scheduling Problem to a Markov Decision Process, different design choices have been tried and simulated using OpenAI Gym to project the environments. Although the present literature that shows the possibility of appling RL methodologies to Scheduling Problem, the dimentionality of the problem under exam and the timing requirements for reaching a solution make it infeasible for our case: being heavily reliant on the number of simulations needed to reach an optimal policy might not be the best choice when speed at which a feasible solution is found is the key performance indicator for the comparison between the proposed methods. Relaxing the Markov property of the SHASP might also be another solution but that would help the implementation of this kind of methodology but at that point, choosing

another heuristic method might be better.

For this reason, a Local Search heuristic based on the Tabu Search has been developed. The designed heuristic improved the classic Tabu Search so as to avoid getting stuck into a local minima by adding diversification with random start.

The chosen case study for this research focused on a Smart Home, connected to the Power Grid and using photovoltaic power generation as renewable energy source. The appliances considered within the smart home where schedulable and non interruptible.

The results suggests that in the presence of auxiliary power generation the proposed heuristic improves significantly the computational times needed to solve the SHASP without loss in accuracy of the solution, that is very close to the one given by the MILP.

5.1 Future Works

Much can still be done with the research presented in this thesis. Regarding the Reinforcement Learning approach to the problem, while the results obtained were unsatisfactory, other iterations of the MDP can be tried in order to reach a design that can be handled better by off-the-shelf RL agents. This approach becomes prominent and deserving of a more thorough study when multiple agents are involved such as in energy communities (Multiple Smart Homes, multiple renewable energy sources, battery systems and electric vehicles stations). Further analysis has to be done on the impact of different appliances characteristics such as possibility to be scheduled or not and the addition of preemption. Moreover, the usage of real data might also of careful interest, both regarding the RES generation and consumption patters of the user.

Appendix A

Additional Figures

A.1 Results

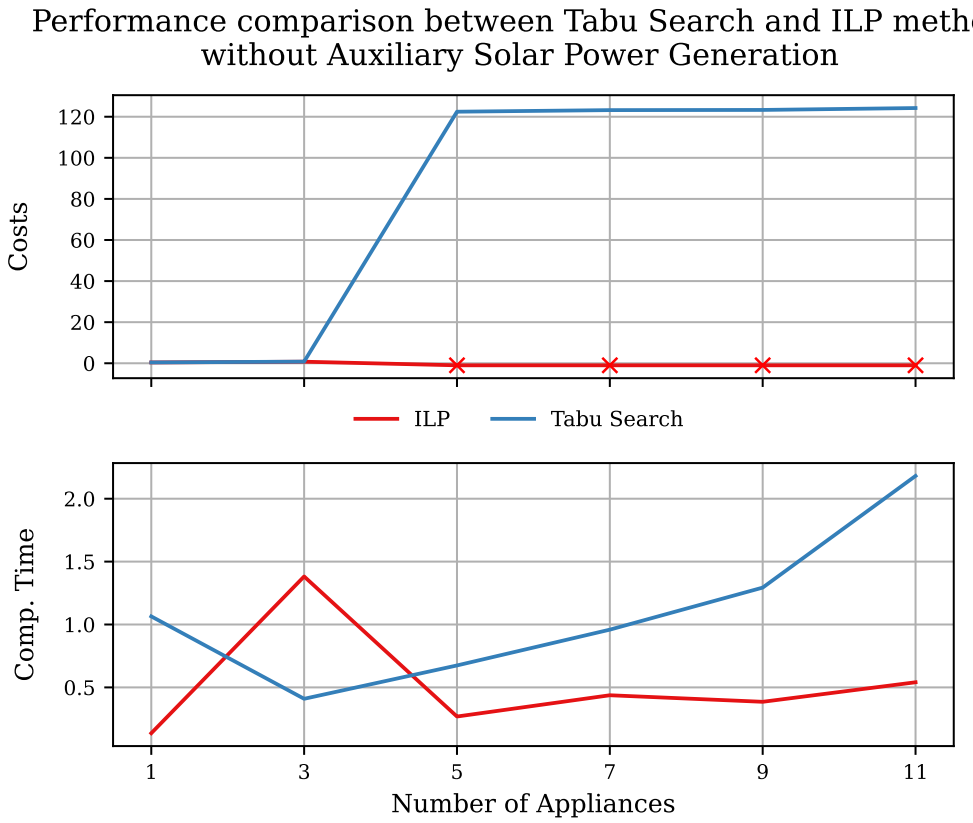

Figure A.1: Performance evaluation of ILP vs the proposed heuristic on both Cost of each instance and computational time to reach the solution, without using any photovoltaic power generation

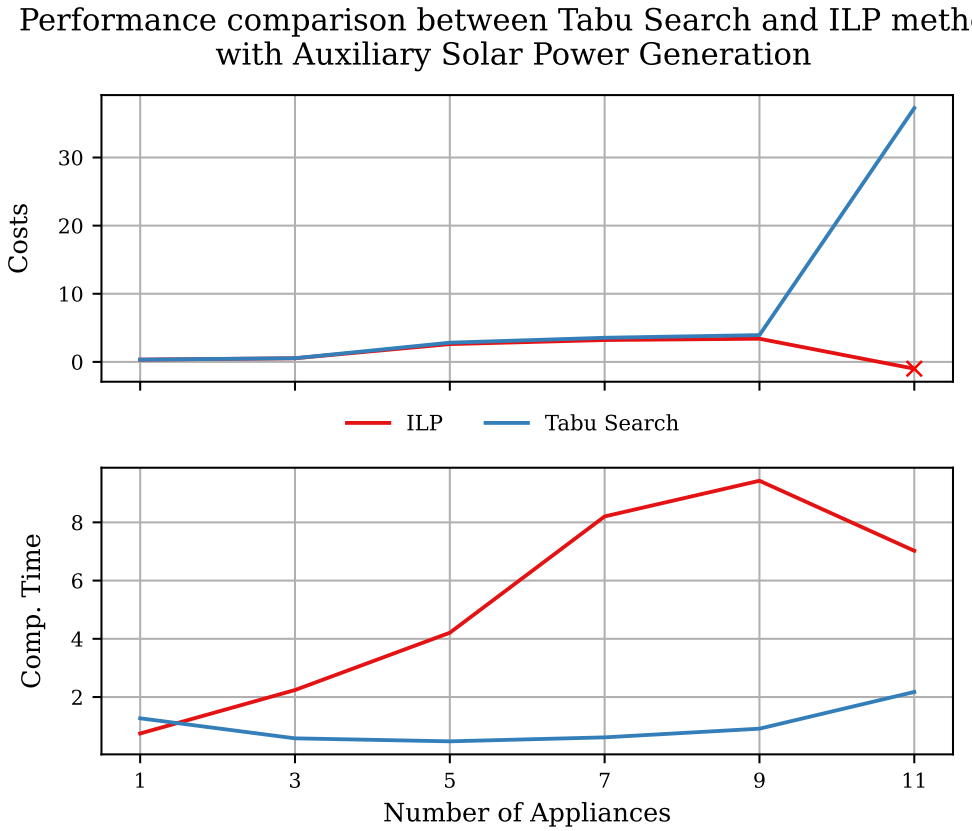

Figure A.2: Performance evaluation of ILP vs the proposed heuristic on both Cost of each instance and Computational Time to reach the solution, using photovoltaic power generation

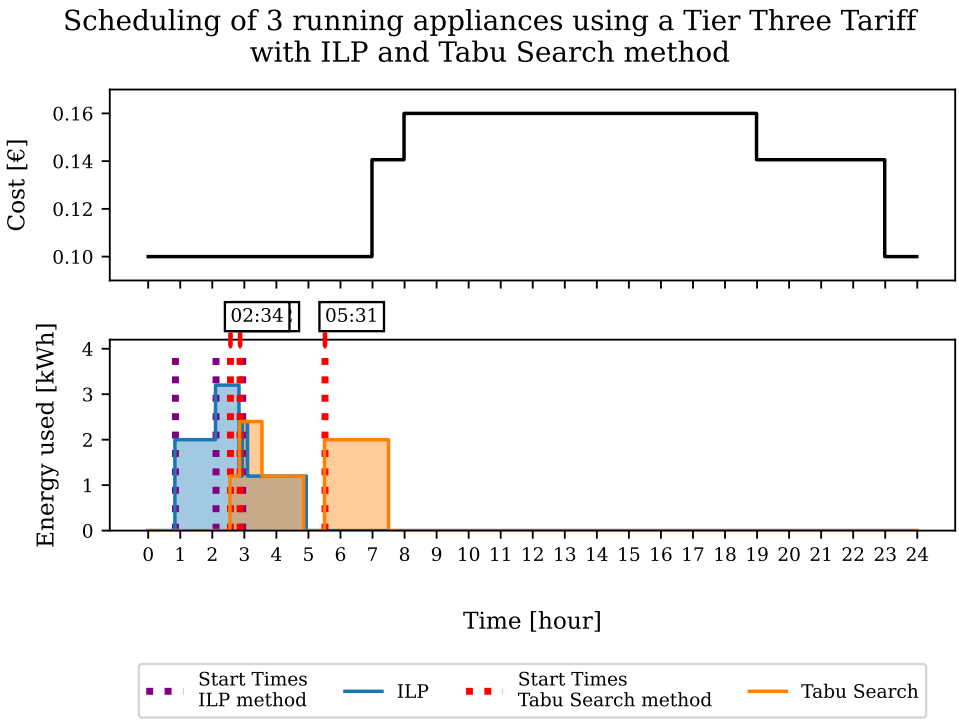

Figure A.3: scheduling of 3 appliances without a photovoltaic panel

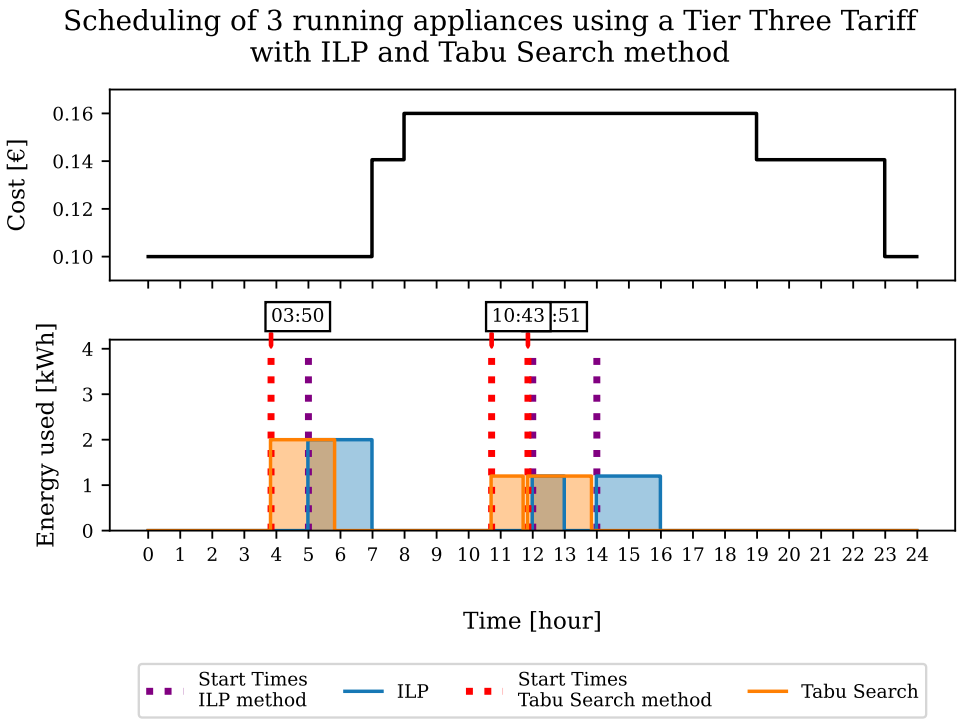

Figure A.4: scheduling of 3 appliances with a photovoltaic panel

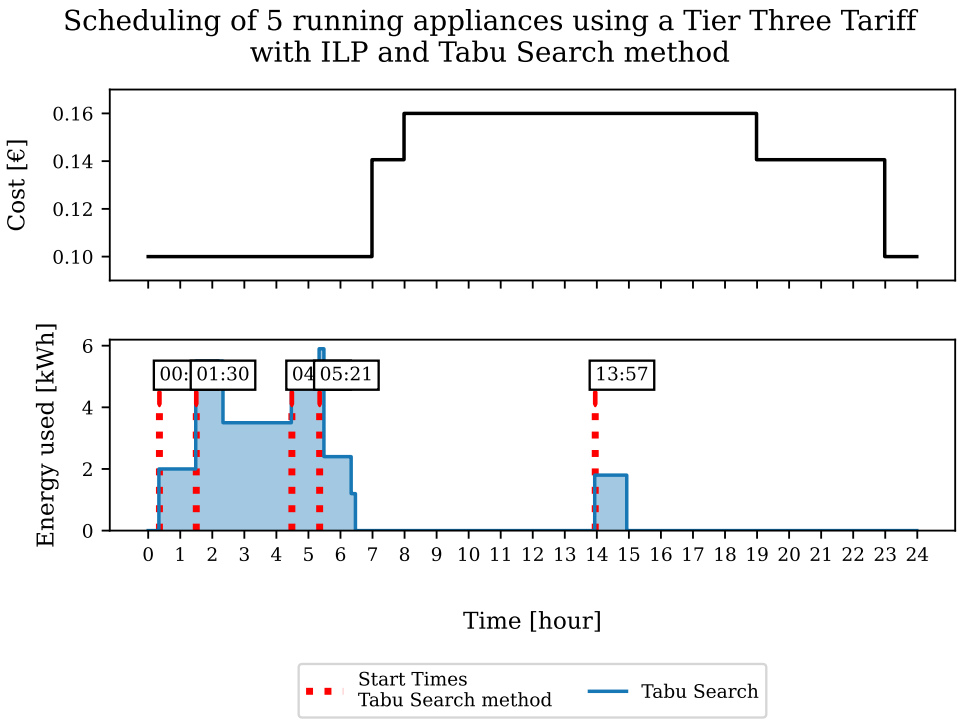

Figure A.5: Scheduling of 5 appliances without photovoltaic panel

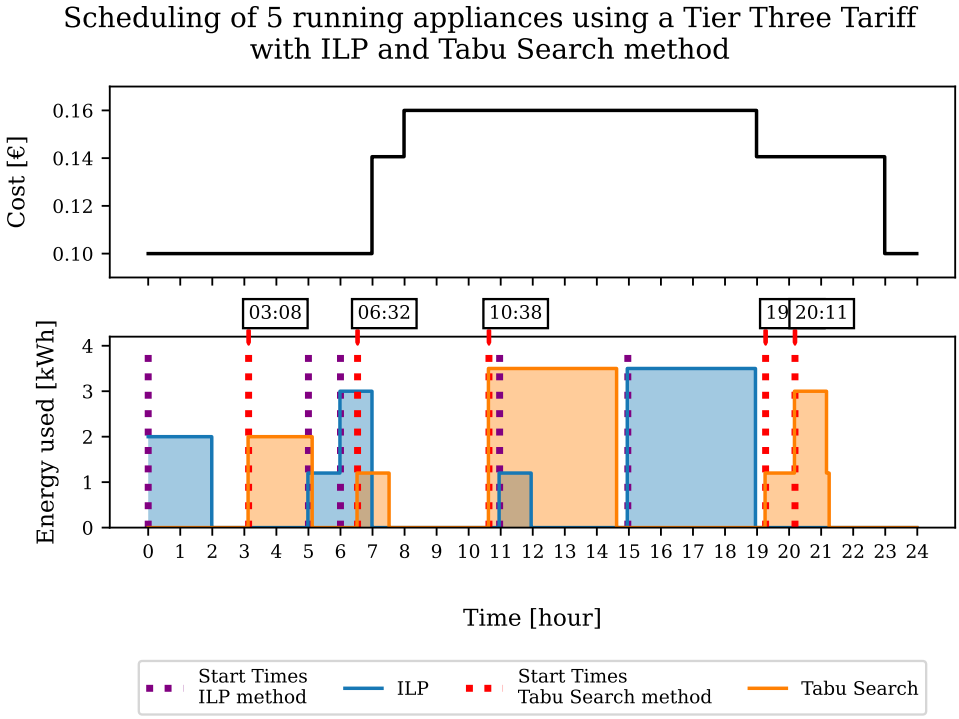

Figure A.6: Scheduling of 5 appliances with photovoltaic panel

Bibliography

[1] Avi Gopstein, Cuong Nguyen, Cheyney O'Fallon, Nelson Hastings, and David Wollman. *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0.* en. 2021-02-18 2021. DOI: https://doi.org/10.6028/NIST.SP.1108r4 (cit. on p. 4).

[2] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. «Smart grid—The new and improved power grid: A survey». In: *IEEE communications surveys & tutorials* 14.4 (2011), pp. 944–980 (cit. on p. 5).

[3] Doğan Çelik and Mehmet Emin Meral. «Current control based power management strategy for distributed power generation system». In: *Control Engineering Practice* 82 (2019), pp. 72–85 (cit. on p. 5).

[4] Amir Safdarian, Mahmud Fotuhi-Firuzabad, and Matti Lehtonen. «Benefits of demand response on operation of distribution networks: A case study». In: *IEEE systems journal* 10.1 (2014), pp. 189–197 (cit. on p. 5).

[5] Amit Shewale, Anil Mokhade, Nitesh Funde, and Neeraj Dhanraj Bokde. «An overview of demand response in smart grid and optimization techniques for efficient residential appliance scheduling problem». In: *Energies* 13.16 (2020), p. 4266 (cit. on pp. 7, 11).

[6] Hussein Jumma Jabir, Jiashen Teh, Dahaman Ishak, and Hamza Abunima. «Impacts of demand-side management on electrical power systems: A review». In: *Energies* 11.5 (2018), p. 1050 (cit. on p. 8).

[7] Mohamed H Albadi and Ehab F El-Saadany. «A summary of demand response in electricity markets». In: *Electric power systems research* 78.11 (2008), pp. 1989–1996 (cit. on p. 8).

[8] Kai Ma, Ting Yao, Jie Yang, and Xinping Guan. «Residential power scheduling for demand response in smart grid». In: *International Journal of Electrical Power & Energy Systems* 78 (2016), pp. 320–325. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2015.11.099. URL: https://www.

sciencedirect.com/science/article/pii/S014206151500530X (cit. on p. 9).

[9] Ditiro Setlhaolo and Xiaohua Xia. «Optimal scheduling of household appliances with a battery storage system and coordination». In: *Energy and Buildings* 94 (2015), pp. 61–70 (cit. on p. 11).

[10] Hamed Shakouri and Aliyeh Kazemi. «Multi-objective cost-load optimization for demand side management of a residential area in smart grids». In: *Sustainable cities and society* 32 (2017), pp. 171–180 (cit. on p. 11).

[11] Zhenyu Wang and Guilin Zheng. «Residential appliances identification and monitoring by a nonintrusive method». In: *IEEE transactions on Smart Grid* 3.1 (2011), pp. 80–92 (cit. on p. 11).

[12] Tùng T Kim and H Vincent Poor. «Scheduling power consumption with price uncertainty». In: *IEEE Transactions on Smart Grid* 2.3 (2011), pp. 519–527 (cit. on p. 11).

[13] Zhi Chen, Lei Wu, and Yong Fu. «Real-Time Price-Based Demand Response Management for Residential Appliances via Stochastic Optimization and Robust Optimization». In: *IEEE Transactions on Smart Grid* 3.4 (2012), pp. 1822–1831. DOI: 10.1109/TSG.2012.2212729 (cit. on p. 11).

[14] Takeshi Yamada and Ryohei Nakano. «Job shop scheduling». In: *IEE control Engineering series* (1997), pp. 134–134 (cit. on p. 13).

[15] Ahmed R Sadik and Bodo Urban. «Flow shop scheduling problem and solution in cooperative robotics—case-study: One cobot in cooperation with one worker». In: *Future Internet* 9.3 (2017), p. 48 (cit. on p. 13).

[16] Zhuang Zhao, Won Cheol Lee, Yoan Shin, and Kyung-Bin Song. «An optimal power scheduling method for demand response in home energy management system». In: *IEEE transactions on smart grid* 4.3 (2013), pp. 1391–1400 (cit. on p. 13).

[17] Federico Della Croce, Michele Garraffa, Fabio Salassa, Claudio Borean, Giuseppe Di Bella, and Ennio Grasso. «Heuristic approaches for a domestic energy management system». In: *Computers & Industrial Engineering* 109 (2017), pp. 169–178 (cit. on p. 15).

[18] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. «Optimization by simulated annealing». In: *science* 220.4598 (1983), pp. 671–680 (cit. on p. 17).

[19] Fred Glover. «Tabu search—part I». In: *ORSA Journal on computing* 1.3 (1989), pp. 190–206 (cit. on pp. 17, 18).

[20] Fred Glover. «Tabu search—part II». In: *ORSA Journal on computing* 2.1 (1990), pp. 4–32 (cit. on p. 17).

[21] David Silver. *Lectures on Reinforcement Learning*. URL: `https://www.davidsilver.uk/teaching/`. 2015 (cit. on p. 22).

[22] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on pp. 22, 24, 26).

[23] Richard E Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Princeton university press, 2015 (cit. on p. 27).

[24] Julien Perez, Balázs Kégl, and Cecile Germain-Renaud. «Non-Markovian Reinforcement Learning for Reactive Grid scheduling». In: *7e Plateforme AFIA, Association Française pour l'Intelligence Artificielle, Chambéry, 16 au 20 mai 2011* (2011), p. 327 (cit. on p. 29).

[25] Steven D Whitehead and Long-Ji Lin. «Reinforcement learning of non-Markov decision processes». In: *Artificial intelligence* 73.1-2 (1995), pp. 271–306 (cit. on p. 29).

[26] Maor Gaon and Ronen Brafman. «Reinforcement learning with non-markovian rewards». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 3980–3987 (cit. on p. 29).