UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Network resource allocation policies with energy transfer capabilities

Nicola Piovesan

Universitat Politècnica de Catalunya

Department of Network Engineering

PhD Thesis

Network Resource Allocation Policies with Energy Transfer Capabilities

Nicola Piovesan

A thesis submitted in fulfillment of the requirements for the
International doctoral degree

Director Dr. Paolo Dini
Centre Tecnològic de Telecomunicacions de Catalunya

Tutor Prof. Miquel Soriano
Universitat Politècnica de Catalunya

Nicola Piovesan

Network Resource Allocation Policies with Energy Transfer Capabilities

PhD Thesis

Director: Dr. Paolo Dini

Tutor: Prof. Miquel Soriano

Universitat Politècnica de Catalunya

Department of Network Engineering

C. Jordi Girona, 1-3

08034 Barcelona

Abstract

During the last decades, mobile network operators have witnessed an exponential increase in the traffic demand, mainly due to the high request of services from a huge amount of users. The trend is of a further increase in both the traffic demand and the number of connected devices over the next years. The traffic load is expected to have an annual growth rate of 53% for the mobile network alone, and the upcoming industrial era, which will connect different types of devices to the mobile infrastructure including human and machine type communications, will definitely exacerbate such an increasing trend.

The current directions anticipate that future mobile networks will be composed of ultra dense deployments of heterogeneous Base Stations (BSs), where BSs using different transmission powers coexist. Accordingly, the traditional Macro BSs layer will be complemented or replaced with multiple overlapping tiers of small BSs (SBSs), which will allow extending the system capacity. However, the massive use of Information and Communication Technology (ICT) and the dense deployment of network elements is going to increase the level of energy consumed by the telecommunication infrastructure and its carbon footprint on the environment.

Current estimations indicates that 10% of the worldwide electricity generation is due to the ICT industry and this value is forecasted to reach 51% by 2030, which imply that 23% of the carbon footprint by human activity will be due to ICT. Environmental sustainability is thus a key requirement for designing next generation mobile networks.

Recently, the use of Renewable Energy Sources (RESs) for supplying network elements has attracted the attention of the research community, where the interest is driven by the increased efficiency and the reduced costs of energy harvesters and storage devices, specially when installed to supply SBSs. Such a solution has been demonstrated to be environmentally and economically sustainable in both rural and urban areas. However, RESs will entail a higher management complexity. In fact, environmental energy is inherently erratic and intermittent, which may cause a fluctuating energy inflow and produce service outage. A proper control of how the

energy is drained and balanced across network elements is therefore necessary for a self-sustainable network design.

In this dissertation, we focus on energy harvested through solar panels that is deemed the most appropriate due to the good efficiency of commercial photovoltaic panels as well as the wide availability of the solar source for typical installations. The characteristics of this energy source are analyzed in the first technical part of the dissertation, by considering an approach based on the extraction of features from collected data of solar energy radiation.

In the second technical part of the thesis we introduce our proposed scenario. A federation of BSs together with the distributed harvesters and storage devices at the SBS sites form a micro-grid, whose operations are managed by an energy management system in charge of controlling the intermittent and erratic energy budget from the RESs. We consider load control (i.e., enabling sleep mode in the SBSs) as a method to properly manage energy inflow and spending, based on the traffic demand. Moreover, in the third technical part, we introduce the possibility of improving the network energy efficiency by sharing the exceeding energy that may be available at some BS sites within the micro-grid.

Finally, a centralized controller based on supervised and reinforcement learning is proposed in the last technical part of the dissertation. The controller is in charge of opportunistically operating the network to achieve efficient utilization of the harvested energy and prevent SBSs blackout.

Acknowledgement

My deepest gratitude goes first and foremost to my supervisors, Dr. Paolo Dini and Dr. Marco Miozzo, for their guidance and encouragement during my postgraduate study and research activity. I would also like to express my gratitude to Dr. David Lopez-Perez for the support and valuable advice he gave me during my secondment period in Nokia Bell Labs.

I would like to thank my colleagues Dagnachew and Trinh, for their friendship and the good times we spent together.

I would like to express my heartfelt gratitude to Giulia for her love, support and precious motivation.

I would like to thank my beloved parents and sister for the continuous support all through my life and for settling a positive and constructive environment around me.

Finally, I would like to acknowledge the European Commission for supporting me in this work with a Marie Skłodowska-Curie Fellowship.

Acronyms

5G 5th Generation Mobile Networks

AC Alternating Current

ANN Artificial Neural Network

ARPU Average Revenue Per Unit

BS Base Station

C-RAN Cloud-Radio Access Network

CAPEX Capital Expense

CoMP Coordinated Multi Point

DC Direct Current

DLC Direct Load Control

DP Dynamic Programming

DQL Deep Q-Learning

DR Demand Response

EE Energy Efficiency

EPN Energy Packet Network

EC Energy Controller

HetNet Heterogeneous Network

ICT Information and Communications Technology

IoT Internet of Things

ITU International Telecommunication Union

IL Imitation Learning

IEMS Intelligent Energy Management System

LTE Long-Term Evolution

MBS Macro Base Station

MNO Mobile Network Operator

ML Machine Learning

NFV Network Function Virtualization

OPEX Operative Expenditure

PC Principal Component

PCA Principal Component Analysis

PV Photovoltatic

QoE Quality of Experience

QoS Quality of Service

QL Q-Learning

RAN Radio Access Network

RB Resource Block

ReLU Rectified Linear Unit

RES Renewable Energy Source

RF Radio Frequency

RL Reinforcement Learning

SBS Small Base Station

SDN Software Defined Networking

SG Smart Grid

SINR Signal to Interference plus Noise Ratio

TD Temporal Difference

UE User Equipment

UAE Under-complete Autoencoder

Contents

1 Introduction

The research work discussed in this PhD dissertation has been carried out as part of the SCAVENGE project, funded by the European Union in the framework of the Horizon 2020 research and innovation programme. The project tackles sustainable design, protocols, architectures and algorithms for next generation 5G cellular networks. The overall purpose is to allow mobile systems and especially their constituting base stations, mobile devices and sensors to take advantage of sources harvesting ambient energy (such as renewable sources).

1.1 Motivation

We live in the digital era. Dematerialization is becoming a reality, humans and machines alike are globally connected through the Internet. The International Telecommunication Union (ITU) estimated that 750 millions households are online and that there exist almost as many mobile subscribers as people in the world (around 6.8 billions) [46]. The trend is of a further increase in the traffic demand, in the number of offered and connected devices, especially mobile. The traffic load forecast in [19] is of an annual traffic growth rate of 53%, for the mobile traffic alone. This new industry era is undoubtedly opening up new possibilities for individuals as well as new opportunities for businesses and organizations. However, the massive use of Information and Communications Technology (ICT) is also increasing the level of energy consumed by the telecommunication infrastructure and its footprint on the environment. In a report of 2013, the Digital Power Group [69] has calculated that 10% of the worldwide electricity generation is due to the ICT industry, which is more than twice that of the avionic sector. The report also highlights that the ICT energy consumption Compound Annual Growth Rate is around 10%. In fact, forecasts for 2030 are that 51% of the electricity consumption and 23% of the carbon footprint by human activity will be due to ICT [4]. Hence, any future development in the ICT technology and in its infrastructure should be undertaken with environmental sustainability in mind.

Besides such an increase in the demand, the ICT industry has to solve an economical problem, since operators' Average Revenue Per Unit (ARPU) is decreasing every year. The case of Vodafone Germany is particularly striking: its ARPU has been shrinking

annually by 6% on average in the period 2000-2009. One of the reasons of this is the annual increase of the Operative Expenditure (OPEX) of its network. Energy has been dominating these costs: it has been calculated that the energy bill equals the cost of the personnel required to run and maintain the network for a western Europe Mobile Network Operator (MNO) in 2007 [27]. Considering the rise in the energy price during the last few years, we conclude that energy saving is key for the economical sustainability of ICT.

In this thesis, we discuss the crucial role of energy in the design of future networks, by considering *mobile networks*, which are growing the most among all ICT sectors in terms of number of subscribers, traffic demand, connected devices and offered services [19].

1.2 Reference Scenario

Current trends anticipate that 5G mobile networks will be composed of *ultra dense* deployments of heterogeneous Base Stations (BSs), where BSs using different transmission powers coexist to provide the 1000x network capacity increase that is required by 2020 [62]. Accordingly, the traditional Macro Base Station (MBS) layer will be complemented or replaced with multiple overlapping tiers of smaller BSs, which extend the system capacity, thanks to a higher spatial reuse and to a better spectral efficiency. Despite such benefits, researchers have already identified new issues raised by an ultra dense scenario, such as: user association and mobility management, interference management and mitigation, MBS offloading, and energy saving [50]. Also, 5G subscribers will be equipped with a large and diverse set of devices and BSs may need to support high-rate mobile equipment (such as smartphones and laptops) [5] as well as a huge number of low-rate devices (such as environmental or wearable sensors) [34], as envisaged by the Internet of Things (IoT) paradigm. This makes new generation networks challenging to operate, control and monitor. Moreover, such systems are also very demanding in terms of energy consumption from the power grid, due to their high capacity requirements. Different architectural designs have been proposed for next generation mobile networks including: 1) Cloud-Radio Access Network (C-RAN) [80, 3], 2) Software Defined Networking (SDN) [30], 3) Network Function Virtualization (NFV) [58] and 4) Fog Computing [79, 44]. All these proposals rely on the cloud principle of sharing storage and computing resources. Moreover, they enable control and data plane decoupling and entail a pure software implementation of network functions, which may be opportunistically placed in different network elements.

Such architectural proposals offer higher flexibility and scalability to operate, control and monitor new generation networks, however, a big effort is still necessary to reduce their energy demand. In fact, they are not specifically designed to reduce the energy consumption of 5G networks and to potentially make them energy self-sufficient. Based on the literature reviewed in Chapter 2, we acknowledge that there are studies that try to improve the energy efficiency of these designs. Nevertheless, we have identified the following limitations of existing schemes: (i) their energy savings are still insufficient and most of the research is still in a preliminary stage, (ii) they do not involve energy harvesting capabilities and energy self-sustainability. In this dissertation, we advocate gathering environmental energy through dedicated harvesting hardware to supply BSs. This translates into OPEX savings and into a reduction of the environmental footprint of ICT. The Capital Expense (CAPEX) can also be reduced [82] through the adoption of BSs with a small form-factor, as these require smaller energy amounts to be operated and this lessens the requirements in terms of harvesting and energy storage capabilities.

5G BSs can be classified into two main groups, depending on transmission power and coverage range. We distinguish between MBSs with a transmission power range of 20-40 W covering few kilometers and Small Base Stations (SBSs) with a transmission power of 0.05-6 W covering from few meters to few hundred meters. More details on the characteristics of those BS are given in Chapter 3. The power consumption at full system load of the different types of BSs can range from about 6 W for a femto BS to 1 kW for a MBS [6, 22, 84].

Recently, the use of sustainable energy for supplying network elements has attracted the attention of the research community, where the interest is driven by the increased efficiency and the reduced costs of energy harvesters and storage devices, specially when installed to supply SBSs [39]. Such a solution has been demonstrated to be environmentally and economically sustainable in both rural and urban areas [114].

Figure 1.1 illustrates our reference scenario, which includes BSs, mobile devices, sensors, energy harvesters and energy storage devices. In particular, we envision a Radio Access Network (RAN) setup in which a hierarchical cell structure is deployed within the same geographical area with BSs of different scale factors, transmission power, computational capabilities and coverage areas [62]. We focus on a two-tier architecture where SBSs are in the first tier and are powered only by renewable energy they harvest through a solar panel and store in a battery. The second tier is composed of MBSs powered by grid energy and used for backup operations when a SBS is switched off. In this way, SBSs are utilized to increase the system capacity whereas energy harvesters and energy storage devices ensure energy sustainability. Furthermore, energy cooperation and transfer are utilized to balance the energy reserve across BSs.

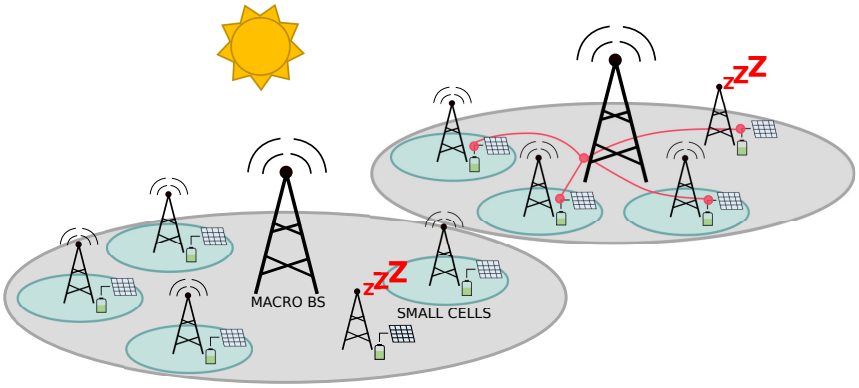


Figure 1.1: Diagram illustrating the reference scenario. SBSs are utilized to increase the system capacity, energy harvester and energy storage devices ensure energy sustainability. Energy cooperation and transfer are accounted to balance the energy harvesting reserve across BSs.

Renewable Energy Sources (RESs) will entail a higher management complexity. In fact, environmental energy, such as solar and wind, is inherently erratic and intermittent, which may cause fluctuating energy inflow and produce service outage. A proper control of how the energy is drained and balanced across network elements is therefore necessary for a self-sustainable network design. The flexibility introduced by the cloud principles into the 5G architecture will definitely support the design of optimal strategies for network energy management. Moreover, sustainable design of 5G systems shall rely on a set of procedures enabling energy efficient communication. While reviewing the scientific literature, it appears that the most promising approach to reduce the energy consumption of mobile networks is to enable sleep modes in some network elements during periods of low traffic [36].

In such a context, SBSs may transfer their energy to provide ancillary services to the next generation electrical grid, the Smart Grid (SG). The SG is the technological paradigm proposed to enable highly efficient energy production, transport, and consumption along the whole chain, from the source to the user. In particular, a key concept is the design of micro-grids, which has been defined by the US Department of Energy as “a group of interconnected loads and distributed energy resources (mainly renewables) within clearly defined electrical boundaries that act as a single controllable entity with respect to the power grid”. A micro-grid can connect and disconnect from the grid to enable it to operate in both grid connected and island mode. Similarly, the European Union recently released the EU Winter Package, aimed at providing guidelines for the next generation of power grids. The main idea is to foster cooperation among local energy communities by providing them with the infrastructure to work in island mode and with market-based retail energy prices.

In our reference scenario, the federation of BSs together with the distributed harvesters and storage devices at the SBS sites form a micro-grid, whose operations are

managed by an energy management system in charge of controlling the intermittent and erratic energy budget from the RESs. We consider load control (i.e., enabling sleep mode in the SBSs) as a method to properly manage energy inflow and spending, based on the traffic demand. Moreover, we propose the possibility of improving the network energy efficiency by sharing the exceeding energy that may be available at some BS sites within the micro-grid. This combination of sleep modes and energy sharing represents an advance with respect to state-of-the-art approaches, as will be also discussed in Chapter 2.

1.3 Objectives and Methodology

The objective of the thesis is to investigate on the design of possible integration architectures between the energy harvesting mobile network and the SG. In particular, the main scope is to study the capability of 5G mobile networks of intelligently routing energy in a micro grid of interconnected conventional/renewable energy sources and loads. This is needed to satisfy the demand of communication networks while avoiding energy outages in zones with high user density and/or low ambient energy availability.

In particular, the following tasks need to be performed to achieve the above-mentioned objectives:

1. **Self-sustainable mobile networks.** Review of the state-of-the-art on the area of self-sustainable mobile networks. In particular, we focus on paradigms to control energy inflow and spending within the micro-grid. In this task, we pose particular attention to the methodology used to solve the energy control problem, including optimal control and machine learning.

2. **Characterization of the RESs.** It is fundamental to properly characterize the RESs since a consistent part of the network, represented by the SBSs, is supplied by them. In this dissertation, we focus on energy harvested through solar panels that is deemed the most appropriate due to the good efficiency of commercial photovoltaic panels as well as the wide availability of the solar source for typical installations [60].

3. **Load control algorithms.** SBSs may install self-organizing agents, which enable intelligent energy management policies, such as Direct Load Control. In this way, the agents can dynamically turn ON and OFF the SBS to minimize the consumption of grid energy and the system outage. In particular, the bounds on the network performance and the monetary cost need to be estimated

to analyze the feasibility of the proposed approach in a given geographical location.

4. **Integration with the SG.** In some cases, SBSs may experience high energy harvesting (e.g., in summer months) and this energy may not be stored into the batteries due to their limited capacity. Therefore, we consider the possibility of exchanging the exceeding energy available at the SBS sites with the MBS to reduce its grid energy consumption and increase the mobile network self-sustainability.

Much effort has been posed on the methodological part to solve the energy control and the relevant optimization problem. In particular, we have adopted Dynamic Programming (DP) and Machine Learning (ML) as efficient methods to deal with our problem.

DP has been applied to implement the offline load control algorithm. DP is a method that allow simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. The use of DP is particularly helpful since it enables solving optimization problems based on step-by-step decisions, in which future events have an impact on the current decision. In this way, the optimal control can be computed given the a-priori knowledge of the energy harvesting and traffic demand processes. In particular, we formulated the classical DP problem as a graph-theory shortest-path search problem, and we solved it by using an efficient variant of the popular label-correcting algorithm. The graph is build in a step-by-step fashion while discarding the nodes leading to unfeasible solutions. This method allows to efficiently solve the complex optimal load control problem. The optimal control solution provides a bound on the performance that can be achieved by our proposed architecture.

ML algorithms have been used as an efficient way to characterize the solar energy source and to perform online load control. Unsupervised learning methods have been used to learn hidden features of the solar energy generation from a PV system. Normally, this geographical analysis is performed using solar maps, which provide easy-understandable information. The total annual solar electricity generation from a PV system is used to characterize national and regional differences [115]. In this dissertation, we adopt unsupervised learning methods to extract hidden features that may give a more accurate characterization of the process beyond that usual metric. In particular, clustering the extracted features allows grouping the locations according to their similar *behaviors* in terms of solar energy generation. Moreover, supervised and reinforcement learning have been used to design different implementations of a centralized energy-aware RAN controller. These methods allow performing the energy management in an online fashion, without any prior

knowledge of the system. In the supervised case, a learning agent implemented in the centralized controller learns the optimal policies from labeled data collected by a supervisor. The creation of the training set is a hard task since it relies on an expert supervision. For this reason, supervised learning approaches are generally feasible in scenarios with a limited number of SBSs due to complexity issues. Therefore, we investigated solutions based on reinforcement learning as a way to deal with the problem complexity. In details, reinforcement learning methods allows learning from the interaction with the environment and no specific supervision is required.

1.4 Outline of the Thesis

A brief overview of the content of the following chapters is given in this section.

Chapter 2 - State of the Art

We describe the state of the art about energy efficiency in mobile networks. Moreover, we introduce the concept of *energy cooperation* in a renewable powered mobile network, and we provide a classification of the literature in this field.

The work presented in this chapter has been published in the following papers:

- N. Piovesan, A. Fernandez Gambin, M. Miozzo, M. Rossi, P. Dini, "Energy sustainable paradigms and methods for future mobile networks: A survey", *Computer Communications*, Volume 119, 2018, Pages 101-117.

Chapter 3 - Modeling

We provide a description of the mathematical models adopted in this dissertation. In particular, we focus on the models for the solar energy generation, the mobile traffic demand and the BSs power consumption.

Chapter 4 - Mathematical Framework

We provide a description of the mathematical tools adopted in this dissertation. We give an introduction on DP, focusing on finite state systems, and we describe the shortest-path algorithm adopted to efficiently solve our DP problem. Finally, an

introduction on ML and neural networks is presented. DP and the shortest-path algorithm have been adopted in Chapter 6 and 7 to compute the performance bounds of the optimal direct load control approach. Moreover, ML has been adopted in Chapter 5 to characterize the solar energy source and in Chapter 8 to implement different online load control algorithms.

Chapter 5 - Characterization of the Solar Energy

We discuss an unsupervised method to learn hidden features of the solar energy generation from a photovoltaic system that may give a more accurate characterization of the energy process. In a first step, solar radiation data is converted into instantaneous solar power through a detailed source model. Then, two different machine learning approaches are used to extract meaningful features from the traces of the solar energy generation.

The work presented in this chapter has been published in the following papers:

- N. Piovesan, P. Dini, "Unsupervised Learning of Representations from Solar Energy Data", in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 9–12 September 2018, Bologna (Italy).

Chapter 6 - Optimal Direct Load Control

We introduce an optimal direct load control of renewable powered BSs in a two-tier mobile network. We introduce an optimal algorithm based on DP and we analyze the optimal policies and the achieved performance in different energy and traffic scenarios.

The work presented in this chapter has been published in the following papers:

- N. Piovesan, P. Dini, "Optimal Direct Load Control of Renewable Powered Small Cells: A Shortest Path Approach" in *Internet Technology Letters*, Wiley, 2017.

- N. Piovesan, M. Miozzo, P. Dini, "Optimal Direct Load Control of Renewable Powered Small Cells: Performance Evaluation and Bounds", in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 15-18 April 2018, Barcelona (Spain).

Chapter 7 - Optimal Direct Load Control plus Energy Sharing

We extend our analysis in Chapter 6 by considering a scenario in which the BSs can exchange energy through a micro-grid to further increase the mobile network energy efficiency. Moreover, we provide an analysis on the optimal dimensions of the harvesting devices and their influence on the performance and the monetary cost experienced by the MNOs.

The work presented in this chapter has been published in the following papers:

- N. Piovesan, D. A. Temesgene, M. Miozzo, P. Dini, "Joint Load Control and Energy Sharing for Autonomous Operation of 5G Mobile Networks in Micro-Grids", *IEEE Access*, Volume 7, 2019, Pages 31140-31150.

Chapter 8 - Online Direct Load Control plus Energy Sharing

We introduce a centralized controller, located at the MBS site, with knowledge of the overall network condition. The controller is in charge of opportunistically operating the network to achieve efficient utilization of the harvested energy and prevent SBSs blackout during periods with low renewable energy arrivals and high traffic demand. In particular, we provide a detailed analysis of three learning agents based on supervised and reinforcement learning.

The work presented in this chapter is included in the following papers:

- N. Piovesan, D. López-Pérez, M. Miozzo, P. Dini, "Joint Load Control and Energy Sharing for Renewable Powered Small Base Stations: a Machine Learning Approach", submitted to *IEEE Transactions on Green Communications and Networking*.

- N. Piovesan, M. Miozzo, P. Dini, "Modeling the Environment in Deep Reinforcement Learning: the case of Energy Harvesting Base Stations", submitted to *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Chapter 9 - Conclusions

This chapter concludes the thesis. Here, we provide a discussion on the achievements accomplished through the research presented herein, the conclusions and the perspectives for future works.

2 State of the Art

In this chapter, we review the state of the art about energy saving techniques in mobile networks. In particular, we advocate and elaborate on the use of energy harvesting hardware as a means to decrease the environmental footprint of future mobile networks technology. To take full advantage of the harvested (renewable) energy, while still meeting the quality of service required by dense mobile network deployments, suitable management techniques are here reviewed, highlighting the open issues that are still to be solved to provide eco-friendly and cost-effective mobile architectures.

The chapter is organized as follows. Energy efficiency techniques are reviewed in Section 2.1. A new network design paradigm, called energy cooperation, is described in Sections 2.2 where it is shown that network nodes can collaborate for energy self-sustainability. Our final remarks are given in Section 2.3.

2.1 Energy Efficiency Techniques

In this section, we concentrate on the techniques to reduce the energy consumption of the mobile system as a whole. Energy Efficiency (EE) is the fundamental brick of any sustainable design and defines the key methods that are to be either enhanced or brought forward when integrating energy harvesting sources. We refer to EE as a set of functions/methods conceived to reduce the energy requirement for a given level of service. EE can be quantified by the ratio between the amount of data successfully delivered (in bit/s) and the total energy spent in such transmission (in Wh or J).

Several surveys have been written to discuss on the energy efficiency of the mobile system. Sources of inefficiencies in the network are described in [20], where some potential improvements are also suggested. The authors of [64] provide an extensive description of energy-aware mechanisms at each protocol layer of the communication stack, including energy efficient hardware design principles. In this section, instead, we only concentrate on the energy efficient techniques at the network and side, which can enable an intelligent use of the harvested ambient energy and support the system self-sustainability.

The EE techniques that are exploited to decrease the energy footprint of BSs fall under two categories: 1) *sleep modes*, to selectively switch off some of the radio units (according to the traffic profile) and 2) *cell zooming*, to adapt the coverage range of BSs to cover areas where BSs are asleep and perform load balancing. These techniques are analyzed in the following.

1) Sleep modes: cellular networks are dimensioned to support traffic peaks, i.e., the number of BSs deployed in a given area should be able to provide the required Quality of Service (QoS) to the mobile subscribers during the highest load conditions. However, during off-peak periods the network may be underutilized, which leads to an inefficient use of spectrum resources and to an excessive energy consumption. For these reasons, sleep modes have been proposed to dynamically turn off some of the BSs when the traffic load is low. This has been extensively studied in the literature, considering different problem formulations [36]. As BSs cannot serve any traffic when asleep, it is important to properly tune the enter/exit time of sleep modes to avoid service outage. Moreover, when a BS is switched ON/OFF, there is an incurred energy cost that should not be ignored. This is tackled in [107] by considering BSs state transitions over time in the optimization problem, such that the overall BSs switching energy cost is minimized.

The authors of [109] propose centralized and distributed clustering algorithms to cluster those BSs exhibiting similar traffic profiles over time. Upon forming the clusters, an optimization problem is formulated to minimize their power consumption. Optimal strategies are found by brute force, since the solution space is rather small and its complete exploration is still doable. A similar approach is presented in [85] where a dynamic switching ON/OFF mechanism locally groups BSs into clusters based on location and traffic load. The optimization problem is formulated as a non-cooperative game aiming at minimizing the BS energy consumption and the time required to serve their traffic load. Simulation results show energy costs and load reductions while also provide insights of when and how the cluster-based coordination is beneficial.

User QoS is added to the optimization problem in [15]. In this case, as the problem to solve is NP-hard, the authors propose a sub-optimal, iterative and low-complexity solution. The same approach is used in [111, 94, 14, 32], playing with the trade-off between energy consumption and QoS. The Quality of Experience (QoE) is included in [108], where a DP switching algorithm is put forward. The user QoE is utilized in place of standard network measures such as delay and throughput. Other parameters that have been considered are the channel outage probability (also referred to as coverage probability), i.e., the probability of guaranteeing the service to the users located in the worst positions (e.g., at the cell edge) and the BS state stability

parameter, i.e., the number of ON/OFF state transitions. For instance, a set of BS switching patterns engineered to provide full network coverage at all times, while avoiding channel outage, is presented in [37]. The coverage probability, along with power consumption and energy efficiency metrics, are derived using stochastic geometry in [98, 42, 90]. A similar approach is considered in [48], where closed-form expressions of coverage probability and average user load are attained through stochastic geometry. Optimal resource allocation schemes are proposed to minimize power consumption and maximize coverage probability in a Heterogeneous Network (HetNet), and are validated numerically. According to the BS state stability concept, a bi-objective optimization problem is formulated in [61] and solved with two algorithms: (i) near optimal but not scalable, and (i) with low complexity, based on particle swarm optimization. The QoE is also affected by the User Equipment (UE) positions according to the channel propagation phenomena. To this respect, in [12] the BSs to be switched off are selected so as to minimize the impact on the UEs' QoE, according to their location and distance from the BSs.

In order to support sleep modes, neighboring cells must be capable of serving the traffic in areas where BSs are temporarily sleeping. To achieve this, proper *user association* strategies are required. In a scenario where sleeping techniques are not applied, each user is associated with the BS that provides the best Signal to Interference plus Noise Ratio (SINR). However, when BSs can go to sleep, user association is more complex and requires traffic prediction as well as very fast decision-making. Otherwise, users may suffer a deterioration of their QoS. A framework to characterize the performance (outage probability and spectral efficiency) of cellular systems with sleeping techniques and user association rules is proposed in [93]. In this paper, the authors devise a user association scheme where a user selects its serving BS considering the maximum expected channel access probability. This strategy is compared against the traditional maximum SINR-based user association approach and is found superior in terms of spectral efficiency when the traffic load is inhomogeneous. User association mechanisms that maximize energy efficiency in the presence of sleep modes are addressed in [112]. There, a downlink HetNet scenario is considered, where the energy efficiency is defined as the ratio between the network throughput and the total energy consumption. Since this leads to a highly complex integer optimization problem, the authors propose a Quantum particle swarm optimization algorithm to obtain a sub-optimal solution. Moreover, a problem that jointly considers energy cost and flow-level performance, such as file transfer delay, is formulated in [91]. This formulation is decomposed into two subproblems: user association and BS operation. For the user association, an optimal policy is derived, also devising a distributed implementation. For the BS operation, some low-complexity algorithms are proposed.

MNOs cooperation is exploited in [77] where a switching off strategy is implemented through a roaming cost based on user association to offload traffic and eventually defines the operational state of the BSs. Similarly, in [11] the switch ON/OFF problem for clusters of BSs has been modeled with a non-cooperative game with complete information algorithm. The game is played by the MNOs for estimating the switching-off probabilities that reduce their expected financial cost when roaming the traffic. The proposed scheme improves both the network energy efficiency and the cost. Results also provide understandings on the MNOs behavior as function of the roaming cost. An auction-based switching off solution has been proposed in [13] where MBSs owned by different MNOs can offload traffic to third party SBSs. A multi-objective auction framework has been used to opportunistically utilize the small BSs. The proposed solution considers different bidding strategies representing different levels of tolerance respect to the QoE that the MNOs want to provide to their UEs. Simulation results show improvements for throughput, energy efficiency and cost savings, providing also guidelines concerning the behaviors that the MNOs should follow in the auction. Finally, cooperation between MNOs in a C-RAN architecture is analyzed in [97]. The authors propose a novel scheme based on coalitional game theory to identify the potential room for cooperation among MNOs that provide service to the same area. Simulation results show that for the operators it is always more convenient collaborating, with profit gains ranging above 98% when compared to the stand-alone case.

2) Cell zooming: this method is also known as *cell breathing*, it is complementary to the above user association techniques and has been introduced to fill the coverage gaps that may occur as BSs go to sleep. It amounts to adjusting the cell size according to traffic conditions, leading to several benefits: (i) load balancing is achieved by transferring traffic from highly to lightly congested BSs, (ii) energy saving through sleeping strategies, (iii) user battery life and throughput enhancements [47]. To compute the right cell size, cell zooming adaptively adjust the transmitted powers, antenna tilt angles, or height of active BSs. There exists a large number of works that apply this approach to achieve energy savings in cellular networks. For instance, a cell zooming scheme, to be used in two-tier cellular networks with MBSs and SBSs, is put forward in [17]. The considered formulation entails a Capacitated Facility Location Problem (CFLP), which is known to be NP-hard. Hence, the authors provide a practical implementation allowing BSs to be smartly switched ON/OFF and filling coverage holes zooming in and out the active BSs. Further, centralized and distributed cell zooming algorithms are proposed in [75], where a cell zooming server, which can be either implemented in a centralized or distributed fashion, controls the zooming procedure by setting its parameters based on traffic load distribution, user requirements, and Channel State Information (CSI). The same server-based solution can be found in [47]. A different approach is proposed in [55], where the authors design a BS switching mechanism based on a power control

algorithm that is built upon non-cooperative game theory. A closed-form expression cell zooming factor is defined in [105], where an adaptive cell zooming scheme is devised to achieve the optimal user association. Then, a cell sleeping strategy is further applied to turn off light traffic load cells for energy saving. In general, most zooming scenarios entail a computationally intractable formulation, so affordable solutions based on iterative algorithms or heuristics abound in the literature, see, e.g., [43, 113].

Remarkably, cell zooming entails an increase in the transmitted power of the active BSs, which leads to a higher energy expenditure for the BSs that are on. However, when used in combination with sleeping strategies, this leads to additional energy savings. Some researchers are oriented towards the study of sleeping schemes in conjunction with cooperative communication strategies for distributed antennas, also referred to as *Coordinated Multi Point (CoMP)*. This technique increases spectral efficiency and cell coverage without entailing a higher BS transmitted power and reducing the co-channel interference. The authors of [18] prove the effectiveness of this approach in terms of energy and capacity efficiency when sleep modes are combined with downlink CoMP. Despite these advantages, their results also reveal that imperfect downlink channel estimations and an incorrect CoMP setup can lead to energy inefficiency. A stochastic geometry analysis is presented in [42] to evaluate the energy efficiency performance of joint sleeping and CoMP in HetNets. The authors of this paper compare the coverage probability and the energy efficiency in scenarios with and without CoMP. Their results demonstrate that the combined use of CoMP and BS sleeping techniques can improve the energy efficiency and increase the coverage probability when compared with the sole use of sleep modes.

2.1.1 Main Outcomes

The main findings are the following:

1. Grouping BSs with similar traffic patters through clustering techniques provide valuable results when applied to BS sleep modes.

2. Stochastic geometry has been vastly used to analyze the EE performance in switching ON/OFF strategies.

3. BS sleeping solutions shall be combined with other techniques such as user association, cell zooming and CoMP to ensure satisfactory network performance.

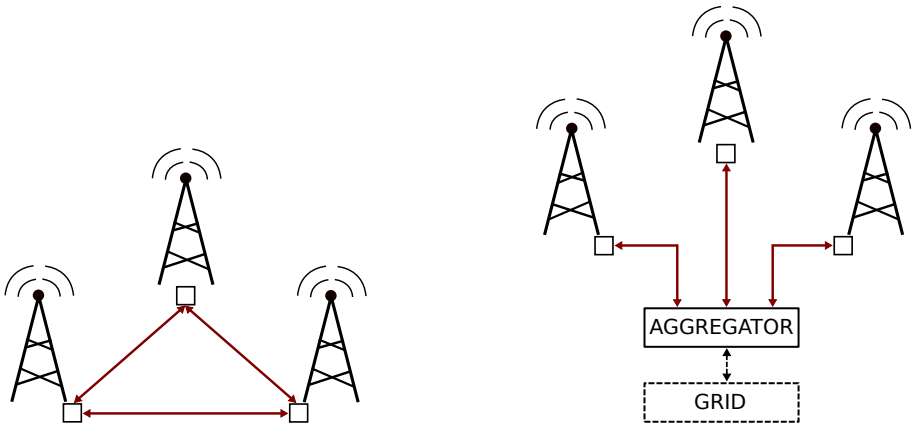
4. MNOs cooperation has been exploited through game theory and auction-based approaches with promising outcomes.

Sleeping techniques have been widely investigated for cellular networks, but there are still some open problems to be solved. In the review of the literature we noticed that the traffic models are usually over-simplified, considering uniform traffic distributions and arrival patterns in all cells at all times. However, actual network traffic is dynamic and undergoes spatial and temporal fluctuations [6] due to the movement of UEs. Hence, accurate mobility models should be inferred from real data, and used to investigate the performance of sleep modes and cell zooming. Moreover, BS switching operations are usually modeled without considering activation frequency and time. Although the most recent BSs have been conceived for frequently entering sleep modes, most of the BSs that are still in use today were designed foreseeing only occasional switch ON/OFF operations, as otherwise the failure rate of some of their parts would be too high [101]. Besides, fast switching operations can lead to a *ping-pong* effect, which occurs when the service is handed over from one cell to another, but is quickly handed back to the original cell increasing control messages to the core network, leading to an increased energy consumption and to a decreased user QoS [88]. This is more severe when there is a non-negligible BS activation time, as resources may be deactivated due to a temporary decrease in the load, and cannot be rapidly reactivated in response to a sudden increase of the same [25]. These aspects are to be taken into account to avoid service outage in real world scenarios.

2.2 Energy Cooperation

We now consider a scenario where the BSs are supplied by energy harvesters and storage devices (rechargeable batteries) and may be disconnected from the power grid (*off-grid*). There, cooperation strategies can be conceived to make them *quasi* self-sustainable, i.e., to operate mostly relying on the harvested (and stored) energy.

In this context, geographical diversity shall be exploited to mitigate the well-known temporal and spatial variability in the energy harvesting process, especially when using renewable sources such as the wind. This aspect is partially investigated in [16], where a network made of two BSs equipped with energy harvesters and some limited energy storage capability is considered. The authors propose an offline linear programming algorithm, which limits the power drained from the power grid when the energy profiles are deterministic. Furthermore, an online algorithm is put forward for a more realistic scenario where they are stochastic and not known a priori. As expected, the best results are achieved when the harvested energy profiles at the two BSs are sufficiently uncorrelated. In fact, if the amount of energy harvested is highly correlated, we have a problem when the energy inflow is little,



(a) Energy sharing through a microgrid

(b) Energy sharing through a microgrid with an aggregator

Figure 2.1: Energy sharing scenarios.

as this concurrently occurs at both BSs. When the correlation is low, it is instead very likely that one BS will experience an abundant energy inflow when the other one is in a low energy state. The former BS could then transfer some of its energy to the latter. The performance gap between the two algorithms in [16] is small, reaching the minimum value for anti-correlated energy profiles. We observe that a low correlation in the energy profiles can be more easily reached by using different renewable types, for example solar and wind, where the latter may be very useful to mitigate the shortage of energy from solar panels during the night. In the following, two cooperation types are considered:

1) **Energy sharing:** in this case, BSs are interconnected with electric wires, forming a sort of microgrid that provides mechanisms to exchange the harvested energy among the BSs. In Figure 2.1 two deployment scenarios are depicted: direct connections among BSs (Figure 2.1a) and BSs connected through an aggregator (Figure 2.1b).

2) **Communication cooperation:** BSs are not interconnected via electric cables and their cooperation involves mechanisms to support the radio communication such as power control, bandwidth control, sleep modes and traffic offloading. In this case, high-capacity mmWave backhaul connections [45] can be exploited to facilitate the deployment of *drop-and-play* devices, such as SBSs. The scenario is depicted in Figure 2.2.

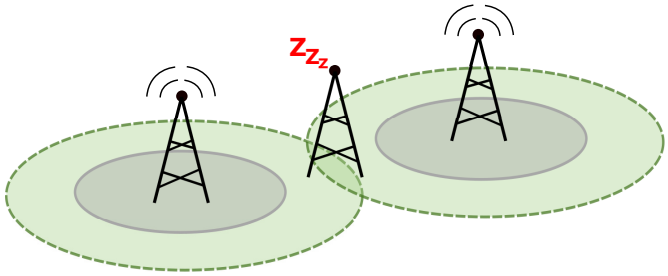


Figure 2.2: Communication cooperation scenario.

2.2.1 Energy Sharing

Energy sharing among BSs is investigated in [35] through the analysis of several basic multiuser network structures, namely, (i) an additive Gaussian two-hop relay channel with one-way energy transfer from the source to the relay node, (ii) a Gaussian two-way channel with one-way energy transfer and (iii) a two-user Gaussian multiple access channel with one-way energy transfer. A two-dimensional and directional water-filling algorithm is devised to control the harvested energy flows in both time and space (among users), with the objective of maximizing the system throughput for all the considered network configurations. The allocation algorithm is offline, relies on a priori information, i.e., the amount of energy harvested by sources and relays, and assumes unlimited data and energy buffers. However, these assumptions are unrealistic.

A very interesting energy sharing framework is presented in [31], where the concept of the Energy Packet Network (EPN) is introduced. In an EPN, discrete units of energy, termed *energy packets*, can be exchanged among network elements or acquired from the environment through harvesting hardware. Accordingly, the harvested energy can be modeled as a packet arrival process, the energy storage as a packet queue and the energy consumption process as a queue of loads, i.e., one or more servers. These three components of the EPN are interconnected thanks to power switches. Electronic systems of this type, named *power packet systems*, have been recently experimented with. In some approaches [54] the packet takes the form of a pulse of current with fixed voltage and duration. Each energy packet is equipped with an encoded header, containing the information about the destination identity (i.e., its address), which is used to route the energy packet through the EPN.

The cost of deploying the micro-grid infrastructure that would be required by an EPN can be high. In [35, 110], the use of wireless energy transfer is considered as a means to avoid the installation cost of electric cables. However, such technology has a low energy transfer efficiency nowadays, see [96, 10].

A solution to reduce the costs of deploying electrical connections between BSs is presented in [104], where a new entity named *aggregator* is introduced, as shown in Figure 2.1(b). The aggregator is in charge of mediating between the grid operator and a group of BSs to redistribute the energy flows. In [103], the authors propose an algorithm that tries to jointly optimize the transmit power allocations and the transferred energy, so as to maximize the sum-rate throughput for all the users. This joint communication and energy cooperation problem is proven to be convex. Numerical simulation shows that this approach achieves better performance than *no cooperation* or *cooperation through communication* in terms of average sum-rate.

Infrastructure sharing may be exploited to reduce power consumption by fairly distributing the harvested energy by the MNOs [76]. The problem to capture the energy interactions among MNOs is stated as a bankruptcy game. The authors focus on the fairness among operators to further motivate cooperation. The results show that all cooperative MNOs could be provided with 6 - 7 hours of operation during non-solar hours, regardless the traffic demand. Furthermore, MNOs buy grid energy at similar percentages when no green energy is available.

2.2.2 Communication Cooperation

The micro-grid deployment cost (i.e., the EPN installation cost) is one of the main aspects that motivate the introduction of this second cooperation mode. In this case, each BS has an energy harvester and may have a storage unit (*battery*), but it is not connected with the other BSs via electric cables and, in turn, cannot directly exchange energy with them, as shown in Figure 2.2. This approach eliminates the CAPEX related to the deployment of the micro-grid infrastructure (e.g., wires, converters and controllers). However, it may require harvesters and storage units with higher capacity, to achieve a certain QoS. Ongoing research aims at finding the optimal size of harvesting devices and batteries to sustain the traffic demand through the available energy budget. In particular, methods that allow the BSs to cooperatively optimize the network energy usage are proposed.

In [114], the fraction of time during which a BS cannot satisfy the traffic demand, due to energy scarcity is defined as *outage*. The authors compute the size of harvesters and batteries as a function of the outage probability. A photovoltaic panel is considered as the harvester and the size-outage region is obtained for different geographical locations. The authors conclude that full network self-sustainability may be feasible in locations with high solar irradiation, considering the cost and dimension of the energy harvesting hardware (panels and batteries). In [23], the authors define a system model of a K-tier heterogeneous cellular network, where BSs independently switch off when their energy reserve is insufficient. The authors

determine the availability region, i.e., the uncertainty in BS availability due to the finite battery capacity and to the inherent randomness in the energy harvesting process. This provides a fundamental characterization of the conditions under which standalone BSs provide the same performance as BSs relying on traditional energy sources. The introduction of sleeping capabilities in some BSs in order to reduce the size of their harvesting and storage devices is explored in [66]. In this paper, sleep modes are enabled for 50% of the BSs, when the traffic is below 50% of its peak. Although simple, this scheme allows reductions in the power consumption from 10% to 40%, depending on the sleep policy, and to reduction in the size of batteries and photovoltaic panels. However, the impact of sleep modes on the user QoS is not assessed. In [57], an optimization problem that seeks to minimize delay and power consumption by turning off small BSs is investigated. The proposed algorithm is online and is based on the so called *ski rental framework*. Each agent operates autonomously at each SBS and without having any a priori information about future energy arrivals. The algorithm is compared against a greedy scheme that uses sleep modes when the battery level is below a fixed threshold. It is shown that the proposed solution outperforms the greedy approach in terms of power consumption and network cost. The performance is evaluated assuming that energy arrivals are Poisson. This assumption is however unrealistic in most energy harvesting scenarios, as demonstrated in [72], where a stochastic Markov process has been derived for solar energy harvesting systems.

Approaches based on ML have been recently proposed as a way to schedule the switch ON/OFF of the BSs without any prior knowledge of the system and to manage the algorithm complexity. A two-tier urban cellular network is considered in [71], where MBSs are powered by the power grid and energy harvesting SBSs are deployed for capacity extension. The authors propose an algorithm based on multi-agent reinforcement learning that controls the energy spent according to the energy harvesting inflow and the traffic demand. Each node independently decides whether entering a sleep mode or serving the users within coverage. This algorithm is also shown to outperform a greedy scheme. The adoption of a distributed architecture allows to reduce the complexity of the algorithm. However, this approach suffers from a lack of coordination. In [70], this problem has been addressed by considering a layered learning algorithm based on the decomposition of the problem into two layers. The first layer is based on reinforcement learning and in charge of local control at each SBS, whereas the second layer is based on artificial neural networks and manages the network wide coordination among the SBSs.

2.2.3 Main Outcomes

The main findings of this section are described as follows:

1. Energy cooperation between BSs give better results when exploiting different types of RESs and geographical diversity.

2. Energy sharing possibilities are limited by the cost of deploying a microgrid of BSs. Some architectural solutions have been provided. In particular, the most feasible is represented by the use of an aggregator. However, EPNs represent an interesting challenge for future energy sharing deployments.

3. Cooperation between BSs avoids the deployment of a microgrid. The dimension of energy harvesting and storage devices depends on the system outage constraints and on the deployment site.

4. BSs sleeping represents one of the most promising cooperation strategies.

Energy cooperation is a recent and open field of research. Moreover, the definition of cooperation methods is crucial in case of energy self-sustainability. A key aspect is the characterization of the network load that is still not precisely captured by current analysis as already described in Section 2.1.1. We also underline the lack of performance assessments for the user perceived quality in the presence of energy cooperation mechanisms.

The harvesting process is usually characterized by very intensive power generation periods, interleaved with periods where the energy harvested is scarce of even absent. In the case of solar energy, for example, the generated power depends (among other things) on the season of the year. Since the system is designed for the worst case (e.g., winter months), the imbalance in the power generation across a full year may lead to an excess of energy during high power periods, which may be poorly handled. Investigations on an efficient use of the energy surplus shall be carried out to avoid this. The impact of energy storage devices still has to be investigated. In such a case, the adoption of energy storage leads to a higher CAPEX and the trade-off between installation cost and network performance would also have to be assessed, taking into consideration the payback period.

Most of the work cited in this section solves offline optimization problems assuming a full knowledge of energy and load patterns. This is useful as a feasibility study and to obtain performance bounds, but it is still far from the design of a practical solution. In the literature, we see an increasing interest in learning and distributed approaches for the design of online algorithms. However, these control methods are not yet mapped into the proposed 5G architecture. Concepts like network softwarization and virtualization should be included in their design and their performance should be evaluated considering real traffic (user demand) and energy harvesting traces.

Moreover, all the algorithms that have been published so far entail a zero delay when a BS transitions between active and sleep states.

Finally, a new research field addresses the design of EPNs. There, energy packets would represent a flexible and convenient method to route energy when and where needed. However, the design of power switches, as well as the definition of proper energy routing protocols, are still open research directions.

2.3 Conclusions

In this chapter, we have elaborated on the use of energy harvesting hardware as a means to decrease the environmental footprint of 5G technology. To take full advantage of the harvested (renewable) energy, while still meeting the quality of service required by dense 5G deployments, suitable management techniques have been reviewed, highlighting the open issues that are still to be solved to provide eco-friendly and cost-effective mobile networks.

While several techniques have recently been proposed to tackle capacity, coverage and efficiency problems we believe that none of these comprehensively addresses and solves the energy efficiency problem, especially in the presence of energy harvesting devices/hardware. In fact, current studies do not generally consider network elements with energy harvesting capabilities. From the analysis that we have carried out in this chapter, we have identified several open issues that range from the need for accurate energy, transmission and consumption models, to the lack of accurate data traffic profiles (from real mobility traces), to the use of energy cooperation techniques.

Specifically, energy cooperation techniques look very promising and should be better addressed, including energy harvesting and traffic dynamics. In this respect, energy packet networks are envisaged to be an interesting solution to be further explored for energy transfer among network nodes.

3 Modeling

In this chapter, we provide a description of the mathematical models adopted in this dissertation. In details, the chapter is structured as follows. In Section 3.1, we introduce the model used to convert the solar radiation into solar energy generation. In particular, we describe an astronomical model, a solar panel model and a power processor model. In Section 3.2, we describe a model for the traffic demand experienced by the network. In Section 3.3, we analyze the BS consuming components, and we introduce the adopted model. Finally, Section 3.4 concludes the chapter.

3.1 Solar Energy Harvesting Devices

Among the RESs, solar is the most popular one and is today exploited in numerous applications. Figure 3.1 shows the distribution of irradiation and thus the availability of solar power across the globe. The amount of solar energy that can be harvested mainly depends on the geographical location, the time of the day, the season, the position of the deployment and the weather. The stochastic nature of the latter and thus uncertainty of available energy is a major obstacle to the large-scale use of solar energy in many regions. An example of temporal behavior of the solar energy process is depicted in Figure 3.2 for a week of December and July.

A solar Photovoltatic (PV) array consists of one or more electrically connected PV modules, where each module contains many individual cells. When exposed to sunlight, a solar cell connected to an external circuit generates a Direct Current (DC), and using adequate components (combiners, inverters, and transformers), this current can be converted into grid-compatible Alternating Current (AC), if needed. Meanwhile, charge controllers and batteries could be also used to store energy during the day and provide on-demand power during the night.

When modeling the harvestable solar power, a good metric is the average solar radiation arriving at the surface. The radiation must then be multiplied with the solar panel size to get the received radiation. However, only a fraction of the solar radiation can be converted into electrical power for the following reasons:

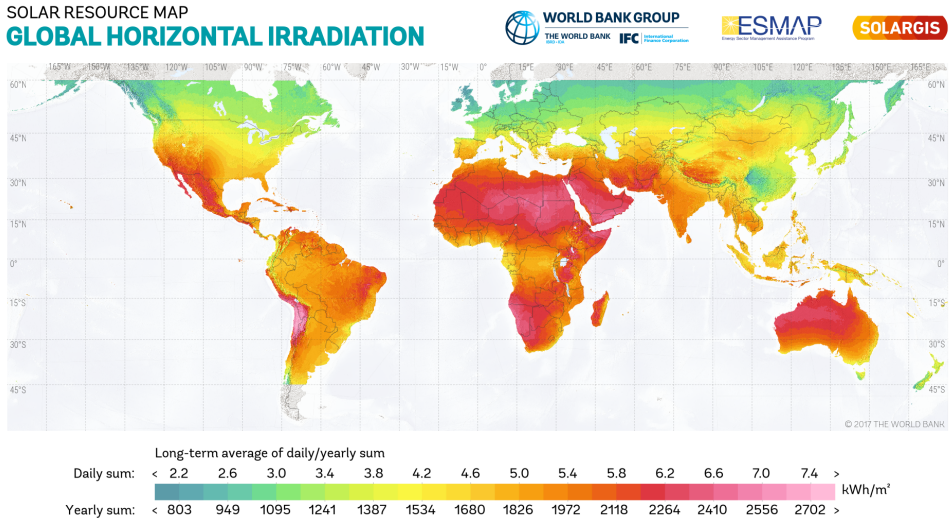


Figure 3.1: Map of average solar irradiation across the globe (©2017 The World Bank, Solar resource data: Solargis)

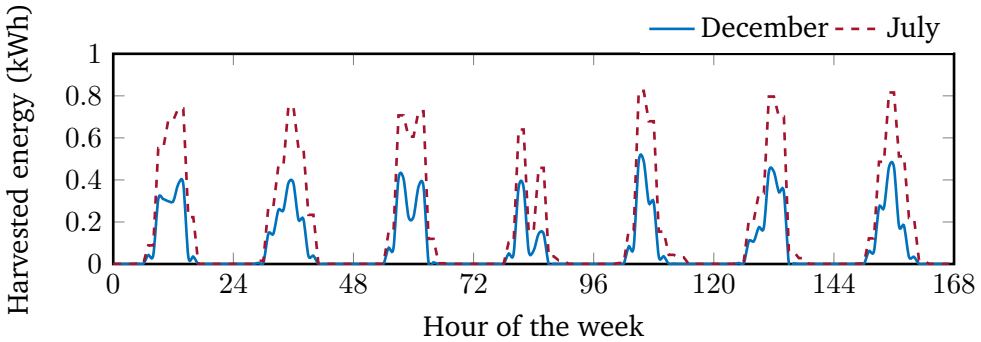


Figure 3.2: Example of temporal variation of the energy harvesting process in a week of December and July

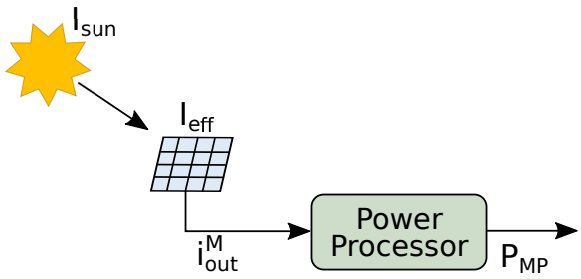


Figure 3.3: Source model used to characterize the solar energy generation process.

- each solar panel features a specific efficiency, and a reduction factor must be introduced.

- the radiation angle reduces the harvested energy: while the standard test conditions assume that the solar radiation hits the panel orthogonally, this is unrealistic for real deployments as the sun moves over the day as well as over the year. Hence, the factor that considers the angular deviation from orthogonal radiation must be included.

- if the harvested electric power is passed through a voltage regulator or used for charging a battery, losses will occur here as well.

Many models are used in the literature to describe the solar energy. The most common ones are based on linear and nonlinear models. These models give a correlation between the solar energy on a horizontal surface and some meteorological parameters such as shining hours, ambient temperature and relative humidity. The linear models use simple linear functions while the nonlinear models employ polynomial functions of higher order [52].

In the following, we describe the source model adopted in this dissertation to characterize the electricity generated by a PV panel. The key building blocks of the model are illustrated in Figure 3.3 and they are the solar source, the PV panel and the DC/DC power processor.

Astronomical Model

We define the effective solar irradiance that hits a PV panel as I_{eff}. This term depends on several factors, e.g., the inclination of the panel, the geographical location, the hour of the day and season.

In order to translate the solar irradiance, I_{sun}, into effective solar irradiance, I_{eff}, we consider the following astronomical model. According to [21], the effective solar

irradiance, I_{eff} is proportional to $\cos\Theta$, where $\Theta \in [0, 90°]$ is the angle between the sunlight and the normal to the solar module surface. It can be computed as a function of time t by considering the following formula:

$$I_{\text{eff}}(t, N) = I_{\text{sun}}(t, N) \max(0, \cos\Theta(t, N)), \tag{3.1}$$

where N is the day number in a year. We consider $N = 1$ for January 1st and $N = 365$ for December 31st (or $N = 366$ in leap years).

The value of $\cos\Theta$ is calculated as:

$$\begin{aligned}\cos\Theta(t, N) = &\sin\gamma(N) \cdot \sin La \cdot \cos\beta - \\ &- \sin\gamma(N) \cdot \cos La \cdot \sin\beta \cdot \cos\alpha + \\ &+ \cos\gamma(N) \cdot \cos La \cdot \cos\beta \cdot \cos\omega(t, N) + \\ &+ \cos\gamma(N) \cdot \sin La \cdot \sin\beta \cdot \cos\alpha \cdot \cos\omega(t, N) + \\ &+ \cos\gamma(N) \cdot \sin\beta \cdot \sin\alpha \cdot \sin\omega(t, N)\end{aligned} \tag{3.2}$$

where:

- $La \in [0, 90°]$ is the location latitude;
- Lo is the location longitude;
- γ is the declination angle;
- $\omega(t, N) \in [0, 360°]$ is the hour angle;
- β is the inclination of the solar panel towards the sun on the horizon;
- α is the azimuthal displacement, which takes values different from zero if the normal to the plane of the solar panel is not aligned with the plane of the corresponding meridian i.e., the solar panel faces East ($\alpha < 0$) or West ($\alpha > 0$).

The declination angle γ is due to the elliptic orbit of the Earth around the sun and the fact the Earth is tilted on itself at an angle of $23.45°$. It is defined as the angular distance North or South of the Earth's equator and it can be calculated as:

$$\gamma(N) \simeq \sin^{-1}[\sin(23.45°)\sin(D(N))] \tag{3.3}$$

where $D(N) = 360(N - 81)/365°$.

The hour angle ω is defined as the azimuth's angle of the sun's rays due to the Earth's rotation and it can be calculated as

$$\omega(t, N) = 15\,(\mathrm{AST}\,(t, N) - 12)^{\circ} \tag{3.4}$$

where $\mathrm{AST}(t, N) \in [0, 24]$ hour is the apparent solar time. We can calculate it as:

$$\mathrm{AST}(t, N) = t' + \Delta t + \mathrm{ET}(N) \tag{3.5}$$

where t' is the local standard time adjusted to account for the daylight saving time. Moreover, Δt is the time displacement between the selected time zone and the time at the reference Greenwich meridian. It is computed as $\Delta t = (Lo - GMA)/15^{\circ}$ where $GMA = UTC_{\mathrm{off}} \times 15^{\circ}$ is the Greenwich meridian angle and corresponds to the angle between the Greenwich meridian and the meridian of the selected time zone. Finally, UTC_{off} is the time offset between Greenwich and the time zone whereas 15 is the rotation angle of the Earth per hour. The function $\mathrm{ET}(N)$ is known as *the equation of time* and is defined as:

$$\mathrm{ET}\,(N) \simeq \frac{9.87 \sin\,(2D\,(N)) - 7.53 \cos\,(D\,(N)) - 1.5 \sin\,(D\,(N))}{60} \tag{3.6}$$

Solar Panel Model

We consider a solar panel composed of n_{sc} solar cells connected together. A number n_{p} of them are connected in parallel, whereas n_{s} are connected in series. Thus, $n_{\mathrm{sc}} = n_{\mathrm{p}} n_{\mathrm{s}}$.

The composition of the I-V curves of the solar cells allows obtaining the I-V curve used to characterize the solar panel. The I-V curve of a solar cell is given by the superposition of the current generated by the solar cell diode in the dark with the current due to the sunlight hitting the cell, defined as light-generated current i_l. We can approximate this curve as:

$$i_{\mathrm{out}} \simeq i_l - i_{\mathrm{o}} \left[\exp\left(\frac{qv}{nkT} \right) - 1 \right] \tag{3.7}$$

where q is the elementary charge, v is the cell voltage, k is the Boltzmann's constant, T is the temperature in Kelvin degrees, $n \geq 1$ is the diode ideal factor. Finally, i_{o} is the dark saturation current and corresponds to the diode leakage current when there is not light. It depends on the area and the technology of the solar cell.

We define i_{sc} as the short circuit current, which corresponds to the maximum current for the cell, which occurs when the voltage across the cell is zero. We also define v_{oc}

as the open circuit voltage, which corresponds to the maximum voltage for the cell, which occurs when the net current through the device is zero.

We can normalize the effective irradiance, I_{eff} with respect to the maximum radiation of 1 kW/m^2, obtaining the radiation rate $F(t, N) = 0.001 \cdot I_{\text{eff}}(t, N)$. Then, we can compute the light-generated current for a single solar cells as $i_l(t, N) = i_{\text{sc}} F(t, N)$ and obtain $i_{\text{out}}(t, N)$ for a single solar cells using equation (3.7). Finally, the total current generated by the solar module is $i^{\text{M}}_{\text{out}}(t, N) = n_{\text{p}} i_{\text{out}}(t, N)$.

Power Processor Model

Every voltage or current source has a maximum power point, at which the average power delivered to its load is maximized. In general, the load of a device does not match the optimal one, required to extract the maximum power from the solar source. To solve this problem, a power processor is used to emulate the optimal load by adjusting the source voltage until the power extracted from it is maximized.

In order to account for the DC/DC power processor, we have computed the operating point $(i^{\text{M}}_{\text{out}}, v^{\text{M}})$ for which the extracted power $P = i^{\text{M}}_{\text{out}} v^{\text{M}}$ is maximized.

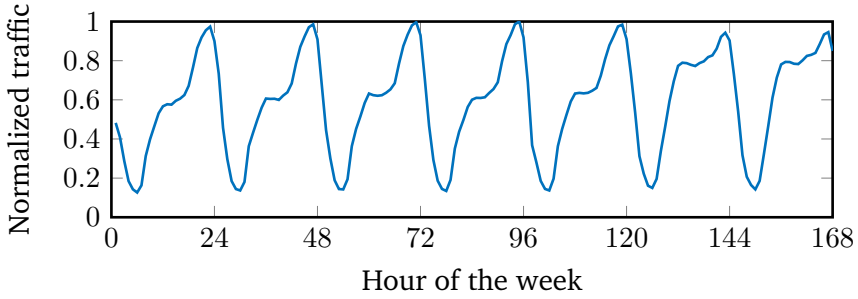
In this phase, we have considered the parameters described in the astronomical model (e.g., solar irradiance, Earth rotation, etc.) and the fact that i_{sc} and v_{oc} depends also on the environmental temperature. Then, we have obtained the maximum power P_{MP} as

$$P_{\text{MP}} = \eta \max_{v} \left\{ i^{\text{M}}_{\text{out}}, v^{\text{M}} \right\} = \eta \cdot n_{\text{p}} n_{\text{s}} \max_{v} \left\{ i_{\text{out}} v \right\} \tag{3.8}$$

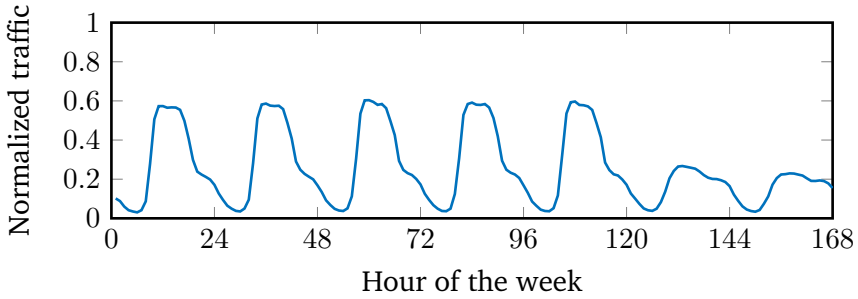
where i_{out} is given by equation (3.7) and $\eta \in (0, 1)$ is the power processor conversion efficiency.

3.2 Traffic Demand

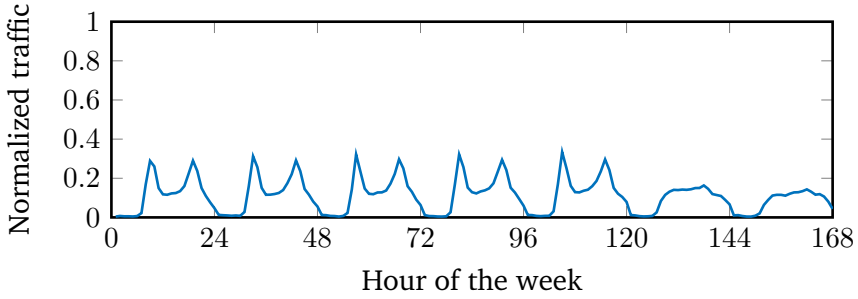
In order to provide a realistic analysis of the energy efficiency of a mobile network, it is essential to properly model the traffic demand that should be served by the network. In particular, it is fundamental to model how the demand of traffic varies during the day.



(a) Residential area



(b) Office area



(c) Transportation area

Figure 3.4: Example of temporal variation of the traffic process in different areas.

According to the analysis done in [6], the daily variation of the number of active users in analogous to the daily variation of the traffic and 16% of the data subscribers are active during the busy/peak hours. Moreover, significant differences are observed between users when comparing their traffic demand. The users are therefore classified in two types, ordinary and heavy, according to their demand of traffic. Ordinary users demand for 112.5 MB/h, whereas heavy users demand for 900 MB/h.

The variation of both the number of active users and the traffic per active user is modeled by the traffic profile function $\vartheta(t)$. Therefore, the average amount of traffic requested to a BSs at time t is defined as:

$$T_{\mathrm{BS}}(t) = N_{\mathrm{UE}} \cdot \vartheta(t) \cdot [(1 - r_h) \cdot 112.5 \text{ MB/h} + r_h \cdot 900 \text{ MB/h}] , \tag{3.9}$$

where N_{UE} is the number of users positioned in the BS coverage area and r_h is the ratio of heavy users.

Different scenarios can be constructed by adjusting the ratio of the heavy users. A scenario with $r_h = 20\%$ of heavy users was the most relevant European scenario for 2015 according to [28]. A scenario with $r_h = 50\%$ is adopted for the current days, whereas $r_h = 100\%$ is considered as an extremity for very high data rates in future networks.

The traffic profile function $\vartheta(t)$ has been designed by considering real traffic profiles described in [102], which are derived combining time, location and frequency information of thousands of cellular towers. Moreover, traffic variability is added following a normal distribution with zero mean and standard deviation derived from the measurements of real mobile traffic traces [95]. The analysis provided in [102] demonstrates that the urban mobile traffic usage can be described by five basic time domain patterns that corresponds to functional regions, i.e., residential, office, transportation, entertainment and comprehensive. In this dissertation, we mainly focus on residential, office and transportation profiles, being them the most traffic intensive in metropolitan areas. An example of their traffic profiles is shown in Figure 3.4. We can notice that both profiles present a high activity during the day and lower during night. However, the highest amount of traffic is concentrated in daylight hours (i.e., from 10 am to 6 pm) for the office profile, whereas the residential profile has a peak during the evening hours (i.e., from 6 pm to 12 pm). The transportation profile has two peaks during the weekdays, at 8 am and 6 pm, whereas on the weekends it has a single peak at 5 pm.

3.3 Energy Consumption

BSs can be classified into two main groups, depending on transmission power and coverage range.

1) **Macro BS (MBS)**: with transmission power of about 40 W for devices with bandwidth of 20 MHz and 80 W for LTE-A devices with 40 MHz [1]. Their communication range reaches up to a few kilometers and they are usually installed on building rooftops.

2) **Small BS (SBS)**: with transmission power ranging between 0.05 W and 6 W. They can be further classified into micro, pico and femto BSs. Micro and pico BSs cover small to medium areas with dense traffic (*hotspots*) such as shopping malls, residential areas, such as hotels or train stations. The typical range of a micro/pico BS spans from a few hundred meters up to one kilometer. Femto cells are designed to serve smaller areas such as private homes or indoor spaces. The range of femto cells is typically only a few meters, and they are generally wired to a private cable broadband connection or to a home digital subscriber line [38]. SBSs can be installed in street furniture like lampposts or traffic lights due to their small form factor.

The power consumption at full system load of the different types of BSs can range from about 6 W for a femto BS to 1 kW for a MBS [6, 22, 84]. Typically, this power consumption is modeled as the sum of a static value and a dynamic and load-dependent value [7, 63]:

$$P_{\text{BS}} = \begin{cases} N_{\text{TRX}} \cdot (P_0 + \alpha P_{\text{out}}), & 0 < P_{\text{out}} \leq P_{\max} \\ N_{\text{TRX}} \cdot P_{\text{sleep}}, & P_{\text{out}} = 0 \end{cases} \tag{3.10}$$

where N_{TRX} is the number of transmit/receive chains, P_0 is the BS power consumption at zero Radio Frequency (RF) output power, α is the slope of the load dependent power consumption curve, P_{out} is the load-dependent part of the RF output power and $P_{\max}$ is the value of P_{out} at maximum load.

Table 3.1 specifies the load dependencies of the different BS types [6]. The power consumed by a MBS increases much more with the traffic load than that of a SBS. This is due to the high consuming power amplifier that MBSs use to cover wide areas, whereas SBSs need amplifier designs for much lower coverage and, consequently, lower energy consumption figures. Remarkably, P_0 represents a significant part of the total energy consumed by any BS and, due to this, researchers have investigated

Table 3.1: Power model parameters for different types of BS (from [6])

BS type	N_{TRX}	$P_{\max}[\text{W}]$	$P_0[\text{W}]$	α	$P_{\text{sleep}}[\text{W}]$
Macro	6	40.0	130.0	4.7	75.0
RRH	6	20.0	84.0	2.8	56.0
Micro	2	6.3	56.0	2.6	39.0
Pico	2	0.13	6.8	4.0	4.3
Femto	2	0.05	4.8	8.0	2.9

the use of sleep modes during low traffic periods. Moreover, it is expected that P_0 and P_{sleep} of new sites will be reduced by about 8% on average thanks to recent technological advances [84], thus further decreasing the BS energy cost during low traffic periods.

In Figure 3.5, we compare the energy drained by the various parts of MBSs and SBSs. According to [6], the power amplifier of a MBS dominates the total power consumption. For SBSs, the baseband processor has a higher impact. Gathering the baseband units of different BSs in a centralized pool, as done in C-RAN systems, may reduce the network energy consumption.

3.4 Conclusions

In this chapter, we have introduced the system models adopted in this dissertation. Astronomical, photovoltaic and power processor models allows to compute the solar energy generation in a given location starting from collected measurements of the solar radiation. Moreover, a model of the traffic demand has been introduced. This models takes into consideration the different kind of users served by a mobile network and their hourly traffic generation. Different traffic areas are taken into account, making this model very useful for understanding how the approaches discussed in this dissertation react to different types of traffic. Finally, the power consumption of the BSs has been described and a state-of-the-art model has been introduced.

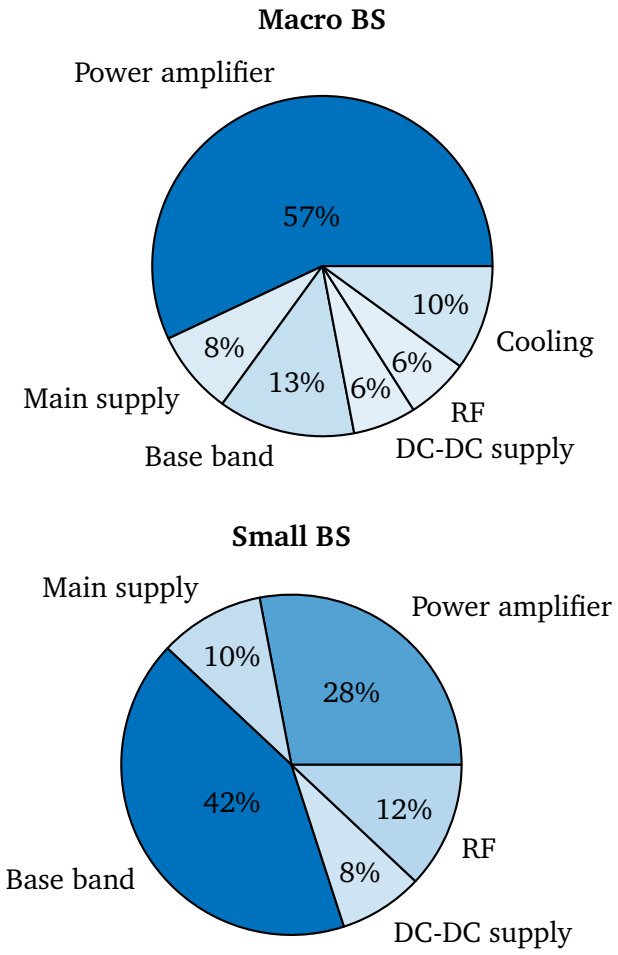


Figure 3.5: Comparison of BS energy consumption figures [6].

4 Mathematical Framework

In this chapter, we provide an introduction on the mathematical framework adopted in this dissertation.

DP has been adopted as an efficient mathematical tool to solve complicated problems by recursively breaking them down into simpler sub-problems. In particular, we consider the case of deterministic finite state system which can be represented by graphs. Therefore, the optimal control problem can be translated into a shortest-path search and solved by using graph theory.

ML has been considered as an efficient mathematical tool for solving complex problems. Supervised learning has been used to perform data classification, unsupervised learning to extract features from data, and Reinforcement Learning (RL) for implementing agents able to learn by interacting with the environment. In particular, we analyze classic RL algorithms based on tabular methods, in which the discrete learning space is small enough to be represented as a table, and Deep RL methods which adopt Artificial Neural Network (ANN) as function approximation to deal with bigger learning spaces.

The chapter is organized as follows. In Section 4.1, we introduce DP and the shortest-path algorithm. In Section 4.2, we discuss ML methods, by focusing on ANNs, dimensionality reduction algorithms and RL. Finally, Section 4.3 concludes the chapter.

4.1 Dynamic Programming

DP is a mathematical optimization method developed mainly by Richard Bellman in the 1950s, based on the simplification of complicated problems by recursively breaking them down into simpler sub-problems.

We consider situations in which decisions are taken in stages and the output of a decision can be anticipated before the next decision is made. The objective is to minimize a cost. The key aspect of this situation is that the decisions cannot be isolated since we want to balance the desire for low present cost with undesired

future high costs. The trade-off between those two aspect is captured by the DP algorithm. At each stage, it ranks decisions based on the sum of the present cost and the expected future ones, assuming optimal decision-making for the subsequent stages. The considered model has two main features: (i) a discrete-time dynamic system, and (ii) a cost function that is additive over time. The dynamic system expresses the evolution of some variables (i.e., the state of the system) under the influence of the decisions that are made at discrete instances of time.

The system is defined as:

$$x_{t+1} = f_t\left(x_t, u_t, w_t\right), \quad t = 0, 1, \ldots, N-1 \tag{4.1}$$

where t is the discrete time index, $x_t \in S_t$ is the state of the system, $u_t \in C_t$ is the control decision variable selected at time t, $w_t \in D_t$ is a random parameter (often defined as noise or disturbance), N is the horizon (i.e., the number of times the control is applied). Finally, f_t is a function that describe the system and in particular how the states are updated, whereas the cost incurred at time t is denoted by the function $g_t\left(x_t, u_t, w_t\right)$.

The control u_t is constrained to take values in a given non-empty subset $U(x_t) \subset C_t$, that depends on the current state x_t; that is $u_t \in U_t(x_t), \forall x_t \in S_t, \forall t$. The random disturbance w_t has a probability distribution $P_t(\cdot|X_t, u_t)$ that does not depend on values of prior disturbances $w_{t-1}, \ldots, w_0$. We consider the class of policies consisting of a sequence of functions

$$\pi = \{\mu_0, \ldots, \mu_{N-1}\} \tag{4.2}$$

where μ_t maps states x_t into controls $\mu_t\left(x_t\right)$ and is such that $\mu_t\left(x_t\right) \in U_t\left(x_t\right), \forall x_t \in S_t$. Those policies are defined as *admissible*.

Given the initial state x_0 and the admissible policy π, the states x_t and disturbance w_t are random variables with distributions defined trough the system of equations $x_{t+1} = f_t\left(x_t, \mu_t\left(x_t\right), w_t\right), t = 0, 1, \ldots, N-1$.

Therefore, for given cost functions g_t, $t = 0, 1, ..., N$, the expected cost of π starting at x_0 is

$$J_\pi(x_0) = \mathbb{E}\left\{g_N\left(x_N\right) + \sum_{t=0}^{N-1} g_t\left(x_t, u_t(x_t), w_t\right)\right\} \tag{4.3}$$

where the expectation is taken over w_t and x_t. We define the optimal policy π^* as the one that minimize this cost, that is

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \tag{4.4}$$

where Π is the set of admissible policies. We highlight that the optimal policy π^* depends on the fixed initial state x_0. However, it is typically possible to find a policy π that is optimal for all the initial states. We define the optimal cost function as

$$J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \tag{4.5}$$

4.1.1 The Dynamic Programming Algorithm

Let's now consider the problem introduced in the previous section. For every initial state x_0, the optimal cost $J^*(x_0)$ of the problem is equal to $J_0(x_0)$, given by the last step of the following algorithm, which proceeds backward in time from stage $N-1$ to stage 0:

$$\begin{aligned} J_N(x_N) &= g_N(x_N), \\ J_t(x_t) &= \min_{u_t \in U_t(x_t)} \mathbb{E}\left[g_t(x_t, u_t, w_t) + J_{t+1}(f_t(x_t, u_t, w_t))\right], t = 0, 1, \ldots, N-1 \end{aligned} \tag{4.6}$$

where the expectation is taken with respect to the probability distribution of the disturbance w_t, which depends on x_t and u_t. Moreover, if the control $u_t^* = \mu_t^*(x_t)$ minimizes the right side of the second equation in (4.6) for each state x_t and stage t, then the policy $\pi^* = \left\{\mu_0^*, \ldots, \mu_{N-1}^*\right\}$ is optimal. A proof of this is reported in [9].

4.1.2 Finite State Systems and Shortest Paths

In this section, we consider a deterministic problem where each disturbance w_t can take only one value and where the state space S_t is a finite space for each t. At any state x_t, a control u_t can be associated with a transition from the state x_t to the state $f_t(x_t, u_t)$ at a cost $g_t(x_t, u_t)$. A finite-state deterministic problem can be represented by a graph such as the one in Figure 4.1. Specifically, each arc correspond to transitions between states at successive stages and each arc as an associated cost.

A single terminal node as been added at time $T = N + 1$ to handle the final stage. Every state x_N at stage N is connected to the terminal node T with an arc of cost $g_N(x_N)$. The control sequences are represented by paths going from the initial state s at stage 0 to one of the final states at stage N. If we view the cost of an arc as its length, the deterministic finite-state problem is equivalent to finding the shortest path from the initial node s to the terminal node T. We define a path as a sequence of arcs of the form $(j_1, j_2), (j_2, j_3), \ldots, (j_{t-1}, j_t)$ whereas the length of a path corresponds to the sum of the lengths of its constituting arcs.

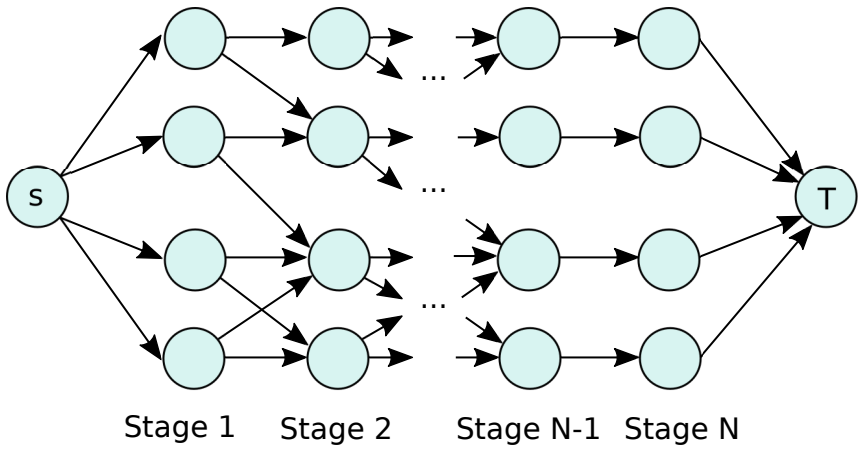


Figure 4.1: Transition graph for a deterministic finite-state system. The states are represented by the nodes, whereas the transitions between states are represented by the arcs.

We define a_{ij}^t as the cost of the transition from the state $i \in S_t$ to the state $j \in S_{t+1}$ at stage t, whereas a_{iT}^N is the terminal cost of state $i \in S_N$. Moreover, if there is not a control that moves the state from state i to state j at stage t we adopt the convention $a_{ij}^t = \infty$ (i.e., infinite arc length).

The dynamic programming algorithm has the following formulation:

$$\begin{aligned} J_N(i) &= a_{iT}, \quad i \in S_N \\ J_t(i) &= \min_{j \in S_{t+1}} \left[a_{ij}^t + J_{t+1}(j) \right], \quad i \in S_t, \quad t = 0, 1, \ldots, N-1 \end{aligned} \tag{4.7}$$

The shortest path from s to T and therefore the optimal cost is $J_0(s)$.

4.1.3 Shortest Path Algorithms

We have seen that deterministic finite-state optimal control problems and shortest path problems are equivalent. This fact have two implications:

1. DP can be used to solve shortest path problems. In practice, it is preferred when dealing with acyclic graph structures and also when parallel computation is available;

2. shortest path methods can be used to solve deterministic finite-state problems.

We focus on shortest path problems with a large number of nodes and a single initial and terminal node (as the ones arising from deterministic optimal control). Since the graph contains a large number of nodes, it is often true that most of them are

unlikely candidates for inclusion in the shortest path between origin and destination. However, in the DP algorithm every node and arc participate in the computation. This fact enables the possibility of looking to more efficient methods, based on the shortest path algorithm, that allow finding the solution by discarding portions of the graph.

In the following, w¨e define the origin node of the path as s and the destination node as T. We assume the destination to be unique. A node j is called child of the node i if there is an arc (i, j) connecting i to j. The length of the arc (i, j) is a_{ij} with $a_{ij} \geq 0$.

Label Correcting Methods

The idea at the base of label correcting methods is to progressively discover shortest paths from the origin node s to every node i. The length of the shortest path to node i found so far is stored in a variable d_i named *label of* i.

The label of the origin node s is set and maintained to $d_s = 0$ whereas the labels of the other nodes are initialized to ∞ (i.e., $d_i = \infty, i \neq s$). Moreover, the label of the destination node T is stored in a variable named UPPER. The value of UPPER is equal to ∞ if no paths to T have been discovered so far, otherwise the value of UPPER corresponds to the length of some path from s to T. Consequently, the value of UPPER represents an upper bound on the shortest distance from s to T. The nodes candidate for further examination by the algorithm to be included in the shortest path, are maintained in the list OPEN. That list initially contains only the node s and other nodes are included while the graph is explored.

When a shortest path to node i is found, the variable d_i is reduced and the algorithm checks if the labels d_j of the children j of i can be *corrected* by setting them to $d_i + a_{ij}$ (i.e., the length of the updated shortest path from s to i plus the length of the arc (i, j)). Furthermore, the node j enters the OPEN list. In this way, the paths passing through j and reaching the children of j can be taken in account in the following iterations of the algorithm. In particular, this is done only when the considered path has a chance of leading to a path from s to T which is shorter than the actual upper bound UPPER. Since the arcs only have non-negative lengths, this is true only if $d_i + a_{ij} < UPPER$.

The label of j is equal to ∞ if the node has not yet entered the OPEN list, or else it is equal to the length of some path from s to j consisting of the nodes that have already entered the list. The path can be constructed by tracing backward the parent

nodes starting from the node j. In particular, when the algorithm terminates, the shortest path can be obtained by tracing back the parent nodes from node T towards node s.

It is proved that if there exists at least one path from the origin to the destination, the label correcting algorithm terminates with UPPER equal to the shortest path from the origin to the destination. Otherwise, the algorithm terminates with UPPER=∞ [9]. The steps of the label correcting method are detailed in Algorithm 1.

Algorithm 1 Label Correcting Algorithm

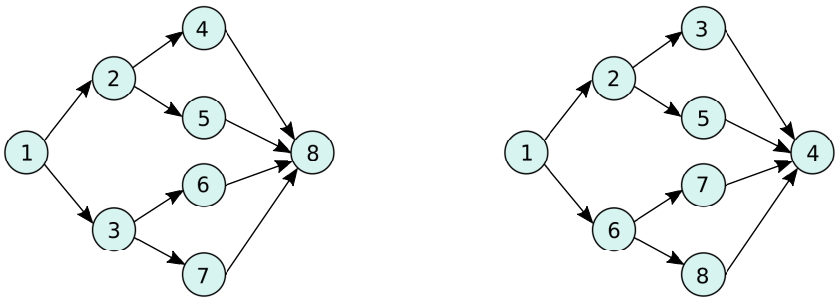
```
Place s in OPEN
while OPEN is not empty do
  Remove a node i from OPEN
  for each child j of i do
    if d_i + a_ij < min(d_j, UPPER) then
      set d_j = d_i + a_ij
      set i to be parent of j
      if j ≠ T and j not in OPEN then
        place j in OPEN
      end if
      if j = T then
        set UPPER = d_i + a_iT
      end if
    end if
  end for
end while
```

Regarding the examination of the nodes in the OPEN list, there is not a specific rule to be followed when deciding which node to extract. This gives rise to several methods. In particular, the principals are:

1. **Breadth-first search:** it adopts a first-in-first-out policy, as shown in Figure 4.2(a). The node is removed from the top of the list and each node entering the list is placed at the bottom.

2. **Depth-first search:** it adopts a last-in-first-out policy, as shown in Figure 4.2(b). The node is removed from the top of the list and each node entering the list is placed at the top. This method requires little memory and allows to have an initial estimate of UPPER than can be used to cut some of the paths, often enabling a faster convergence to the shortest path.

3. **Best-first search:** at each iteration removes from OPEN a node with minimum value of label, i.e., the node i with $d_i = \min_{j \in \text{OPEN}} d_j$. Following this method, a node will enter the list at most once. However, the overhead required to



(a) Breath first search (b) Depth-first search

Figure 4.2: Searching a tree in breath-first (a) and depth-first (b) fashion. The numbers inside the nodes indicate the order in which nodes are extracted from the OPEN list.

find at each iteration the node with minimum label represents a significant drawback to its application.

4.2 Machine Learning

ML is a computer science field that evolved from the study of pattern recognition and computational learning in artificial intelligence. The goal of ML is to study algorithms that operates by building a model from the input data in order to execute *data-driven* tasks (e.g., prediction, classification, clustering or decision) instead of following static program instructions.

The tasks of ML algorithms can be classified into three main categories, depending on the nature of the learning signal available to the learning system:

- **Supervised learning:** the learning agent is trained with example inputs and their desired outputs. The goal is to learn a general rule to map the inputs to the outputs.

- **Unsupervised learning:** the learning agent is trained with inputs and no information about the output is given. In this way, the agent goal is to find structures in the input data. This type of learning can be a goal in itself (discovering hidden patterns) or a means towards an end.

- **Reinforcement learning:** the learning agent interacts with a dynamic environment in which it must perform a certain goal without having a teacher telling it whether it has come close or not to its goal.

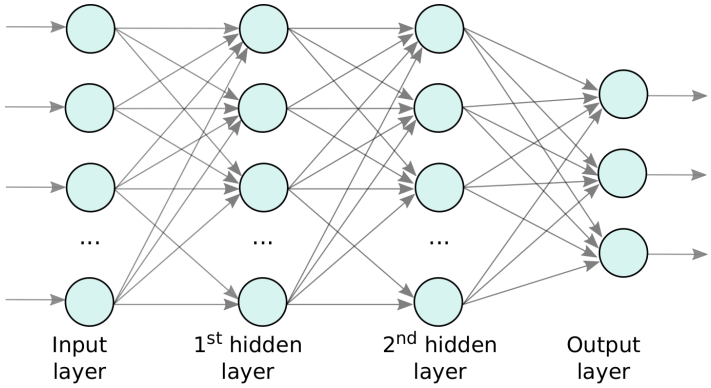


Figure 4.3: Architectural graph of an ANN with two hidden layers

Another classification of ML methods can be provided by considering the output of the learning agent.

- **Classification:** the inputs are divided into classes and the agent must learn a model that assigns unseen inputs to one or more of those classes. This is typically a supervised problem.

- **Regression:** it is a supervised problem. The outputs are continuous rather than discrete.

- **Clustering:** the inputs are divided into groups. Unlike in classification, the groups are not known in advance, making it an unsupervised problem.

- **Density estimation:** the problem consists on finding the distribution of the inputs in some space.

- **Dimensionality reduction:** the inputs are simplified by mapping them into a lower dimensional space.

All of the three main categories of ML algorithms are involved within this thesis work. Supervised learning have been used in Section 8.2.1 to develop a learning agent based on the imitation of an expert behavior. Unsupervised learning have been used in Chapter 5 to perform dimensionality reduction and clustering. Finally, reinforcement learning has been used in Section 8.2.2 to design agents able to learn operative policies to reduce the network energy consumption by interacting with it. In the following sections we describe some of these ML models, by focusing on those that have been applied to this thesis work.

4.2.1 Artificial Neural Networks

ANNs are a family of statistical learning models inspired by biological neural networks. These models are typically used to estimate or approximate functions that can depend on a large number of inputs. ANNs are typically represented as a system of interconnected neurons that exchange messages each other. The connections between neurons have numeric weights that are tuned based on experience, enabling the ANN to adapt to the inputs and learn. The interconnection between the artificial nodes forms a network that mimics a biological neural network. However, there is not a single formal definition of what a ANN is. A class of statistical models is commonly called "neural" if it has the following characteristics:

1. adaptive weights, i.e., numerical parameters tuned by the learning algorithm;

2. capability of approximating non-linear functions of the ANN inputs.

An example of ANN is depicted in Figure 4.3. The ANN is composed of three layers on neurons. The first layer has input neurons which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex networks have increased number of neurons and layers. The synapses store parameters named *weights* that manipulate the data in the calculations.

In the following, we consider *feed-forward ANNs*, wherein connections between nodes do not form cycles. The information moves from the input nodes, through the hidden nodes and finally to the output nodes. In particular, we consider the multylayer perceptron model, which is characterized by three main aspects:

- **Activation function.** Each neuron model includes a non-linear activation function. The non-linearity of this function allows the ANN to learn non-linear models. A commonly used activation function is the sigmoid function, defined as $f(v_j) = \frac{1}{1+e^{-v_j}}$, where v_j is the induced local field of neuron j (i.e., the sum of all the synaptic inputs plus the bias).

- **Hidden neurons.** The ANN includes layers of hidden neurons that allows the ANN to extract meaningful features from the input vectors and solve complicated tasks.

- **Connectivity.** The ANN has a high degree of connectivity expressed by the synapses.

The Back-propagation Algorithm

The back-propagation learning consists on two passes through the layers of the ANN. In the *forward pass*, the input vector is applied to the input neurons and its effect propagates through the ANN, layer by layer, until a set of output is produced. During the forward pass the weights of the ANN synaptic are fixed. The output of the ANN is compared with the target output and an error is computed. In the *backward pass*, the synaptic weights are adjusted in accordance with an error-correction rule. The error is propagated backward in the ANN and the weights of the synaptic are adjusted to make the actual output move closer to the target output.

The set containing N examples is shown to the ANN during the training phase. This set is named *training set* and it is defined as $\{x(n), d(n\}_{n=1}^{N}$, where $d(n)$ is the target output associated to the input $x(n)$. During iteration n, the n-th training example is presented to the network. We refer to the i-th element of the input vector as $x_i(n)$, whereas we define the k-th element of the overall output vector as $o_k(n)$. Moreover, the bias applied to neuron j is denoted by b_j. The effect of the bias is represented by a synapse of weight $w_{j0} = b_j$ connected to a fixed input equal to 1.

We define the instantaneous value of the total error energy at iteration n (i.e., presentation of the n-th example) as

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \tag{4.8}$$

where C is the set of the neurons in the output layer and e_j is the error at the output neuron j, computed as

$$e_j(n) = d_j(n) - y_j(n) \tag{4.9}$$

where $d_j(n)$ is the target output of neuron j and $y_j(n)$ is the actual output of neuron j.

The overall goal of the training process is to minimize the average errors computed in the training set by adjusting the values of the synaptic weights. The adjustment is made with respect to the errors computed for every input pattern presented to the network.

Forward pass. The induced local field $v_j(n)$ produced at the input of the neuron j is computed as

$$v_j(n) = \sum_{i=0}^{m} w_{ij}(n) y_i(n) \tag{4.10}$$

where m is the total number of inputs to the node j. Moreover, for the ease of presentation, we define $w_{j0} = b_j$. Therefore, the output of the neuron j can be calculated as

$$y_j = f_j(v_j(n)) \tag{4.11}$$

where $f_j(\cdot)$ is the activation function adopted at neuron j.

If the node j is in the first hidden layer then the index i refers to the i-th input of the neural network so we write $y_i(n) = x_i(n)$ where $x_i(n)$ is the i-th element of the input vector. On the contrary, if the index j is in the output layer of the neural network then the index j refers to the j-th output neuron of the network, thus it is $y_j(n) = o_j(n)$ where $o_j(n)$ is the j-th element of the output vector. This value is then compared with the target output $d_j(n)$ to compute the error $e_j(n)$.

Backward pass. At this point, it is fundamental to know how the adjustment of the weights affect the error. To compute the partial derivative $\partial E(n)/\partial w_{ij}$ we apply the chain rule of calculus

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ij}(n)} = -e_j(n) f_j'(v_j(n)) y_i(n) \tag{4.12}$$

where the result is easily obtained by applying the partial derivatives to Equations (4.8)-(4.11).

The value of w_{ij} is adjusted by a correction $\Delta w_{ij}(n)$ according to the delta rule

$$\Delta w_{ij}(n) = -\alpha \frac{\partial E(n)}{\partial w_{ij}(n)} = \alpha \sigma_j(n) y_i(n) \tag{4.13}$$

where α is the learning rate and the local gradient $\sigma_j(n)$ is defined as

$$\sigma_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) f_j'(v_j(n)) \tag{4.14}$$

We highlight here that the delta rule has a very intuitive rationale. If the error goes down when the weight increases ($\frac{\partial E(n)}{\partial w_{ij}} < 0$) then it makes sense to continue increasing it. Otherwise, if the error increases ($\frac{\partial E(n)}{\partial w_{ij}} > 0$) then the weight value must be decreased.

The delta rule equation indicates that the error $e_j(n)$ at every node j needs to be known to calculate the weight adjustment Δw_{ij}. At this point we need to distinguish between two cases. If the neuron is located in the output layer, the error $e_j(n)$ can be simply computed by using Equation (4.9) since both the target output and the actual output are known. On the contrary, if the node j is located in a hidden layer, there is not a specified target output for that neuron. Therefore, the error for a hidden

neuron needs to be determined recursively in terms of errors of all the neurons to which that hidden neuron is connected.

In the following, for ease of presentation, we use j to refer to a hidden node and k for an output node (or a hidden neuron in the immediate right of j). The local gradient $\sigma_j(n)$ for a hidden neuron j can be rewritten as

$$\sigma_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial y_j(n)} f'_j(v_j(n)) \tag{4.15}$$

The computation begins from the total error introduced in Equation (4.8). We rewrite it by substituting the index with k in order to avoid confusion with the index adopted for hidden neurons. Therefore, we have

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \tag{4.16}$$

By differentiating the total error with respect to the neuron output $y_j(n)$ we obtain

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_{k \in C} e_k \frac{\partial e_k(n)}{\partial y_j(n)} = \sum_{k \in C} e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} = -\sum_{k \in C} e_k(n) f'_k(v_k(n)) w_{jk}(n) \tag{4.17}$$

where the last step is computed by considering the definition of error in Equation (4.9) and the definition of local field given in Equation (4.10). The back-propagation formula for the local gradient $\sigma_j(n)$ is then obtained by substituting Equation (4.17) in Equation (4.15):

$$\sigma_j(n) = f'_j(v_j(n)) \sum_{k \in C} \sigma_k(n) w_{jk}(n) \tag{4.18}$$

Therefore, when computing the weight adjustment Δw_{ij} (as defined in Equation (4.13)) we apply the following rule: if neuron j is an output node, $\sigma_j(n)$ is computed as in Equation (4.14) whereas if the node j is a hidden node, $\sigma_j(n)$ is computed as in Equation (4.18).

4.2.2 Dimensionality Reduction

The original data dimensionality corresponds to the number of variables that are measured on each observation. In many cases, when dealing with high-dimensional data, the observation variables are redundant and correlated. Therefore, it may be that only a small subspace of the original data space is populated by the sample [86].

Dimensionality reduction is the process of reducing the number of variables representing the observations (i.e., representations or features) with a minimal information loss. This process allows removing irrelevant and redundant data, increasing learning accuracy, and improving the comprehensibility of the results [51].

The reduction of the data dimensionality can be achieved in two ways:

- **Features selection:** the relevant variables are selected from the original data;
- **Features extraction:** the data redundancy is exploited to find a smaller set of new variables which are a linear combination of the representations variables. These new variables contain basically the same information of the original variables.

Recently, feature extraction has received significant attention as a highly effective alternative to conventional feature sets handcrafted by a domain expert [8]. These techniques have been shown to be superior to feature engineering for a plethora of tasks, including speech recognition, music transcription, audio and video recognition [8, 29].

In the following sections, we discuss two techniques to perform dimensionality reduction through feature extraction. The first is based on Principal Component Analysis (PCA) whereas the second is based on a specific ANN architecture named *Autoencoder*.

Principal Component Analysis (PCA)

PCA is a non-parametric technique for extracting relevant features from a dataset of representations. The purpose is to reduce the dimensionality of the dataset by finding a new set of variables, smaller than the original, that retains most of the original information. Those new variables are called Principal Components (PCs). They are uncorrelated and they are ordered by the fraction of the total information each retains [49].

Given n observations of the vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, the first PC is computed as:

$$z_1 \equiv \boldsymbol{a}_1^T \boldsymbol{x} = \sum_{i=1}^{n} a_{i1} x_i \tag{4.19}$$

where $\boldsymbol{a}_1 = (a_{1,1}, a_{2,1}, \ldots, a_{n,1})$ is the vector that maximize the variance of z_1.

In a similar way, the k^{th} PC (with $k = 1, \ldots, n$) is computed as:

$$z_k \equiv \boldsymbol{a}_k^T \boldsymbol{x} \tag{4.20}$$

where the vector $\boldsymbol{a}_k$ is chosen such that the variance of z_k is maximum, subject to $\text{cov}[z_k, z_l] = 0$ for $k > l \geq 1$ and $\boldsymbol{a}_k^T \boldsymbol{a}_k = 1$.

The generic observation $\boldsymbol{x}_i$ can be written as the sum of its PCs:

$$\boldsymbol{x}_i = \sum_{k=1}^{n} z_{ik} \boldsymbol{a}_k \tag{4.21}$$

According to the given definitions, it can easily be proved that the first PC retains the greatest amount of variation in the sample, whereas the k^{th} PC, z_k, retains the greatest k^{th} fraction of the variation in the sample. This fact allow us to approximate each observation by truncating the sum at the first $m < n$ PCs:

$$\boldsymbol{x}_i \simeq \boldsymbol{x}_i^m = \sum_{k=1}^{m} z_{ik} \boldsymbol{a}_k \tag{4.22}$$

Under-complete Autoencoder (UAE)

An Under-complete Autoencoder (UAE) is an ANN used for unsupervised learning of representations from a set of data, for the purpose of dimensionality reduction [33]. It learns to compress data from the input layer into a short code, and then uncompress that code into something that closely matches the original data (output layer). The set of hidden layers with decreasing number of neurons till reaching the central layer is called *encoder*. Another set of hidden layers from the central layer to the output layer is for the reconstruction of the original data and named *decoder*.

Figure 4.4 shows an example of UAE. The input and output layers are composed of n neurons. The single hidden layer, named *feature layer*, is composed of $m < n$ neurons. In this way, the original observation of size n is coded into a feature of dimension m.

The dataset $\mathcal{D}$ is split into two portions: the training set $\mathcal{T}$ and the validation set $\mathcal{V}$. The first set is used for the training of the UAE. During this phase, the backpropagation algorithm iteratively updates the weights of the connections between neurons to minimize the reconstruction loss (training loss). At the same time, the UAE is used to reconstruct the input data contained in the validation set. The validation

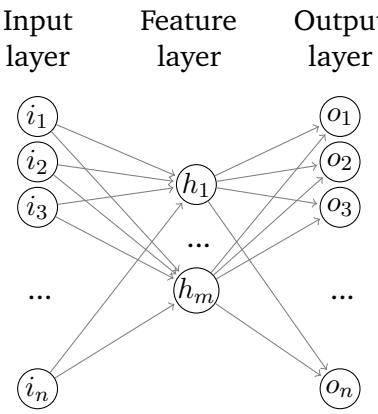


Figure 4.4: Architecture example of an autoencoder with a single hidden layer.

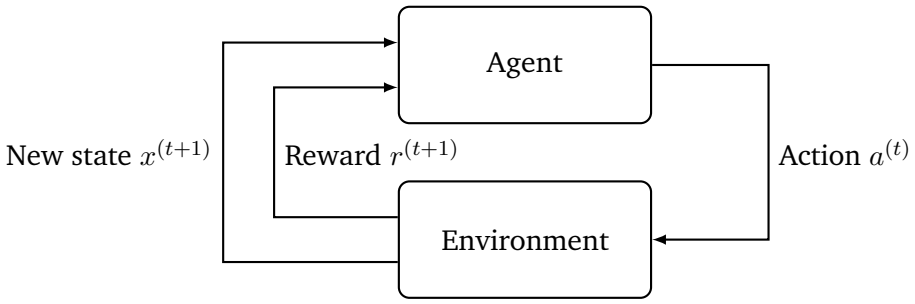


Figure 4.5: Scheme of the agent-environment interaction in reinforcement learning.

set provides an unbiased evaluation of the model fit on the training set (validation loss).

4.2.3 Reinforcement Learning

In RL, the learning process is represented by the interaction between a learner and decision maker, named *agent*, and the environment. The agent selects actions and the environment responds to those actions by returning new states of the environment. The environment also returns a numerical value named *reward* that the agent tries to maximize over time.

The agent and the environment interact at every time step. In particular, at each time step t, the agent receives a representation of the state of the environment, $x^{(t)} \in \mathcal{X}$, where $\mathcal{X}$ is the set of all the possible states. The agent selects an action $a^{(t)} \in \mathcal{A}(x^{(t)})$, where $\mathcal{A}(x^{(t)})$ is the set of actions available in state $x^{(t)}$. After one step, the agent receives a reward, $r^{(t+1)} \in \mathbb{R}$ and finds itself in the new state $x^{(t+1)}$. A scheme representing the interaction between agent and environment is depicted in Figure 4.5.

The objective of RL is to learn how to map experienced situation (i.e., the state of the system) into action to take so as to maximise a numerical reward. In more details, the learning agent must discover which actions yield the highest reward by trying them. This is a challenging goal because the taken action affects not only the immediate reward but also the next situations, though, all the subsequent rewards.

Basically all the RL algorithms are based on the estimation of value functions [92]. The argument of these functions can be a state or a state-action pair. In the first case the value function indicates how good is for an agent to be in a given state, whereas in the latter the value function indicates how good it is to perform a given action in a given state.

The quality of a state or state-action pair is measured in terms expected future reward, which depends on the policies being followed, π.

The value of a state x when the agent is following a policy π corresponds to the expected return when starting in state x and following the policy π thereafter. It corresponds to

$$v_\pi(x) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r^{(t+k+1)} | X^{(t)} = x \right] \tag{4.23}$$

where $\mathbb{E}_\pi$ represents the expectation of a random variable considering that the agent is following a policy π, t is the time step and $\gamma \in [0, 1]$ is the discount factor, which represents how much a future reward is affecting the current agent decision.

We define value of taking an action a in state x under the policy π, as $q_\pi(x, a)$. The action-value function for policy π is defined as the expected return starting from state x, taking action a and therefore following the policy π. We can write it as

$$q_\pi(x, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r^{(t+k+1)} | X^{(t)} = x, A^{(t)} = a \right] \tag{4.24}$$

The objective of RL is to obtain the highest return in the long run. We define as optimal policy, π^*, the policy that allows reaching this goal. According to [92], this policy is proved to be unique. The optimal value of state x is defined as

$$v^*(x) = \max_\pi v_\pi(x), \forall x \in \mathcal{X} \tag{4.25}$$

whereas the optimal action-value function is defined as

$$q^*(x, a) = \max_\pi q_\pi(x, a), \forall x \in \mathcal{X}, \forall a \in \mathcal{A}(x) \tag{4.26}$$

Temporal Difference Learning

Temporal Difference (TD) learning algorithms allow predicting a quantity that depends on future values of a given signal [92]. In the RL case, these algorithms are used to predict the total amount of reward expected over the future. In particular, the name refers to the fact that the prediction is based on differences over successive time steps.

TD methods are implemented in an online fashion, meaning that the agent learns from every transition without considering the following actions.

We distinguish between two operational phases of the agent: exploration and exploitation. In the *exploration* phase, the agent learns across the available actions in order to learn the best action to take. On the contrary, when in *exploitation* phase the agent exploits acquired knowledge to maximize its overall reward. Defining a good trade-off between these two phases is fundamental to guarantee good learning performance. The epsilon-greedy exploration policy is widely used in the literature. The agent must infinitely select the available actions in order to evaluate and improve the learned policy. Therefore, the agent needs to continue selecting all the actions to ensure that all of them are selected infinitely. There are two approaches to guarantee this, namely on-policy methods and off-policy methods. The *on-policy* methods evaluate/improve the policy that is used to make decisions (i.e., they estimate a policy while using it for control), whereas *off-policy* methods evaluate/improve a policy different from the one used to generate the data .

In this dissertation, we consider the Q-Learning (QL) algorithm and its variant Deep Q-Learning (DQL). More in details, QL is an off-policy TD control algorithm that is able to learn an approximation of the optimal policy independent of the policy being followed [92].

Q-learning (QL)

QL is an off-policy TD control algorithm introduced in [100] whereas the proof of the convergence has been given in [99]. The objective of QL is to recursively estimate the Q-function $q^*(x, a)$ by considering the available information $\left(x^{(t)}, a^{(t)}, x^{(t+1)}, r^{(t+1)}\right)$ where $x^{(t)}$ is the state at time t, $a^{(t)}$ is the chosen action, $x^{(t+1)}$ is the state at time

$t+1$ and $r^{(t+1)}$ is the reward returned by the environment after taking action $a^{(t)}$ in state $x^{(t)}$. The Q-function is updated according to the following rule

$$Q\left(x^{(t)}, a^{(t)}\right) \leftarrow Q\left(x^{(t)}, a^{(t)}\right) + \alpha\left[r^{(t+1)} + \gamma \max_{a'} Q\left(x^{(t+1)}, a'\right) - Q\left(x^{(t)}, a^{(t)}\right)\right], \tag{4.27}$$

where α is the learning rate and γ is the discount factor. The learning rate indicates to what extent newly acquired information overrides old information. Setting $\alpha = 0$ makes the agent learn nothing, whereas $\alpha = 1$ makes the agent consider only the most recent information. The discount factor sets the importance of future rewards. A value $\gamma = 0$ makes the agent short-sighted, only considering current rewards, whereas a value of γ close to 1 makes it strive for a long-term high reward.

The continuous exploration of the state-action pairs is guaranteed by the adoption of an ϵ-greedy policy.

The steps of the QL algorithm are detailed in Algorithm 2.

Algorithm 2 Q-Learning algorithm

1: Initialize $Q(x, a)\, \forall x \in \mathcal{X}, a \in \mathcal{A}$ arbitrarily
2: **for** episodes = $1, ..., M$ **do**
3: Initialize $x^{(1)}$
4: **for** $t = 1, \ldots, T$ **do**
5: Select action $a^{(t)} = \max_{a'} Q(x^{(t)}, a')$ with probability $1 - \epsilon$ otherwise take a random action with probability ϵ.
6: Execute action $a^{(t)}$ and observe the reward $r^{(t+1)}$ and the next state $x^{(t+1)}$
7: Update the Q-value by using (4.27)
8: **end for**
9: **end for**

In the standard QL algorithm, the Q-function is stored in a tabular form. The table describing the Q-function is therefore named *Q-table*. We highlight that continuous state variables need to be quantized due to the discrete nature of the Q-table, leading to quantization errors.

Deep Q-learning (DQL)

In the DQL approach, the Q-function is estimated by using a ANN approximator [73]. In details, the Q-function is approximated by the function $Q\left(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}|\theta\right)$, where θ represents the ANN parameters. The state of the system is the input of the ANN, whereas the output layer corresponds to the predicted Q-values of the individual action for the input state.

The agent experience at each time step t, $e_t = (x^{(t)}, a^{(t)}, r^{(t+1)}, x^{(t+1)})$, is stored into a replay memory $\mathcal{R} = \{e_1, \ldots, e_L\}$ with dimension L. A batch of l experiences is randomly sampled from $\mathcal{R}$ and used to perform the update of the Q-function by training the ANN. This phase is defined as *experience replay*.

During the training of the ANN, stochastic gradient decent is used to minimize a sequence of loss functions that changes at every training iteration i,

$$L_i(\theta_i) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{a}\sim\rho(\cdot)}\left[\left(y_i - Q\left(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}; \theta_i\right)\right)\right]^2 \tag{4.28}$$

where $y_i = \mathbb{E}_{\boldsymbol{x}'\sim\varepsilon}\left[r + \gamma \max_{a'} Q(x', a'; \theta_{i-1}) \,|x^{(t)}, a^{(t)}\right]$ is the target at iteration i and $\rho(x, a)$ is the probability distribution over states and actions named *behaviour distribution*. The ANN parameters from the previous iteration, θ_{i-1}, are kept fixed when optimizing the loss function $L_i(\theta_i)$. The parameters are updated at every iteration and the expectations are replaced by single samples from the behavior distribution ρ and the state distribution ε. We highlight that this particular solution allows to simultaneously update the Q-values for all the possible actions in a given state with only one forward pass in the ANN. After performing the update of the ANN parameters, an action is taken according to the ϵ-greedy policy.

The detailed steps of the DQL algorithm are listed in Algorithm 3.

Algorithm 3 Deep Q-Learning algorithm

Initialize the replay memory $\mathcal{R}$ to capacity L
Initialize the ANN with random weights
for episodes $= 1, ..., M$ **do**
  Initialize state $x^{(1)}$
  for $t = 1, \ldots, T$ **do**
    Select action $a^{(t)} = \max_{a'} Q(x^{(t)}, a'; \theta)$ with probability $1 - \epsilon$ otherwise take a random action with probability ϵ.
    Execute action $a^{(t)}$ and observe the reward $r^{(t+1)}$ and the next state $x^{(t+1)}$
    Store the experience $(x^{(t)}, a^{(t)}, r^{(t+1)}, x^{(t+1)})$ in $\mathcal{R}$
    Sample a random batch of l experiences $(x^{(j)}, a^{(j)}, r^{(j+1)}, x^{(j+1)})$ from $\mathcal{R}$
    Set $y^{(j)} = r^{(j+1)}$ for terminal state $x^{(j+1)}$ otherwise $y^{(j)} = r^{(j+1)} + \gamma \max_{a'} Q(x^{(j+1)}, a'; \theta)$
    Perform gradient descend step on $\left(y^{(j)} - Q(x^{(j)}, a^{(j)}; \theta)\right)$
  end for
end for

This approach presents several advantages with respect to the standard QL. DQL allows to easily deal with the continuous input space and it takes advantage of the generalization capacity of ANNs to estimate the Q-function. In fact, all the ANN parameters θ are updated when training for a single state-action pair. On the contrary, QL estimates the Q-function independently for each state-action pair due to its tabular form. Moreover, the adoption of experience replay and the randomization

of the replay buffer allows reducing the variance of the ANN parameters θ during the training of the ANN by breaking the correlations between consecutive samples of the stored experience of the agent.

4.3 Conclusions

In this chapter, we provided some background information about the mathematical framework adopted in this dissertation. In details, we introduced DP and the shortest path algorithm. Then, we provided an introduction of ML, with a focus on ANNs, which represent an important building block of this thesis. Finally, TD learning has been discussed as a method to design agents able of learning from the interaction with the environment.

5 Characterization of the Solar Energy

One of the key factor determining the performance of a PV system and its potential application to supply communication network devices, is the solar energy arriving at the surface of the Earth. The exploitation of the solar energy resource is determined by the knowledge of geographical variability and time dynamics. The geographical analysis of the availability of the primary solar energy resource is then essential to understand the potential implementation of PV systems for future energy supply to a specific industrial sector. The geographical dependency and distributed nature of solar electricity generation impose questions that require specific location-dependent answers. In fact, the harvested energy strictly depends on the seasons of the year and the meteorology of the given location. In [72], it has been estimated that even during summer and in good weather conditions, the harvested energy in the peak irradiation hour can vary up to the 85%. Similarly, considering that the solar radiation intensity and the daylight duration vary significantly across the months, seasons have a strong impact in the amount of the harvested energy income. Normally, this analysis is performed using solar maps, which provide easy-understandable information. The total annual solar electricity generation from a PV system is used to characterize national and regional differences [115].

In this chapter, we are interested in learning hidden features of the solar energy generation that may give a more accurate characterization of the process beyond that usual metric. We use unsupervised learning methods to automatically extract features from solar electricity generation data. Then, we interpret the latent variables characterizing the solar energy generation process by analyzing the similarities of the different cities with an agglomerative hierarchical clustering algorithm. Although ML methods have been used in renewable energy modeling, they have been adopted mainly to forecast the next energy arrivals in a given location [106]. Instead, to the best of our knowledge, they have not been used for analyzing geographical differences of solar electricity generation.

The chapter is organized as follows. In Section 5.1, we describe the studied solar energy dataset. In Section 5.2, we introduce the two ML methods used to perform feature extraction. In Section 5.3, we describe the clustering algorithm, and we analyze the obtained results. Finally, in Section 5.4, we draw our conclusions.

5.1 Dataset

The solar irradiance dataset considered in this work [89], contains hourly solar irradiance values, in W/m^2, collected from February 1st, 2004 to December 31st, 2006. In particular, we consider 67 different cities, located in Europe, Middle East, and North Africa.

The solar electricity generation is estimated using the model described in Section 3.1. In particular, the model parameters are set considering a Panasonic N235B PV module, which has single cell efficiencies of about 21%, delivering about 186 W/m^2. Each module is composed of an array of 16 $\times$ 16 solar cells (i.e., a surface of 4.48 m^2). We consider this particular dimension since it represents a realistic size for supplying renewable powered SBSs [81].

We define $\mathcal{D} = \{\mathcal{D}^{y_1}, \ldots, \mathcal{D}^{y_K}\}$ as the dataset containing the traces of generated energy, with $K = 67$. In particular, for each city y, we have a set $\mathcal{D}^y = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ containing n daily traces of the generated energy $\boldsymbol{e}_i$, $i \in \{1, \ldots, n\}$. The vector $\boldsymbol{e}_i$ has 24 elements, each one containing the amount of energy generated in the respective hour of the day. Since our dataset contains 35 months of measurements, $n = 1069$.

5.2 Feature Extraction

Two different approaches have been used to reduce the dimensionality of the input data and extract meaningful features: PCA and UAE.

Principal Component Analysis (PCA)

PCA has been used to extract the features from the input data, as described in Section 4.2.2. In particular, the number of considered PCs has been set to $m = 2$ since tests performed on the available data shows that the first 2 PCs retain the 94% of the information (respectively the 79% and the 15% for the 1st and the 2nd PC). Thus, we can associate a feature vector $\boldsymbol{h}_i^{\text{PCA}} = [z_1, z_2]$ to each observation $\boldsymbol{e}_i$. In this way, for each $\mathcal{D}^y$, we obtain a set of 2-dimensional features $\mathcal{F}^y_{\text{PCA}} = \{\boldsymbol{h}_1^{\text{PCA}}, \ldots, \boldsymbol{h}_n^{\text{PCA}}\}$, which is a compressed representation of the evolution of the solar energy generation in the city y.

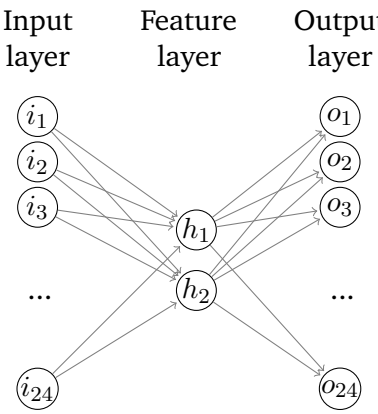


Figure 5.1: Autoencoder topology used for extracting features of the solar energy generation of the 67 cities in the dataset.

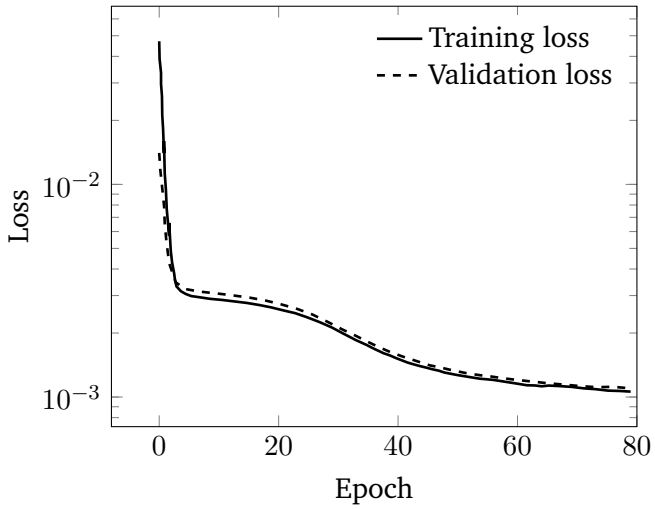


Figure 5.2: Training and validation losses of the UAE used for extracting features of the solar energy generation of the 67 cities in the dataset.

Under-complete Autoencoder (UAE)

An UAE has been implemented to extract the features as described in Section 4.2.2. The adopted architecture is reported in Figure 5.1. The input and output layers are composed of 24 neurons, each one representing the amount of energy generated in the corresponding hour of the day. The single hidden layer, named *feature layer*, is composed of 2 neurons. In this way, the information about the 24-hour solar energy generation trace is coded into a feature of dimension 2. Moreover, the neurons of the input and feature layers use the Rectified Linear Unit (ReLU) as activation function (i.e., $f_a(x) = \max(0, x)$), whereas the neurons of the output layer use the sigmoid function (i.e., $f_b(x) = 1/(1 + e^{-x})$).

The dataset $\mathcal{D}$ is split into a training set $\mathcal{T}$, containing $n_t = 730$ daily traces and a validation set $\mathcal{V}$ containing the remaining $n - n_t$ daily traces. Figure 5.2 shows

the decreasing behavior of the losses with the number of training epochs and the final error value achieved. After training, for each $\mathcal{D}^y$, the encoder computes a set of 2-dimensional features $\mathcal{F}^y_{\text{UAE}} = \{\boldsymbol{h}_1^{\text{UAE}}, \ldots, \boldsymbol{h}_n^{\text{UAE}}\}$.

5.3 Clustering and Result Discussion

We define the *Centroid of the city y* as the centroid of the features of the city y, computed as:

$$\boldsymbol{c}^y_{\text{PCA,UAE}} = \frac{\sum_{i=1}^{n} \boldsymbol{h}_i^{\text{PCA,UAE}}}{n} \tag{5.1}$$

where the value of the centroid depends on the approach used to extract the features (i.e., PCA or UAE).

A distance-based clustering named agglomerative hierarchical clustering algorithm [26] is applied to the centroids to group the cities according to their similarities. The similarity between cities is expressed as the distance between their centroids.

The number of clusters has been selected by performing the silhouette analysis [83]. This technique provides a measure of similarity of an object in its own cluster compared to other clusters. The silhouette value ranges in the interval [-1,1], being the highest value the best match of the object in the cluster and the smallest the poorest. The configuration is appropriate if most of the objects have a high value. Otherwise, the cluster configuration may have too many or too few classes. We have set the clustering algorithm to find the number of clusters that maximize the average silhoutte value and we obtained 2 clusters in the case of PCA and 5 clusters in the case of UAE.

The result of the clustering based on PCA feature centroids is shown in Figure 5.3, whereas the map obtained using UAE is reported in Figure 5.4. Moreover, we show the average hourly solar energy generated for each cluster obtained with PCA and UAE in Figure 5.5 and Figure 5.6, respectively.

The analysis of such variable for each cluster drives our interpretation of the extracted features to a more understandable model. In fact, we can distinguish four different parameters characterizing each identified cluster, namely: average daily generated energy E_H, solar energy peak value P_v, solar energy peak hour P_h and average daylight time D. The values of those parameters are reported in Table 5.1 and Table 5.2 for the PCA and UAE cases, respectively.

The map obtained with PCA is divided into two clusters (Figure 5.3). The first cluster covers the norh-east area and is characterized by a low amount of energy generation.

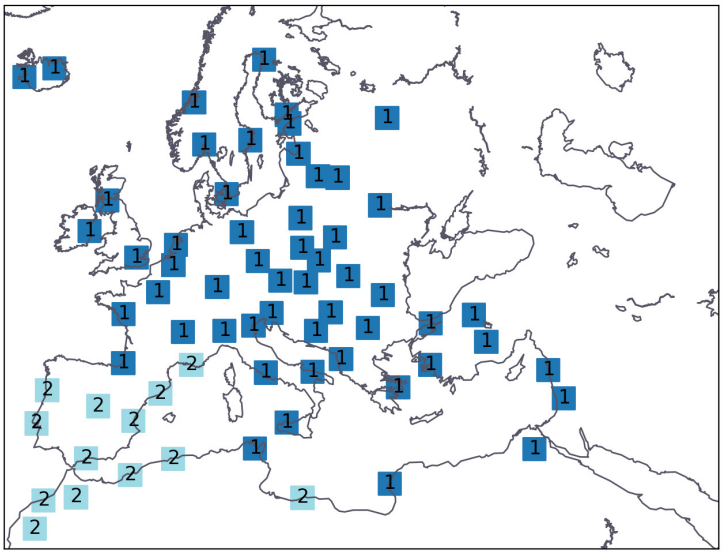


Figure 5.3: Clusters obtained by using the PCA approach.

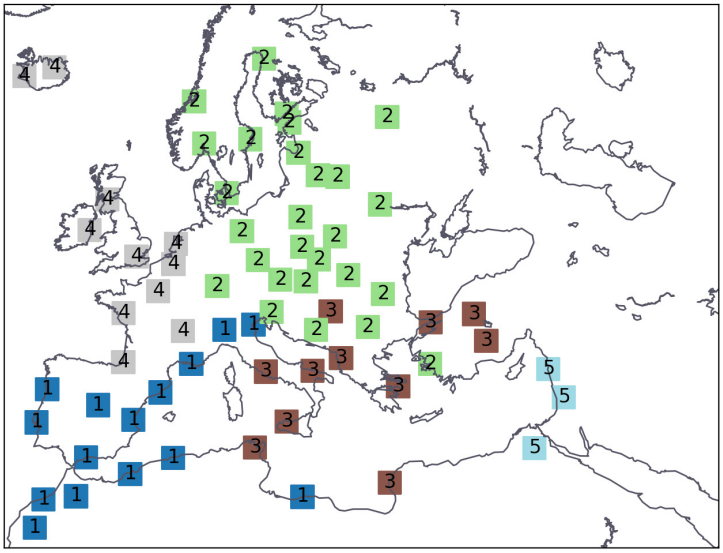


Figure 5.4: Clusters obtained by using the UAE approach.

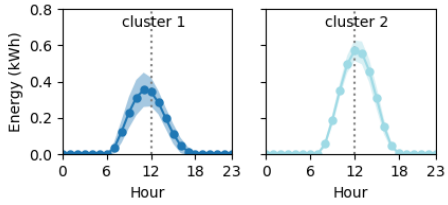


Figure 5.5: Average hourly solar energy generation of the clusters obtained by using the PCA approach. The shaded area represents the standard deviation with respect to the other cities in the cluster.

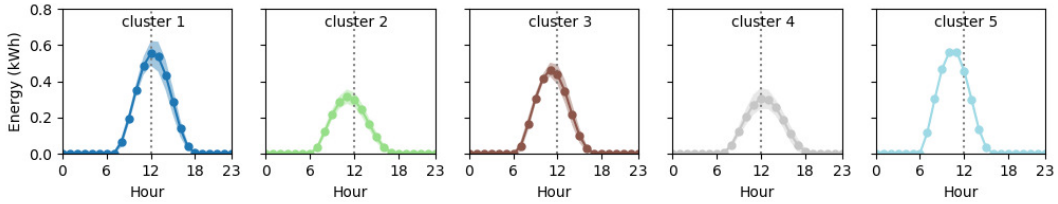


Figure 5.6: Average hourly solar energy generation of the clusters obtained by using the UAE approach. The shaded area represents the standard deviation with respect to the other cities in the cluster.

On the contrary, the second cluster covers the south-west area and is characterized by higher solar energy generation. Note that the two clusters differs also for peak hours. On the other hand, from the map obtained by the UAE approach (Figure 5.4), we see that the north of Europe is divided into two parts (clusters 2 and 4). Those clusters have the same values of P_v, D, similar E_H, but they differ in terms of solar peak hour. The south area is divided into three clusters (1, 3 and 5). Those clusters are very similar in terms of peak energy value, but they differ for peak hour, average energy and daylight hours.

Note that the parameter E_H represents the variable used by common solar maps to discriminate the different locations. In fact, those maps are usually based on the average or annual amount of solar energy generation. Therefore, PCA and UAE are able to extract two and three new variables featuring the solar energy generation process, respectively. The difference on the number of extracted latent variables may be due to the non-linearity of the autoencoder, which allows to obtain a better projection of the input data in the feature space with respect to the linear PCA.

The geographical representation described in this work is based on the characterization of several different behaviors of the solar energy generation in diverse geographical locations. In particular, we believe that it may be helpful in designing energy management systems that have to control industrial processes, which are strictly related to the human activity. In fact, the four parameters identified by our method provide a direct relation between the solar energy generation process and the time of the day.

Table 5.1: Statistics of the clusters obtained on features extracted with PCA

Cluster	P_h [hr]	P_v [kWh]	E_H [kWh]	D [hrs]
1	11	0.4	2.1	12
2	12	0.6	3.2	12

Table 5.2: Statistics of the clusters obtained on features extracted with UAE

Cluster	P_h [hr]	P_v [kWh]	E_H [kWh]	D [hrs]
1	12	0.6	3.1	12
2	11	0.3	1.8	13
3	11	0.5	2.5	11
4	12	0.3	1.9	13
5	10	0.6	2.9	11

5.4 Conclusions

In this chapter, we have proposed an unsupervised method to learn hidden features of the solar energy generation from a PV system that may give a more accurate characterization of the process. In a first step, solar radiation data has been converted into instantaneous solar power through a detailed source model. Then, two different approaches, namely PCA and UAE, have been used to extract meaningful features from the traces of the solar energy generation in an unsupervised manner. A hierarchical clustering algorithm has been used to group the locations according to their similarities in terms of solar energy generation. The results show that clustering on the extracted features provides support for learning latent variables of the solar energy generation process that may be used for a more detailed characterization of different geographical locations.

6 Optimal Direct Load Control

The introduction of RESs in our proposed scenario entails an intermittent and erratic energy budget for the communication operations of the SBSs. Therefore, Demand Response (DR) is fundamental to properly manage energy inflow and spending, based on the traffic demand.

The analysis of the literature in Chapter 2 highlighted that the approaches adopted for the Direct Load Control (DLC) of HetNets powered by RESs are mostly based on greedy schemes and ML. Although these approaches allow achieving energy savings, no information on the computed solution optimality is given. An algorithm for the computation of the optimal policies is fundamental to analyze which is the optimal control policy for the BSs and to compute a bound on the performance that our solution may achieve.

In this chapter, we fill the encountered gaps in the literature with the following contributions:

1. We propose an architecture based on the integration of distributed energy harvesting and storage systems in RANs with heterogeneous BSs deployment in urban environments.

2. We formulate the problem of optimal DLC of a two-tier mobile network based on DP. The key property of DP is that it applies optimal control as a trade off between the present cost and the future expected costs. This feature is fundamental in our scenario to prevent SBSs blackout during periods with low renewable energy arrivals and high traffic demands.

3. We provide a graphical representation of the problem and we use graph theory to model it. Then, we introduce an algorithm based on the shortest-path method to find the optimal ON/OFF policy for the SBSs.

4. We provide an analysis of the optimal policies and the network performance when considering different conditions of traffic demand and energy harvesting.

The chapter is organized as follows. In Section 6.1, we describe the system model. In Section 6.2, we we introduce the optimization problem and the methodology used for its solution. Then, in Section 6.3, we analyze the optimal policies adopted by the SBSs, and the achieved performance in terms of grid energy consumption and traffic drop. Finally, we draw our conclusions in Section 6.4.

6.1 System Model

We consider a Long-Term Evolution (LTE) RAN with a transmission bandwidth BW divided into R Resource Blocks (RBs) of 1 msec each [87]. The RAN is represented as a set of clusters which not interfere among each other. Every cluster is composed of 1 MBS and N SBSs. Each SBSs is powered by a solar panel and it can store energy into a battery. On the contrary, the MBS is connected to the electrical grid. The two feasible SBS operative states are: (i) ON, where the SBS serves the users in its coverage area, and (ii) OFF, where the SBS is in an energy saving mode and its users are handed over the MBS. The state of all the N SBSs at time t is descried by the vector $\boldsymbol{S}^{(t)} = \left[S_1^{(t)}, S_2^{(t)}, ..., S_N^{(t)}\right]$. Each element $S_i^{(t)}$, with $i = 1, ..., N$, is defined as follows:

$$S_i^{(t)} = \begin{cases} 0, & \text{if } i\text{-th SBS is OFF} \\ 1, & \text{if } i\text{-th SBS is ON} \end{cases} \tag{6.1}$$

The system evolves in time based on the variation of the traffic load and energy harvested. The traffic load experienced by the SBSs in the cluster at time t is defined as $\boldsymbol{L}^{(t)} = \left[L_1^{(t)}, L_2^{(t)}, \ldots, L_N^{(t)}\right]$ where $L_i^{(t)} \in [0, 1] \, \forall i$. The energy harvested by the SBSs at time t is indicated by the vector $\boldsymbol{H}^{(t)} = \left[H_1^{(t)}, H_2^{(t)}, ..., H_N^{(t)}\right]$, while the amount of energy stored in the SBSs batteries at time t is indicated by the vector $\boldsymbol{B}^{(t)} = \left[B_1^{(t)}, B_2^{(t)}, ..., B_N^{(t)}\right]$. The energy stored into the batteries at the beginning of the next time step is evaluated according to the formula

$$\boldsymbol{B}^{(t+1)} = \min\left(\boldsymbol{B}^{(t)} + \boldsymbol{H}^{(t)} - \boldsymbol{P}^{(t)}\Delta_t, B_{\text{cap}}\right) \tag{6.2}$$

where $\boldsymbol{P}^{(t)} = \left[P_1^{(t)}, P_2^{(t)}, \ldots, P_N^{(t)}\right]$ is the power consumed by the SBSs at time t, Δ_t is the duration of the time step, and B_{cap} is the capacity of the batteries.

Following the model introduced in Section 3.3, the BS energy consumption is approximated by the linear function $P = P_0 + \beta L$, where P_0 is the baseline power consumption, β is a hardware-specific constant and $L \in [0, 1]$ is the normalized traffic load. Typical values are $P_0^{\text{MBS}} = 750\text{W}$, $\beta^{\text{MBS}} = 600$ for MBSs and $P_0^{\text{SBS}} = 105.6\text{W}$, $\beta^{\text{SBS}} = 39$ for SBSs. This model is supported by real measurements and closely matches the real power profile of BSs [7].

If the SBS i is OFF at time t then its associated UEs are managed by the MBS and we assume that the SBS can be entirely switched OFF (i.e., $P = 0$). However, the MBS may have reached its capacity limit at that time instant (i.e., cannot allocate any RB to the UEs) and may drop part of the handed over UEs. We define this situation as *system outage*.

Moreover, the UE allocation scheme uses the methodology defined in [68]. The modulation and coding scheme is assigned to each UE as a function of its SINR, which is given by

$$\text{SINR} = \frac{|g_0|^2 P_{t,0}}{\sum_{i=1}^{N_I} |g_i|^2 P_{t,i} + \sigma_0^2}, \tag{6.3}$$

where $P_{t,0}$ and g_0 are the transmission and the channel gain for the useful transmission, respectively, N_I is the number of interferes, and $|g_i|^2$ and $P_{t,i}$ represent the channel gain and the transmission power of the i-th interferer. Finally, σ_0^2 is the power of the thermal noise.

6.2 Optimization Problem

The periodicity of the traffic demand and the energy arrivals leads to a cyclic evolution of the system. At every cycle t, a centralized controller computes the optimal state configuration of the SBSs in the cluster.

This sequential decision making process is modeled as a DP optimization problem. The objective is to minimize the grid energy consumed by the MBS while keeping the traffic drop rate of the system below a threshold D_{th}. Since there is a linear relation between the energy consumption and the BS load, the objective is converted into the minimization of the MBS load over a given time horizon, by offloading the traffic to the renewable powered SBSs. Furthermore, a threshold B_{th} on the battery level is introduced to prevent damages on the storage devices [67]. This optimization problem can be formulated as

$$\begin{aligned} \min_{\{\boldsymbol{S}^{(t)}\}_{t=1,\dots,K}} & \sum_{t=1}^{K} L^{\text{MBS}}\left(\boldsymbol{S}^{(t)}, t\right) \\ & D^{(t)} < D_{\text{th}} \\ & B_i^{(t)} > B_{\text{th}} \ \forall i. \end{aligned} \tag{6.4}$$

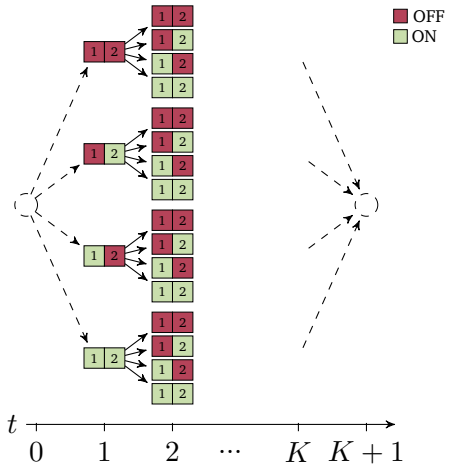


Figure 6.1: Graph showing the ON-OFF sequence possibilities in the case of a cluster with two SBSs. Green nodes represent ON states, red nodes represent OFF states. The two dashed nodes indicate the artificial nodes.

where K is the time horizon or the number of times the control is applied, $L^{\mathrm{MBS}}(\boldsymbol{S}^{(t)}, t)$ is the normalized load of the MBS given the SBSs states and the time instant t. Finally, $D^{(t)}$ is the traffic drop rate of the system at time t. Its value ranges from 0 (when all the traffic is served by the system) to 1 (when all the traffic is dropped by the system).

6.2.1 Graphical Representation

We represent the DP optimization problem as a graph. A node i at time t in the graph (V_i^t) represents a possible combination of states of the SBSs in the cluster. Each combination returns a different SBSs battery level and a different amount of dropped traffic in the system.

In Figure 6.1 a cluster of 2 SBSs is represented. In the first time step ($t = 1$) the SBSs can be in one of the four combinations of ON (green) / OFF (red) states. At each cycle t, the energy harvesting and traffic processes are evolving, based on $\boldsymbol{H}^{(t)}$ and $\boldsymbol{L}^{(t)}$. Each node V_i^t generates 4 child nodes V_j^{t+1}, as possible combinations at the cycle $t + 1$. The battery levels of the child nodes V_j^{t+1} are calculated based on Equation 6.2 and each arc connecting two nodes has a cost given by the function $L^{\mathrm{MBS}}(\boldsymbol{S}^{(t)}, t)$. The number of combinations is then evolving in time till reaching its maximum at time instant K. Two artificial nodes have been added at time step $t = 0$ and $t = K + 1$, to have a single initial node and a single terminal node. The cost associated to the arcs connecting the artificial nodes are set to zero.

The cost associated to each arc, $L^{\mathrm{MBS}}(\boldsymbol{S}^{(t)}, t)$, may be interpreted as the length of the corresponding arc. In this case, the problem of minimizing the total cost is equal to the problem of finding the path with the minimum-length from the initial to the terminal node.

The problem of finding the shortest path between the initial and the terminal node involves a very large number of nodes. However, most of these nodes are unlikely candidates for inclusion in the shortest path. Therefore, considering that we deal with a single initial and terminal node, and that each arc has a positive cost, we use the Label Correcting Algorithm described in Section 4.1.3 to achieve an efficient exploration method.

The graph is explored in a depth-first fashion (following the procedure described in Section 4.1.3) and the list OPEN contains only the nodes that are candidates for further examination and possible inclusion in the shortest path. More specifically, we exclude from the list all those nodes that cannot satisfy the constraint on the battery or on the drop or that return a minimum path longer than UPPER. This exploration policy avoids exploring the whole graph and requires relatively little memory, especially in the case of graphs with a tree-like structure, as in our case.

The algorithm steps are detailed in Algorithm 4.

Algorithm 4 Optimal policy algorithm

initialize OPEN with possible states at time t
while OPEN is not empty **do**
  remove a node V_i^t
  compute battery value $\boldsymbol{B}_j^{(t+1)}, j = 1, .., 2^N$, for all possible $\boldsymbol{S}^{(t+1)}$ using formula (6.2)
  for each node V_j^{t+1} child of V_i^t **do**
    $A_{i,t}^{j,t+1} = L^{\mathrm{MBS}}(\boldsymbol{S}_j^{(t+1)}, t+1)$
    if $d_i^t + A_{(i,t)}^{(j,t+1)} < \min\{d_j^{t+1}, \mathrm{UPPER}\}$ **and** $\boldsymbol{B}_j^{(t+1)} > B_{\mathrm{th}}$ **and** $\boldsymbol{D}_j^{(t+1)} > D_{\mathrm{th}}$ **then**
      $d_j^{t+1} \leftarrow d_i^t + A_{(i,t)}^{j,t+1}$
      set V_i^t parent of V_j^{t+1}
      if $t \neq K$ **then**
        place V_j^{t+1} in OPEN (if not already)
      else
        $\mathrm{UPPER} = d_i^t + A_{(i,t)}^{(j,t+1)}$
      end if
    end if
  end for
end while

6.3 Results and Discussion

In this section, we discuss the numerical results obtained by running the proposed optimal algorithm. In particular, we present an analysis of the optimal time horizon, and we evaluate the optimal policies when considering different conditions of the traffic demand and the energy harvesting. Finally, we provide an analysis of the network performance in terms of the grid energy consumption and the dropped traffic.

6.3.1 Simulation Scenario

We consider a square area with a side of 1 km. The MBS is located at the center of the area and 3 SBSs are randomly positioned. The SBSs have a transmission power of 38 dBm, which corresponds to a coverage radius of 50 m. The coverage areas of the SBSs do not overlap. The BSs have a transmission bandwidth of 5 MHz. Aggregated downlink traffic has been generated based on the traffic profiles and user classification described in Section 3.2. In particular, we consider three different weekly traffic profiles: residential, transportation and office. We underline that, with the considered approach, the traffic is described both in time (temporal variation during the week) and in space (spacial distribution in the area).

As for the RES system, we consider the Panasonic N235B solar modules, which have single cell efficiencies of about 21% delivering about 186 W/m^2. Each SBS is equipped with an array of 16$\times$16 solar cells (i.e., 4.48 m^2) and a 83 Ah battery. Panel and battery sizes have been chosen so that SBS batteries can be replenished in a full winter day. Realistic energy harvesting traces are obtained by using the model introduced in Section 3.1, considering the city of Los Angeles. All the simulations have been performed focusing on a generic week of December and July, in order to highlight the differences between the worst and the best months in terms of energy arrivals.

The optimal approach is compared with a naive algorithm that operates by turning OFF the SBSs when their battery levels go below the threshold B_{th} and turning them ON when the level is above it. Additional simulation parameters may be found in Table 6.1.

Table 6.1: Simulation parameters

Parameter	Value
BS Bandwidth	5 MHz
Channel model	Okumura-Hata [40]
Macro BS TX power	43 dBm
SBS TX power	38 dBm
Solar modules	Panasonic N235B
Solar cell efficiency	21%
B_{th}	0.2

6.3.2 Optimal Time Horizon

Here below we empirically analyze the optimal duration of the time horizon K that allows achieving the minimum cost. This parameter may give an idea on the temporal correlation among the control actions.

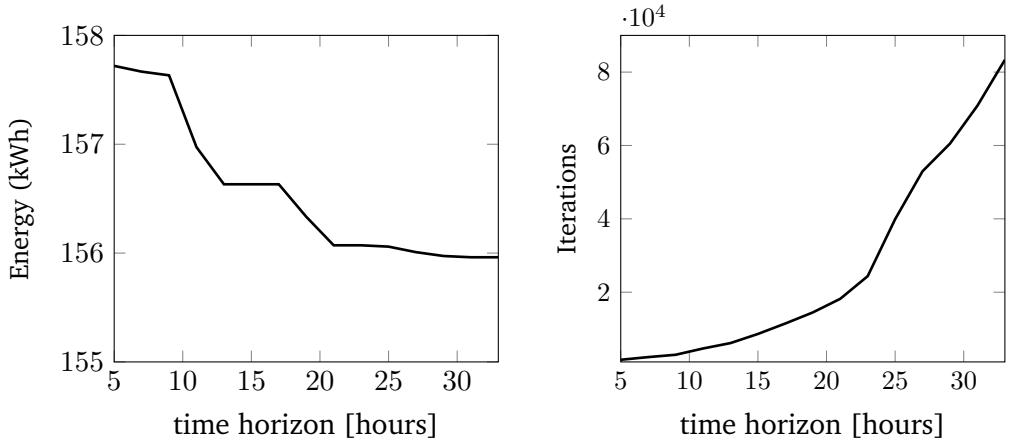
Figure 6.2 represents the amount of energy drained from the grid by the MBS in one week for different dimensions of the horizon K (Figure 6.2a) and the algorithm complexity, in terms of number of iterations, over the time horizon K (Figure 6.2b) for a single SBS within the coverage area of the MBS.

The time horizon $K = 21$ represents a turning point for both grid energy and algorithm complexity: the energy drained from the grid approaches an asymptote and the number of iterations explodes to higher values quasi-exponentially. Simulations performed in scenarios with multiple SBSs show the same behavior in terms of energy drained, number of iterations and time horizon K. Therefore, we state that a time horizon of about 21 hours represents a good trade-off between network performance and algorithm complexity.

This value is therefore adopted in all the following simulations.

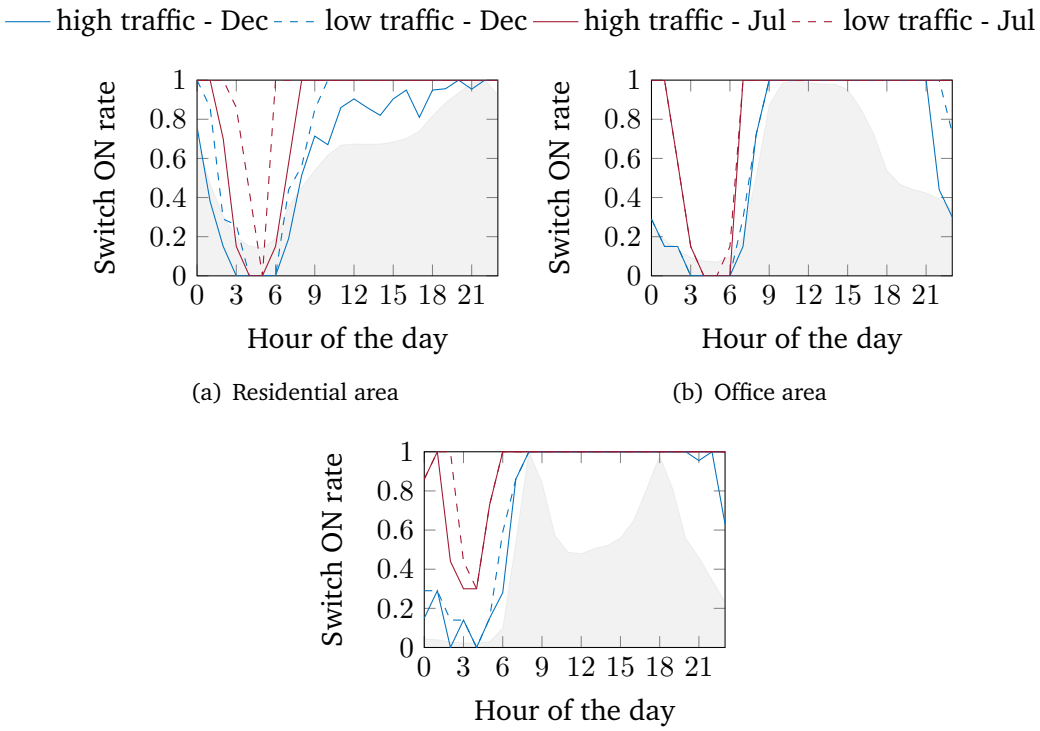
6.3.3 Optimal ON-OFF Policies

The traffic demand in the considered areas differs both in terms of temporal distribution and magnitude. In particular, the residential area is the most demanding, with a weekly aggregated traffic of 2.17 TB. The lowest traffic demand is experienced in the transportation area, with a weekly aggregated traffic of 0.41 TB. The three traffic profiles and the two energy arrival processes are not always time correlated. In fact, in the case of residential traffic, the peak of the demand is at 10 pm. In the



(a) (b)

Figure 6.2: Grid energy consumption (a) and number of algorithm iterations (b) of the optimal policy when varying the time horizon.



(a) Residential area (b) Office area

(c) Transportation area

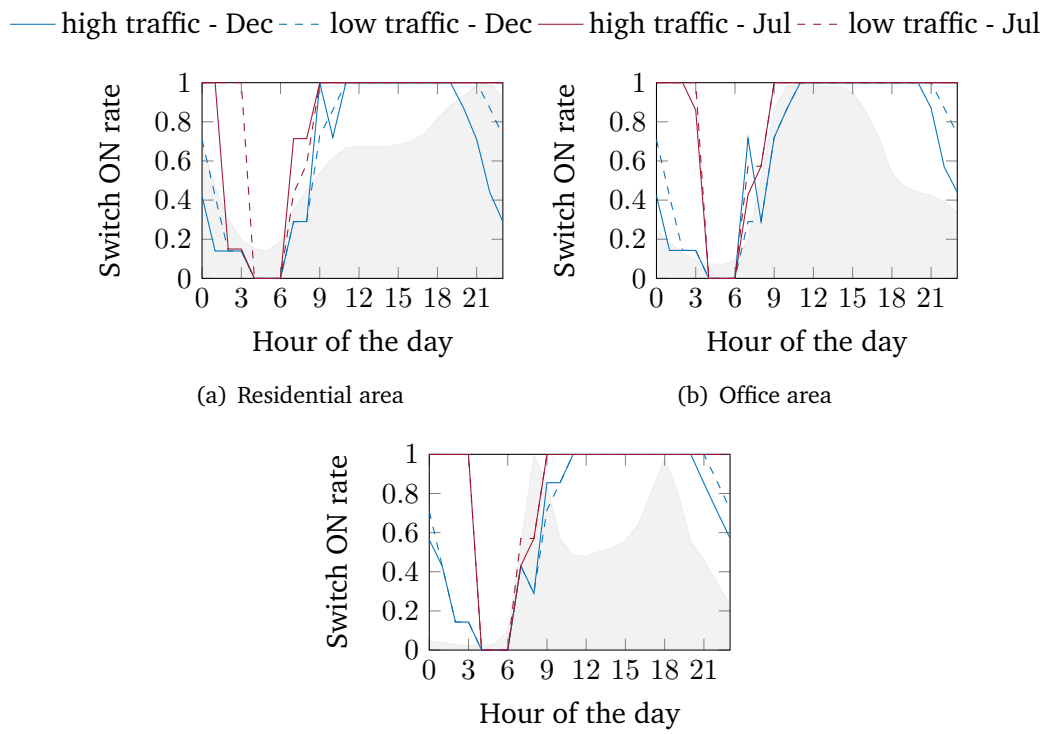
Figure 6.3: Daily average switch ON rate for the optimal algorithm. Simulations on the residential, office and transportation traffic profile for a week of December and July. The scenario with 10 UEs (20% heavy) is indicated as low traffic, whereas the scenario with 90 UEs (50% heavy) is indicated as high traffic. The shaded area represents the shape of the traffic demand.

case of office traffic, it is at 11 am on the weekdays and at midday on the weekends. Finally, in the transportation case we have two peaks during the weekdays, at 8 am and 6 pm, while on the weekends there is a single peak at 5 pm. From the energy side, the peak of energy arrivals is always between midday and 1 pm for the two considered months. This confirms the necessity of taking into account present and future costs when the optimal control is applied.

The daily average switch ON rate of the SBSs for the optimal and the naive policy is reported in Figure 6.3 and Figure 6.4, respectively. The three traffic profiles and the months of December and July are depicted. We consider a high-traffic intensity involving 90 UEs (50% heavy) represented by solid lines, and a low-traffic intensity with 10 UEs (20% heavy) indicated with dashed lines.

In Figure 6.3, we observe that the number of SBSs in OFF is generally higher during night hours due to the scarce availability of the energy and the low traffic demand. More in detail, we can notice that high-traffic intensity and low energy arrivals (December) result in longer and more intensive switch OFF periods during the night. For the residential profile (Figure 6.3a) SBSs experience intense switch OFF in a typical week of July between 3 am and 6 am (low-traffic case) and from 2 am to 8 am (high-traffic case). In a week of December, intense switch OFF is experienced from 1 am to 10 am (low-traffic case). In the case of office (Figure 6.3b) and transportation (Figure 6.3c) profiles, the intensive night OFF periods are less influenced by the total number of UEs served by the SBSs. This is due to the low magnitude of the total traffic demand experienced in these areas.

In Figure 6.4, we observe that the switch ON rate is lower than 1 in periods of high traffic demand. In fact, the naive algorithm takes immediate decisions without considering any future evolution of the traffic and energy arrival processes. In this way, a SBS always consumes the available energy and then it remains in an OFF state until the harvested energy is sufficient to return operative. On the contrary, the optimal policy turns OFF a SBS in an intelligent way, by considering the future evolution of the traffic and energy arrivals. Therefore, it saves energy during low traffic periods (e.g., night hours) to maintain ON the SBS during high traffic peaks, which may correspond to scarce energy arrivals. For instance, let's consider the residential profile with high-traffic intensity in December, where the peak of the traffic demand is at 10 pm. The naive algorithm switch ON rate is 0.5 at 10 pm (Figure 6.4a), whereas it is equal to 1 during the daytime (i.e., from 11 am to 7 pm). In fact, the SBSs immediately use the available energy during the day, and then switch OFF in the evening due to scarce energy availability. On the contrary, the optimal switch ON rate (Figure 6.3a) is almost 1 during the traffic peak hours and it has an average of 0.89 during daytime. This behavior indicates that some SBSs are



(a) Residential area

(b) Office area

(c) Transportation area

Figure 6.4: Daily average switch ON rate for the naive algorithm. Simulations on the residential, office and transportation traffic profile for a week of December and July. The scenario with 10 UEs (20% heavy) is indicated as low traffic, whereas the scenario with 90 UEs (50% heavy) is indicated as high traffic. The shaded area represents the shape of the traffic demand.

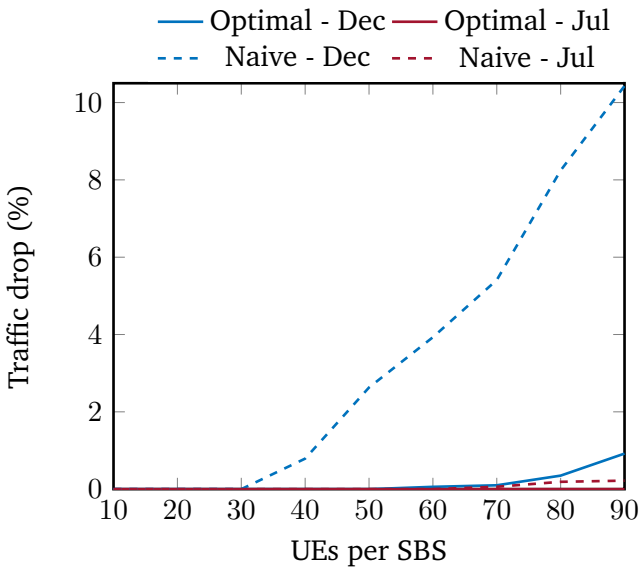


Figure 6.5: Percentage of weekly traffic request not serviced for both the naive and optimal algorithm in December and July. The 50% of the UEs are heavy users.

switched OFF during the day to save the necessary energy to satisfy the traffic peak in the evening.

6.3.4 System Outage

In this section, we analyze the system outage measured as the percentage of the traffic dropped in the system. We present the case of residential profile only, since the others have similar performance. The percentage of the traffic dropped in a week is reported in Figure 6.5 for a number of UEs ranging from 10 to 90 (50% of them are heavy users).

The optimal policy succeeds in delivering all the requested traffic and the system does not experience any outage in almost all the studied situations. In December, however, some traffic is dropped starting from 60 UEs per SBS, reaching the maximum of 0.9% for 90 UEs. The naive approach, on the other hand, always performs worse than the optimal policy. In particular, in December the dropped traffic reaches 10% in the case of 90 UEs per SBS.

This phenomenon is confirmed in Figure 6.6, where the average hourly dropped traffic is shown for a scenario with 90 UEs per SBS with 50% of heavy users. As for the optimal policy, the outage is concentrated in the morning (from 7 am to 10 am), afternoon (5 pm) and night, with values that reach the maximum of 5% at midnight. The naive approach, instead, causes system outage for longer periods

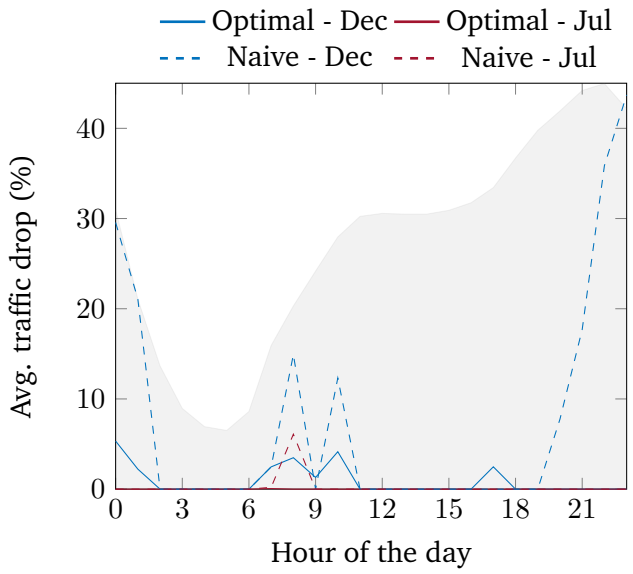


Figure 6.6: Average hourly traffic drop for the optimal and the naive algorithms in December and July. The traffic profile is residential and every SBS has 90 UEs in its coverage area; 50% of them are heavy users. The shaded area represents the shape of the traffic demand.

and with higher values of the dropped traffic, which reaches a maximum of 44% at 11 pm.

6.3.5 Energy Consumption

The amount of grid energy consumed by the MBS is shown is Figure 6.7, varying the number of UEs in the coverage area of the SBSs. We consider two cases: a scenario with 20% and 50% of heavy users, respectively. In both scenarios, the grid energy consumption increases linearly with the number of UEs. The slope of the curves is higher for the scenario with 50% of heavy users since the traffic increases faster with the number of UEs. Grid energy consumption is higher in December since the scarce availability of the renewable energy turns out into longer SBS sleeping periods and higher MBS operation.

The naive approach presents higher values of the grid energy consumption than the optimal policy. However, in the case of December and with 50% of heavy users, we observe that the naive approach has lower energy consumption for more than 70 UEs per SBS. This behavior is due to the fact that the system is heavily in outage and loses a considerable amount of traffic, as described in the previous section.

Finally, Figure 6.8 reports the grid energy consumption for a week of July in different architecture scenarios. We compare a solution where MBS and SBSs are connected to the grid (also referred to as *grid-only*) with our scenario where SBSs are solely

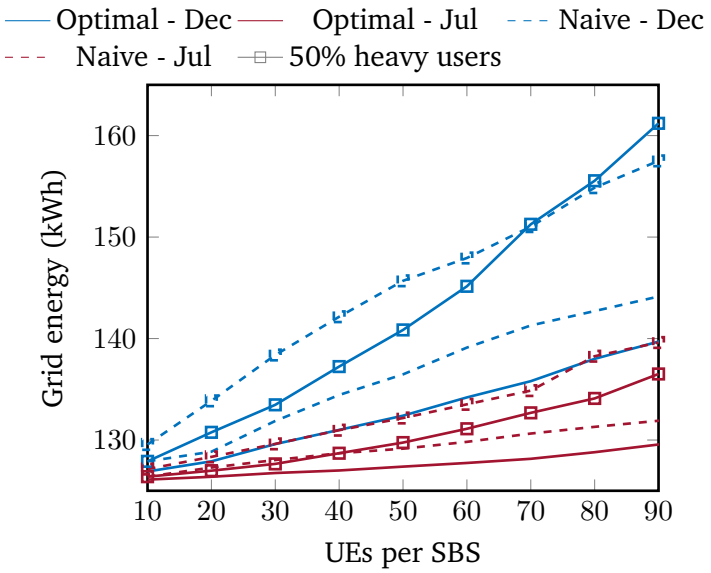


Figure 6.7: Grid energy consumption for the optimal and the naive algorithms in December and July, while increasing the number of UEs in the SBS area.

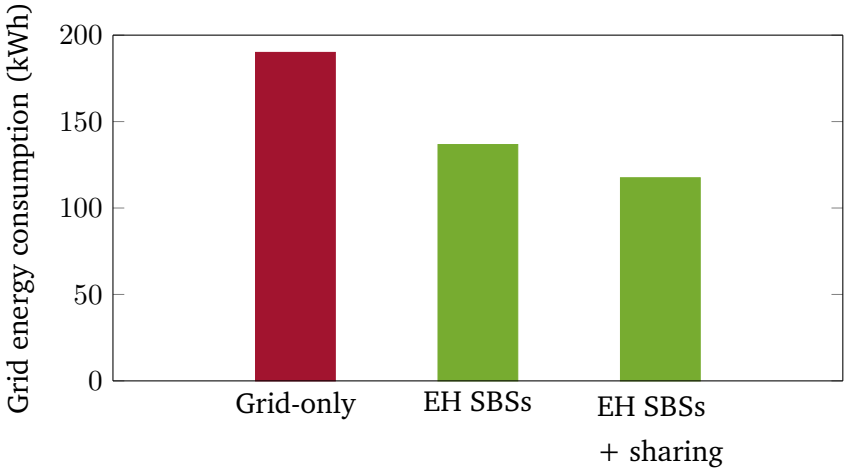


Figure 6.8: Grid energy consumption for different deployment architectures during a week of July. The traffic profile is residential and every SBS has 90 UEs in its coverage area; 50% of them are heavy users.

powered by solar panel plus battery (also referred to as *EH SBSs*). The grid-only scenario consumes 190.3 kWh in a week; deploying renewable powered SBSs saves 28% of the grid energy. Moreover, since the RES systems have been dimensioned for winter, the harvested energy may be abundant during summer and be discarded by the SBSs, i.e., it can neither be used for transmission nor stored in the battery. This redundant energy is concentrated at the peak hours of the energy arrival process (i.e., between midday and 2 pm). Considering that the SBSs may be connected through a power micro-grid, the excess energy can be used for ancillary services (e.g., light system) or shared to support the MBS operation, thus reducing its grid energy consumption. In fact, a grid energy saving of 38% is achieved, in case the MBS uses the energy shared by the SBSs (also referred to as *EH SBSs + energy sharing*).

6.4 Conclusions

In this chapter, we have introduced an optimal DLC of renewable powered SBSs in a two-tier mobile network. We have analyzed the optimal policies and their dependence on the traffic and the energy arrival processes. We have compared the optimal approach with a naive algorithm and analyzed the network performance in different scenarios. We have also introduced a new possibility of energy sharing among the network elements to reduce the dependence on the power grid and increase the energy savings.

From this analysis, we can draw the following conclusions. The different temporal behavior of the traffic and energy arrival processes highlights the necessity of deploying storage devices along with solar panels. Moreover, it is fundamental to properly manage the storage to maintain good network performance. The comparison between the optimal and the naive approach shows that grid energy savings and traffic drop limitations are possible only if the control algorithm is able to forecast the evolution of the two processes. Finally, the analysis of the redundant energy shows that sharing energy among BSs may lead to considerable amount of grid energy savings. This scenario will be further analyzed in the next chapter.

7 Optimal Direct Load Control plus Energy Sharing

In this chapter, we focus on the design of energy self-sustainable mobile networks, by enabling intelligent energy management that allows the BSs to mostly operate off-grid by using renewable energies. Building on the conclusions of the previous chapter, we advocate future mobile networks with a hierarchical cell structure and powered by energy harvesting hardware. BSs within the same geographical area are grouped in a micro-grid and operate almost autonomously from the power grid. To achieve this goal, we target the design of an optimal DLC method with energy sharing within the micro-grid. We solve the optimization problem by using the graph-based method introduced in Chapter 6 and we demonstrate, via software simulations, that a combination of load control plus energy sharing represents a viable and economically convenient solution for enabling energy self-sustainability of mobile networks grouped in micro-grids.

The analysis of the literature in Chapter 2 highlighted that both communication cooperation and energy sharing methods are adopted as a way to cooperate between BSs. The approach presented in this chapter is the first that consider a combination of them for minimizing the mobile network power consumption.

The contributions of this chapter are summarized in the list below:

- We propose a RAN architecture with SBSs powered by distributed energy harvesters and storage devices. Energy sharing between BSs is used to improve the energy efficiency of the network.

- We provide a theoretical formulation of the joint load control and energy sharing optimization problem. The objective of the problem is to minimize the grid energy consumption and the system outage.

- We design a joint optimal load control plus energy sharing method for a two-tier HetNet.

- We analyze the dimensioning of the energy harvesting and storage system and its influence to the network performance.

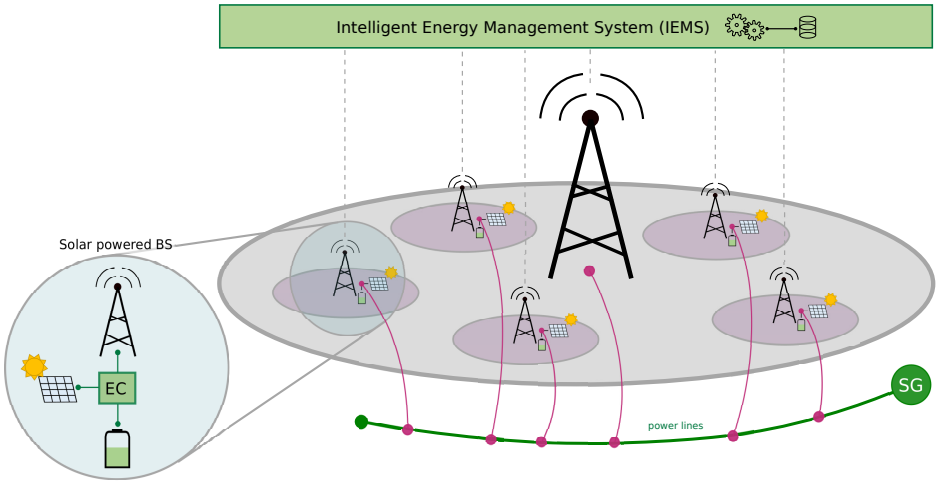


Figure 7.1: Diagram illustrating the reference framework, including the RAN with multiple tiers, the intelligent energy management system and the micro-grid connections. In left-bottom side, a simplified scheme of the solar-powered BS is shown.

- We provide an analysis of the energy saving and monetary cost for different energy harvesting and storage design approaches.

The chapter is organized as follows. In Section 7.1, we introduce the reference framework. In Section 7.2, we describe the system model. In Section 7.3, we introduce the optimization problem and the methodology used for its solution. Then, in Section 7.4, we discuss results on the dimensioning of the harvesting and storage system, the optimal SBS configuration policies and the energy shared within the micro-grid. In Section 7.5, we provide an energy and monetary cost analysis of the studied architecture. Finally, we draw our conclusions in Section 7.6.

7.1 Reference Framework

We consider the multi-tier architecture described in the previous chapter. However, in this new scenario, the BSs in the cluster are connected into a micro-grid and the renewable energy is managed based on a *harvest-store-share* approach. In details, the power harvested by the solar panel is consumed by the SBS and any excess is stored into the battery. Whenever the battery has reached its full capacity, the excess energy is shared with the MBS that will exploit it to reduce its consumption of energy from the electricity grid. The micro-grid is implemented by deploying power lines. Low resistive losses (i.e., the energy lost in the conductor due to Joule heating) are guaranteed by the short distances between MBS and SBSs. When the SBSs have exceeding energy that is not needed by the MBS, this energy is locally dissipated at the SBSs site and not injected into the micro-grid. In this way, the micro-grid energy balance is maintained.

In the proposed framework, we envision a central control unit named Intelligent Energy Management System (IEMS), located at the MBS site. The central controller has a full knowledge of the the network condition, enabling a better coordination of the SBSs. In particular, the SBSs operational states depends on the dynamics of the energy harvesting and traffic demand. Therefore, the IEMS is in charge of opportunistically operate the network to achieve efficient utilization of the harvested energy and prevent SBSs blackout during periods with low renewable energy arrivals and high traffic demand.

The Energy Controller (EC) is an entity located at the SBSs site which is in charge of communicating to the IEMS the necessary local information (e.g., the battery level) and to implement the decisions taken by it. The control architecture is shown in Figure 7.1.

7.2 System Model

In this section, we modify the system model defined in Section 6.1 by introducing the energy sharing between SBSs and MBS. The RAN consists of a set of clusters which not interfere among each other. Every cluster is composed of 1 MBS and N SBSs. Each SBS is powered by a solar panel and it can store energy into a battery. On the contrary, the MBS is connected to the electrical grid.

Each SBS can be in one of the following operative states: (i) ON, where it serves the users in its coverage area, or (ii) OFF, where it is in an energy saving mode and its users are handed over the MBS. The vector $\boldsymbol{S}^{(t)} = \left[S_1^{(t)}, S_2^{(t)}, ..., S_N^{(t)}\right]$ describes the state of all the N SBSs at time t. Each element $S_i^{(t)}$, with $i = 1, ..., N$, is defined as follows:

$$S_i^{(t)} = \begin{cases} 0, & \text{if } i\text{-th SBS is OFF} \\ 1, & \text{if } i\text{-th SBS is ON} \end{cases} \tag{7.1}$$

The vector $\boldsymbol{L}^{(t)} = \left[L_1^{(t)}, L_2^{(t)}, \ldots, L_N^{(t)}\right]$ describes the traffic load experienced by the SBSs in the cluster at time t, where $L_i^{(t)} \in [0, 1] \, \forall i$. The energy harvested by the SBSs at time t is indicated by the vector $\boldsymbol{H}^{(t)} = \left[H_1^{(t)}, H_2^{(t)}, ..., H_N^{(t)}\right]$, while the amount of energy stored in the SBSs batteries at time t is indicated by the vector $\boldsymbol{B}^{(t)} = \left[B_1^{(t)}, B_2^{(t)}, ..., B_N^{(t)}\right]$. The energy stored into the batteries at the beginning of the next time step is evaluated according to the formula

$$\boldsymbol{B}^{(t+1)} = \min\left(\boldsymbol{B}^{(t)} + \boldsymbol{H}^{(t)} - \boldsymbol{P}^{(t)}\Delta_t, B_{\text{cap}}\right) \tag{7.2}$$

where $\boldsymbol{P}^{(t)} = \left[P_1^{(t)}, P_2^{(t)}, \ldots, P_N^{(t)}\right]$ is the power consumed by the SBSs at time t, Δ_t is the duration of the time step, and B_{cap} is the capacity of the batteries. The amount of energy that exceeds the battery capacity, and can be shared within the network by the SBSs, is defined by the vector $\boldsymbol{X}^{(t)} = \left[X_1^{(t)}, X_2^{(t)}, \ldots, X_n^{(t)}\right]$, and calculated as

$$\boldsymbol{X}^{(t+1)} = \max\left(\boldsymbol{B}^{(t)} + \boldsymbol{H}^{(t)} - \boldsymbol{P}^{(t)}\Delta_t - B_{\text{cap}}, 0\right) \tag{7.3}$$

Moreover, we adopt the BS energy consumption model and the UE allocation scheme described in Section 6.1.

7.3 Optimization Problem

The periodicity of the traffic demand and the energy arrivals leads to a cyclic evolution of the system. At each cycle, the IEMS decides the optimal configuration of the SBSs in the cluster to serve the traffic demand in that area. The optimization goal is to minimize the grid energy consumption, while keeping the percentage of dropped traffic experienced in the cluster below a threshold D_{th}, and the SBSs battery levels above a threshold B_{th}. This optimization problem can be formulated as

$$\begin{aligned} \min_{\{\boldsymbol{S}^{(t)}\}_{t=1,\ldots,K}} & \sum_{t=1}^{K} \mathrm{E}^{\text{MBS}}\left(\boldsymbol{S}^{(t)}, t\right) \\ & D^{(t)} < D_{\text{th}} \\ & B_i^{(t)} > B_{\text{th}}\ \forall i. \end{aligned} \tag{7.4}$$

where K is the time horizon, $D^{(t)}$ is the normalized traffic drop at time t and E^{MBS} is the normalized grid energy consumption drained by the MBSs, given the operative modes of the SBSs at time t, whether ON or OFF. The normalized grid energy consumption drained by the MBSs at time t is computed as:

$$\mathrm{E}^{\text{MBS}}\left(\boldsymbol{S}^{(t)}, t\right) = \frac{P_{\text{MBS}}\left(\boldsymbol{S}^{(t)}, t\right) \cdot \Delta_t - \sum_{i=1}^{N} X_i^{(t)}}{P_{\text{MBS}}^{\max}\Delta_t} \tag{7.5}$$

where P_{MBS} and $P_{\text{MBS}}^{\max}$ are respectively the power and the peak power consumed by the MBS.

This optimization problem has been solved by using the graph-based approach introduced in Section 6.2.1.

Table 7.1: Simulation parameters

Parameter	Value
BS Bandwidth	5 MHz
Channel model	Okumura-Hata [40]
MBS TX power	43 dBm
SBS TX power	38 dBm
Solar modules	Panasonic N235B
Solar cell efficiency	21%
Traffic area	Residential
B_{th}	0.2

7.4 Results and Discussion

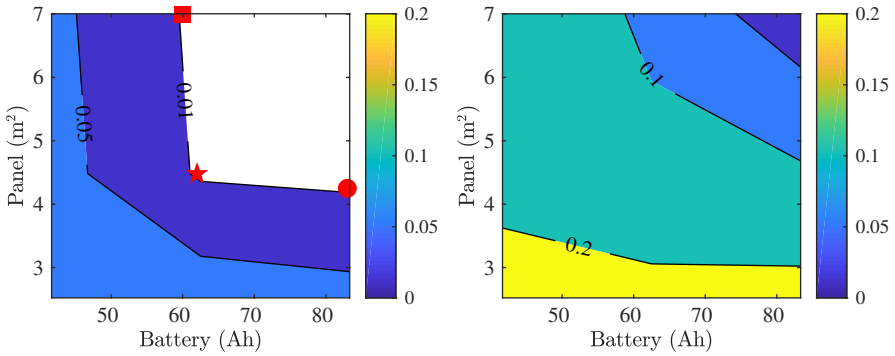
In this section, we provide an analysis of the dimensioning of the harvesting and storage devices. Then, we analyze the operative state configuration of the SBSs and the energy shared with the MBS.

7.4.1 Simulation Scenario

We consider a square area with a side of 1 km and one MBS located in the center providing baseline coverage. Three SBSs are positioned in hotspots for capacity enhancement. The coverage areas of the SBSs do not overlap. Energy arrivals and aggregated downlink traffic have been generated according to the realistic models described in Section 3.1 and Section 3.2. In particular, the city of Los Angeles has been used for generating the solar energy traces. The adopted power consumption model is described in Section 3.3. The results provided in what follows are averaged among ten different independent realizations of both energy arrivals and traffic processes. In the discussion presented next, we refer to January February, October, November and December as winter months; the remaining part of the year is considered as summer. Additional simulation parameters are listed in Table 7.1.

7.4.2 Dimensioning of the Harvesting and Storage System

Figure 7.2 shows the contour plots of the traffic drop rate during the month of December (the worst in terms of harvested energy). Different colors are used to indicate traffic drop rate regions (maximum values are specified in the associated color map). The white filled area indicates the parameter region where the traffic drop is smaller than 1%. Our optimal analysis is compared with a naive approach.



(a) Optimal (b) Naive

Figure 7.2: Contour plot of the traffic drop rate of the optimal and the naive algorithm. Different colors indicate traffic drop rate regions, whose maximum drop value is specified in the color map in the right hand side of the plot. The white filled region indicates a traffic drop rate smaller than 1%.

The naive algorithm switches OFF a SBS when its battery level is below the threshold B_{th} and turns it ON when the battery is above.

Taking 1% as our design parameter, all the points on the boundary of the white filled region are equally good. It is evident that the use of the optimal load control with energy sharing allows the network to work with much lower sizes of the harvesting/storage system compared to the naive approach. These results confirm that an intelligent energy management system is essential for an efficient use of the renewable energy resource and its installation in town facilities.

The analysis in the following parts of the chapter considers various harvesting/storage design approaches corresponding to the different points laying in the boundary of the white filled area of Figure 7.2 and labeled with star, circle and square symbols.

7.4.3 SBSs Operative State Configuration

In this section, we provide an analysis of the behaviors adopted by the SBSs when experiencing different conditions of harvested energy.

In Figure 7.3, we report the different choices of the operative states of the SBSs by the optimal policy across the different months of the year. The graphs refer to the selected harvesting/storage dimensions.

The SBSs are active and offload the MBS longer during the summer months (high energy inflow). We can appreciate different behaviors based on the harvesting/storage dimensions: star approach ranges between 53% (in December) and 65% (in August)

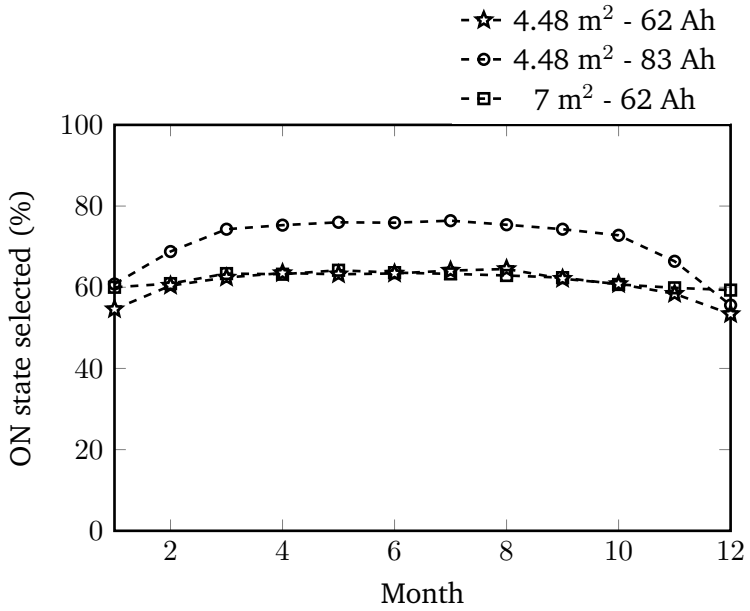


Figure 7.3: Percentage of time the ON state is selected by the optimal algorithm when considering different months and deployment sizes.

of time, circle between 56% (in December) and 76% (in July) of time and square between 59% (in December) and 64% (in July) of time.

7.4.4 Shared Energy Assessment

In Figure 7.4, we show the energy shared and used by the MBS for every month of the year. The graphs are collected considering the selected harvesting/storage design approaches. For comparison purposes, we also indicate the amount of shared energy experienced when using the naive approach. In particular, we consider the smallest configuration of the harvesting/storage devices that allows meeting the drop requirements.

We observe a general trend of sharing a bigger amount of energy during the summer months, when a higher solar energy inflow occurs and, hence, a higher probability of exceeding the battery capacity of the SBSs. The square harvesting/storage deployment presents the highest amount of shared energy and the circle approach the poorest.

7.5 Energy Savings and Cost Analysis

In Table 7.2, we provide an energy and monetary cost comparison between a scenario where MBS and SBSs are connected to the grid (also referred to as *grid-only*) and

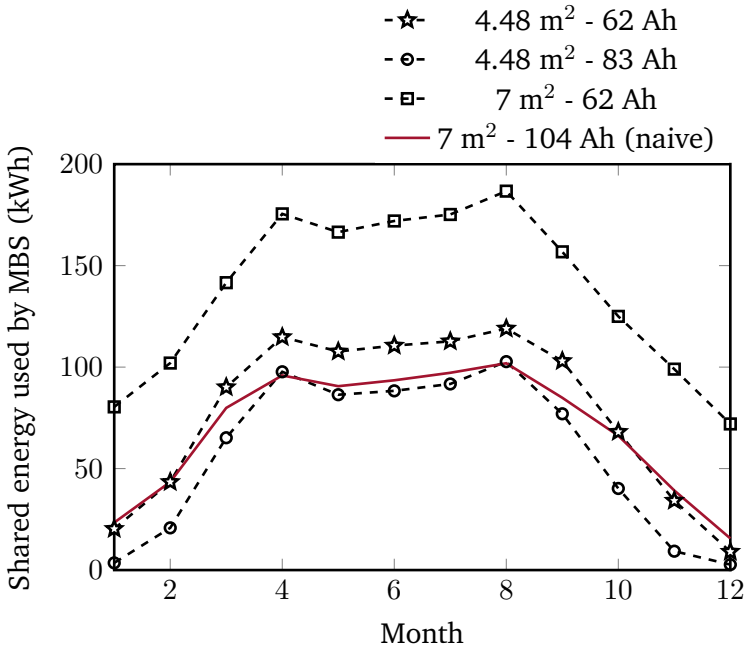


Figure 7.4: Energy shared by the SBSs and used by the MBS per month when considering different deployment sizes.

Table 7.2: Energy savings and costs for different deployment dimensions

Configuration		Energy (kW)	Costs ($)		
panel (m²)	battery (Ah)	1yr consumption	CAPEX	cost [5yrs]	cost [10yrs]
Grid-only					
-	-	10,264	0	10,775	21,550
EH SBS + sharing					
⋆ 4.48	62	7,982 (-22%)	2,345	10,725 (-0.5%)	19,105 (-11%)
○ 4.48	83	7,160 (-30%)	2,541	10,060 (-7%)	17,578 (-18%)
□ 7.00	62	7,169 (-30%)	4,100	11,625 (+8%)	19,150 (-11%)

our scenario where SBSs are solely powered by solar panel plus battery and can share energy with the MBS (also referred to as *EH SBSs+sharing*). We report the grid energy consumption, the CAPEX and the monetary cost (i.e., CAPEX + OPEX) due to the harvesting/storage add-on after 5 and 10 years, for the different panel and battery dimensions introduced in the previous section. The values between brackets indicate the savings with respect to the grid-only scenario. We consider a cost of 1.17 $/W for the solar panel (which also includes the installation cost) and 131 $/kWh for the battery. Moreover, the energy purchased from the grid has a cost of 0.21 $/kWh in our calculation.

The additional harvesting/storage hardware jointly with the optimal load control and energy sharing method allows reducing the grid power consumption. The energy savings range between 22% and 30%, depending on the harvesting/storage size. Carbon footprint and OPEX are decreased accordingly. In particular, monetary cost

savings range between 0.5% and 7% after 5 years. The only exception is represented by the third harvesting/storage size (i.e., square), where 7 years are needed to reach the breakeven point. The results at 10 years show higher savings, ranging between 11% and 18%.

Table 7.2 provides useful insights on the energy and cost savings and permits the MNOs to design their harvesting/storage systems by considering the tradeoff between dimensions and economic cost. In general, the harvesting/storage system with the lowest CAPEX is not the most economically convenient option in the long run. The highest savings are achieved for deployments in the right side of the boundary region of Figure 7.2 (i.e., circle). With this design approach, the proposed method achieves the highest energy and cost savings by maintaining active the SBSs for longer periods and sharing less energy with the MBS compared to the other two harvesting/storage dimensions.

As a final remark, we can assume that higher revenues and savings can be achieved during the lifetime of the network in a near future considering that: i) equipment hardware is designed to be always more energy efficient, ii) the actual market trends show a decreasing cost of the solar panels and batteries, and increasing prices of the grid energy, iii) future radio access networks will be ultra-dense and longer offloading periods may occur due to the higher number of SBSs.

7.6 Conclusions

In this chapter, we have extended the optimal load control method discussed in the previous chapter by introducing energy sharing capabilities that allows to efficiently use the renewable energy coming from distributed sources and to facilitate the off-grid operation of the RAN. The proposed approach enables the dynamic switch ON/OFF of portions of the SBSs. Energy exceeding the battery capacity is managed to be used by the MBS operations and further reduce the energy drained from the power grid. Software simulations demonstrated that an intelligent renewable energy management is essential to reduce the harvesting/storage system dimension with respect to naive approaches and leads to high energy and cost savings for a MNO.

8 Online Direct Load Control plus Energy Sharing

In this chapter, we investigate on the control of the sleep modes plus energy sharing among the BSs by a centralized agent, which implements ML techniques and is capable of learning how to efficiently operate the network. We discuss on the achieved performance, complexity and feasibility of different ML approaches. In particular, we study supervised and reinforcement learning models. The proposed supervised approach, called Imitation Learning, learns the optimal policies from labeled data collected by a supervisor. The creation of the training set is a hard task since it relies on an expert supervision. For this reason, Imitation Learning is generally feasible in scenarios with a limited number of SBSs due to complexity issues. Instead, solutions based on RL learn from the interaction with the environment and no specific supervision is required. However, a Markov Decision Process has to be properly defined together with a reward function to fit the optimization problem that must be solved [92]. In this work, we tailor two RL methods based on tabular and deep models, respectively.

The tabular RL method, i.e., Q-learning, is thought for discrete state/action spaces, and requires quantitation to deal with the continuous states of our scenario. Consequently, the number of quantitation levels have to be properly settled to find a reasonable trade-off between the approximation introduced and the performance in terms of cumulative reward, convergence time and memory footprint.

On the other hand, the Deep RL method, i.e., Deep Q-learning, uses ANNs as function approximation to address the quantization issue of the tabular method, and may handle a continuous state space. Moreover, the ANN tailored for our scenario requires smaller memory and achieves faster convergence than the tabular method. This translates into higher performance in terms of energy saving and system outage. Moreover, deep RL is able to control dense scenarios, i.e., a high (> 4) number of deployed SBSs, where the other models fail for complexity and memory issues.

Considering the literature reviewed in Section 2, the study presented in this chapter is the first focusing on centralized learning methods for the optimization of BS switch ON/OFF plus energy sharing policies. Moreover, the study reported here may also serve as a compendium of supervised and reinforcement learning approaches

for network optimization and their comparison in terms of system performance, implementation complexity, memory footprint and feasibility.

In summary, the contributions of the paper are listed below:

- We develop three different implementations of a centralized energy-aware RAN controller based on ML: namely Imitation Learning, Q-Learning and Deep Q-Learning. The training phase of those three different implementations is analyzed against different setups of the learning parameters.

- We analyze the load control plus energy sharing policies learned by the three different implementations of the agent and the achieved performance are compared in terms of grid energy consumption and dropped traffic.

- We provide an analysis of savings that a mobile operator may achieve by deploying a RAN powered with renewable energy and implementing our centralized learning controller in terms of monetary cost and energy footprint.

- We provide an analysis on how the different representations of the environmental state affects the training and their effects on the learned policies and system performance.

The rest of the chapter is organized as follows. In Section 8.1, we formulate the joint load control and energy sharing optimization problem. In Section 8.2, we discuss the proposed learning approaches, i.e., Imitation Learning, Q-Learning and Deep Q-Learning. In Section 8.3, we introduce the simulation scenario, discuss the setup of the parameters for the learning approaches, analyze the obtained policies and the achieved performance in terms of energy consumption and dropped traffic. In Section 8.4, we extend the analysis by considering different representations of the environmental state and their influence on the training, the learned policies and the system performance. Finally, in Section 8.5, we draw our conclusions.

8.1 Optimization Problem

The evolution of the system under study, and described in the previous chapter, may be defined by a Markov Decision Process as: $\boldsymbol{x}^{(t+1)} = f\left(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{w}^{(t)}\right)$, where $\boldsymbol{x}^{(t)}$ is the state of the system, $\boldsymbol{a}^{(t)}$ is the control and $\boldsymbol{w}^{(t)}$ is the random disturbance at time t (i.e., the randomness of the energy arrival and traffic processes).

The state of the system at time t is defined as $\boldsymbol{x}^{(t)} = \left[\boldsymbol{B}^{(t)}, h^{(t)}\right]$, where $h^{(t)} \in [0, 23]$ is the hour of the day at time t, while the system control is defined as $\boldsymbol{a}^{(t)} = \left[a_1^{(t)}, a_2^{(t)}, \ldots, a_N^{(t)}\right]$, where

$$a_i^{(t)} = \begin{cases} 0 & \text{if the } i\text{-th SBS is switched OFF} \\ 1 & \text{if the } i\text{-th SBS is switched ON.} \end{cases} \tag{8.1}$$

The optimization goal is to minimize the grid energy consumption, while keeping the percentage of dropped traffic experienced in the cluster below a threshold, D_{th}. This optimization problem can be formulated as:

$$\begin{aligned} \text{P1:} \quad & \min_{\{\boldsymbol{a}^{(t)}\}_{t=1,\ldots,K}} \sum_{t=1}^{K} \mathrm{E}^{\text{MBS}}(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{w}^{(t)}) \\ & \text{subject to } D^{(t)} < D_{\text{th}} \end{aligned} \tag{8.2}$$

where K is the time horizon or the number of times the control is applied, $D^{(t)}$ is the normalized traffic drop at time t and E^{MBS} is the normalized grid energy consumption drained by the MBS, given the operative modes of the SBSs at time t, whether ON or OFF.

We highlight here, that SBSs automatically switch OFF when the battery level is below a threshold B_{th}.

In this chapter, we adopt the offline solution introduced in Chapter 7 to compute performance bounds, and compare them with the policies learned through the proposed ML approaches.

8.2 Machine Learning Models

In this section, we discuss the different implementations of the centralized agent at the IEMS, namely Imitation Learning and the two variants of RL, Q-Learning and Deep Q-Learning.

8.2.1 Imitation Learning

Imitation Learning (IL) is a supervised approach that consists on learning from a set of labeled data provided by an external supervisor [78]. Each example in the training set is a description of the system state with a specification of the correct

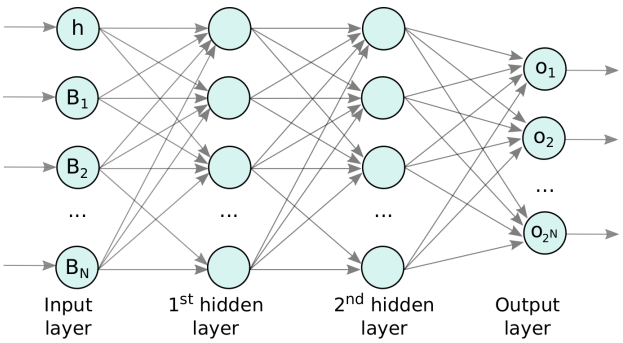


Figure 8.1: Example of a multiclass classifier based on ANNs.

action the agent shall take in that situation. In this case, the objective of the agent is to extrapolate (i.e., generalize) its response, so that it acts correctly in situations not included in the training set. Specifically, the agent is trained on the policies adopted by the optimal algorithm in Chapter 7. More in details, a multi-class classifier based on ANNs is used to learn a mapping from the state of the system $\boldsymbol{x}^{(t)}$ to the action $\boldsymbol{a}^{(t)}$. An example of the general ANN architecture for a multiclass classifier is shown in Figure 8.1. We consider the multilayer perceptron as basic architecture for the ANN, consisting on multiple fully connected layers of neurons [41]. The output of a neuron is computed using the following equation:

$$output = f_a\left(\sum_i (z_i \cdot w_i + b_i)\right), \tag{8.3}$$

where z is the input of the neuron, w is the weight of the connection to the neuron, b is the bias and $f_a(\cdot)$ is the activation function. The ReLU activation function (i.e., $f_a(z) = \max(0, z)$) is used for the hidden layers, whereas the SoftMax activation function is used for the output layer.

The SoftMax activation function takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. It is defined as

$$f_a(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{8.4}$$

where $\boldsymbol{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$ is the input vector. The standard exponential function is applied to each element z_i of the input vector $\boldsymbol{z}$. These values are then normalized by dividing them by the sum of all the exponentials. In this way, the sum of the components of the output vector $f_a(\boldsymbol{z})$ is 1. After applying the SoftMax function, each component in the output is in the interval $(0, 1)$, and the components add up to 1, so that they can be interpreted as probabilities.

The model is optimized using the categorical cross entropy loss function, defined as:

$$\mathcal{L}(y - \hat{y}_i) = -\sum_i y_i \log \hat{y}_i \tag{8.5}$$

where y is a grounded truth vector (i.e., the actions taken by the offline algorithm), and $\hat{y}$ is a vector of predictions. The detailed steps of the training algorithm are specified in Algorithm 5.

Algorithm 5 Training of the ANN classifier

Input: dataset $\mathcal{D}$
Output: classifier $c(\cdot)$
1: Collect all state/action pairs $\mathcal{D} \leftarrow < \boldsymbol{x}, \boldsymbol{a} >$
2: Train a classifier c using the dataset $\mathcal{D}$
3: **return** c

We consider a dataset containing observations collected in one year. A portion of 80% of the dataset is used for training the classifier, whereas the remaining 20% is used to validate the learned model.

The trained ANN is then implemented in the IEMS and manages the operative modes of the SBSs by the procedure described in Algorithm 6.

Algorithm 6 Imitation Learning agent

Input: classifier c
1: **for** $t = 1, \ldots, T$ **do**
2: $\quad \boldsymbol{x}^{(t)} \leftarrow$ current state of the system
3: $\quad \boldsymbol{a} \leftarrow c\left(\boldsymbol{x}^{(t)}\right)$
4: $\quad$ take action $\boldsymbol{a}^{(t)}$
5: **end for**

8.2.2 Reinforcement Learning

The objective of RL is to learn how to map the experienced situation (i.e., the state of the system) into the best action to take at every decision cycle t. By trying different actions, the agent learns the optimal behavior of the system that maximizes a cumulative reward. This phase is called exploration and is aimed at training the algorithm for the stable phase, called exploitation, in which the agent will use the learned policies. A good trade off between exploration and exploitation has to be maintained to assure that the system is continuously exploring new state/action pairs and updating its policies with actions that return higher rewards.

The IEMS implements an agent in charge of maintaining a policy and a Q-function $Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)})$, representing the level of convenience in taking the action $\boldsymbol{a}^{(t)}$ at time

step t, given that the system is in state $\boldsymbol{x}^{(t)}$. As a result of the execution of an action at time t, the environment returns a reward $r^{(t)}$, which is used to update the Q-values, $Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)})$. In our proposals, the reward given to the agent after taking action $\boldsymbol{a}^{(t)}$, being in state $\boldsymbol{x}^{(t)}$ at time t, is defined as:

$$r^{(t)} = -\mathrm{E}^{\mathrm{MBS}}\left(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}\right) + 1\left(D^{(t)} < D_{\mathrm{th}}\right), \tag{8.6}$$

where $1(\cdot)$ is the step function. The rationale behind this equation is to mimic the optimization problem in (8.2): the agent is getting a reward equal to 1 each time it is able to maintain the drop level below the threshold D_{th}, and a discount proportional to the consumed grid energy is applied. Moreover, in our implementations, an ϵ-greedy exploration policy is adopted: the learned action is taken with probability $1-\epsilon$ (i.e., exploitation), whereas a random action is taken with probability ϵ (i.e., exploration).

In this paper, we consider the QL and DQL algorithms described in Section 4.2.3, as instances of RL. Next, we detail the specific implementation of the two algorithms.

Q-Learning

In the standard QL algorithm, the Q-function is stored in a tabular form. The table describing the Q-function is therefore named Q-table. The reward $r^{(t)}$ is used to update the Q-value $Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)})$ according to the following rule:

$$\begin{aligned} Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}) \leftarrow & Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}) + \\ & \alpha[r^{(t)} + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{x}^{(t+1)}, \boldsymbol{a}') - Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)})], \end{aligned} \tag{8.7}$$

where α is the learning rate and γ is the discount factor.

The state variables need to be quantized due to the discrete nature of the Q-table. Therefore, the state of the system at time t is defined by $\hat{\boldsymbol{x}}^{(t)} = \left[\hat{\boldsymbol{B}}^{(t)}, h^{(t)}\right]$, where $\hat{\boldsymbol{B}}^{(t)} = \left[\hat{B}_1^{(t)}, \ldots, \hat{B}_N^{(t)}\right]$ is a vector representing the quantized values of the battery levels of the SBSs in the cluster and h is a variable representing the hours of the day. The detailed steps of the adopted QL algorithm are listed in Algorithm 7.

Deep Q-Learning

In the DQL approach, the Q-function is estimated by using a ANN approximator [73], as described in Section 4.2.3. The Q-function is approximated by the function

Algorithm 7 Q-Learning algorithm

Initialize $Q(\boldsymbol{x}, \boldsymbol{a}) \forall \boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{a} \in \boldsymbol{A}$ arbitrarily
for episodes $= 1, ..., M$ **do**
  Initialize $\boldsymbol{x}^{(1)}$
  for $t = 1, \ldots, T$ **do**
    Select action $\boldsymbol{a}^{(t)} = \max_{\boldsymbol{a}'} Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}')$ with probability $1 - \epsilon$ otherwise take a random action with probability ϵ.
    Execute action $\boldsymbol{a}^{(t)}$ and observe the reward $r^{(t)}$ and the next state $\boldsymbol{x}^{(t+1)}$
    Update the Q-value by using (8.7)
  end for
end for

$Q\left(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)} | \theta\right)$, where θ represents the ANN parameters. The state of the system is the input of the ANN, whereas the output layer corresponds to the predicted Q-values of the individual action for the input state. In details, the number of neurons in the input layer is equal to $N + 1$, i.e., the hour and the values of battery level of all the SBSs, whereas the number of neurons in the output layer is equal to 2^N, i.e., all possible combinations of ON/OFF operative modes of the SBSs in the cluster.

The detailed steps of the DQL algorithm are listed in Algorithm 8.

Algorithm 8 Deep Q-Learning algorithm

Initialize the replay memory $\mathcal{R}$ to capacity L
Initialize the ANN with random weights
for episodes $= 1, ..., M$ **do**
  Initialize state $\boldsymbol{x}^{(1)}$
  for $t = 1, \ldots, T$ **do**
    Select action $\boldsymbol{a}^{(t)} = \max_{\boldsymbol{a}'} Q(\boldsymbol{x}^{(t)}, \boldsymbol{a}'; \theta)$ with probability $1 - \epsilon$ otherwise take a random action with probability ϵ.
    Execute action $\boldsymbol{a}^{(t)}$ and observe the reward $r^{(t)}$ and the next state $\boldsymbol{x}^{(t+1)}$
    Store the experience $(\boldsymbol{x}^{(t)}, \boldsymbol{a}^{(t)}, r^{(t)}, \boldsymbol{x}^{(t+1)})$ in $\mathcal{R}$
    Sample a random batch of l experiences $(\boldsymbol{x}^{(j)}, \boldsymbol{a}^{(j)}, r^{(j)}, \boldsymbol{x}^{(j+1)})$ from $\mathcal{R}$
    Set $y^{(j)} = r^{(j)}$ for terminal state $\boldsymbol{x}^{(j+1)}$ otherwise $y^{(j)} = r^{(j)} + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{x}^{(j+1)}, \boldsymbol{a}'; \theta)$
    Perform gradient descend step on $\left(y^{(j)} - Q(\boldsymbol{x}^{(j)}, \boldsymbol{a}^{(j)}; \theta)\right)$
  end for
end for

8.3 Numerical Results and Discussion

In this section, we discuss the numerical results achieved after our extensive simulation campaign. In particular, we present the performance of the training phase of the three ML models, analyze the learned policies and the energy shared in the micro-grid. After, we provide an analysis of the network performance in terms of the

Table 8.1: Simulation parameters

	Parameter	Value
Scenario	Solar panel size (m^2)	4.48 (16×16)
	Solar panel efficiency (%)	21
	Battery capacity (Ah)	83
	MBS transmission power (dBm)	43
	SBS transmission power (dBm)	38
	Bandwidth (MHz)	20
	Channel model	Okumura-Hata [40]
	Heavy users traffic (MB/h)	900
	Ordinary users traffic (MB/h)	112.5
	Heavy users percentage (%)	50
	Traffic profile	Residential, Office
IEMS	Battery threshold B_{th} (%)	20
	High drop level D_{th} (%)	10

grid energy consumption and the dropped traffic. Finally, we discuss the energy and monetary cost savings achieved by the proposed RAN architecture.

8.3.1 Simulation Scenario

The scenario considered in this analysis consists of a single cluster of 1 MBS placed in the middle of a 1 km^2 area and N SBSs randomly placed. In order to avoid overlap, the distance between SBSs is at least of 100 m, and SBSs have a maximum transmission power of 38 dBm. The UEs are deployed in a radius of 50 m from each SBSs to mimic a hot-spot scenario [2]. Each SBS is supplied by a solar panel of 4.48 m^2 area and a lithium ion battery of 83 Ah capacity. The solar panel consists of an array of 16×16 solar cells of Panasonic N235B solar modules that have a single cell efficiency of about 21%. These dimensions allow to fully recharge the batteries in a typical winter day, and are consistent with the results obtained in Chapter 7. The solar energy arrivals are generated according to the city of Los Angeles, following the approach described in Section 3.1. The traffic demand is modeled as described in Section 3.2, considering a heavy user ratio of 50%. Further details about the simulation parameters are given in Table 8.1.

Finally, the training of all the learning algorithms is performed by considering a cluster of 3 SBSs. The simulations have been run on a machine with an Intel®Core™ i5-6300U CPU @ 2.40GHz and 8 GB of RAM.

Table 8.2: Training and validation performance for different ANN architectures

Neurons per layer			Training			Validation		
1st	2nd	3rd	Loss	Accuracy	MA F-score	Loss	Accuracy	MA F-score
8	-	-	0,17	0,93	0,93	0,19	0,93	0,93
32	-	-	0,15	0,94	0,94	0,16	0,93	0,93
64	-	-	0,14	0,94	0,94	0,16	0,93	0,93
512	-	-	0,11	0,95	0,95	0,15	0,94	0,94
32	32	-	0,12	0,95	0,95	0,14	0,95	0,95
64	64	-	0,10	0,95	0,95	0,15	0,94	0,94
32	32	32	0,11	0,95	0,95	0,16	0,94	0,94

8.3.2 Imitation Learning Training

A dataset containing the offline policies computed over one year has been split into a training dataset and a validation dataset, as described in Section 8.2.1. The training dataset has been used to train ANN classifiers with different architectures using the Adam version of gradient descent [53], with the goal of minimizing the categorical cross entropy, introduced in Equation (8.5).

The loss, the accuracy and the macro average (MA) F-score measured on the training and validation set, are reported in Table 8.2. The accuracy is defined as the number of correct predictions divided by the total number of predictions whereas the MA F-score is computed as the average of the categories F-scores, which are:

$$\mathrm{F1} = 2 \cdot \frac{p \cdot r}{p + r}, \tag{8.8}$$

where p is the number of correct positive results divided by the number of all positive results (i.e., precision), whereas r is the number of correct positive results divided by the number of all the samples that should have been identified as positive (i.e., recall).

The ANN architecture with two hidden layers of 32 neurons is selected, based on the maximum accuracy, MA F-score, training velocity (=300 epochs). Adding more neurons/layers leads to an increment of the computational complexity, without any positive impact to the performance.

8.3.3 Q-Learning Training

An analysis on the influence of the learning parameters to the training performance is presented in this section for the QL algorithm. The QL algorithm exploration rate is set to $\epsilon = 0.9$ at the beginning of the training, and discounted by 10% at each episode, until reaching the minimum value of $\epsilon_{\min} = 0.05$. In this way, the

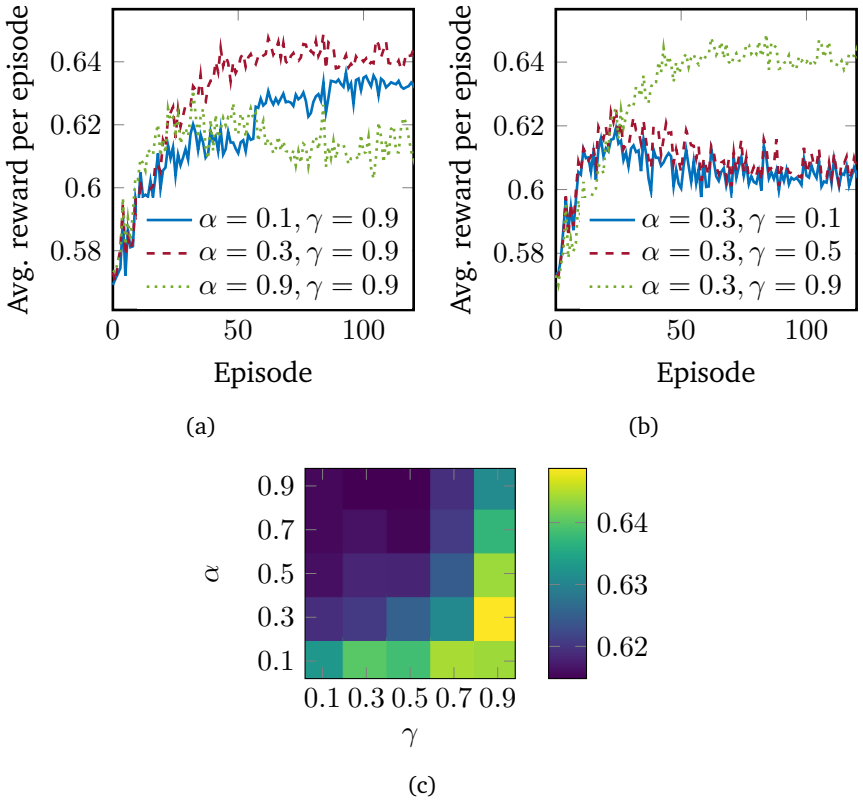


Figure 8.2: Average reward per episode of the QL algorithm when adopting different values of learning rate (a) and discount factor (b). Values of average reward per episode achieved at convergence are indicated in (c)

exploration is prioritized during the first phase of the training, enabling a faster convergence. Moreover, the battery levels $\boldsymbol{B}$ are quantized by considering a uniform quantizer with 3 levels, whereas the hour of the day h is a variable quantized with 24 levels. This choice represents a good trade-off between achieved performance and computational complexity, according to the performed simulations.

Figure 8.2 reports the performance obtained for different setups of the training parameters. The average reward per episode achieved by the QL agent for different values of the learning rate are shown in Figure 8.2(a). The highest reward is achieved for $\alpha = 0.3$. Using a smaller learning rate (e.g., $\alpha = 0.1$) makes the agent to converge slower to the maximum average reward, which is additionally slightly smaller than $\alpha = 0.3$. Instead, higher oscillations are experienced for a bigger learning rate (e.g., $\alpha = 0.9$), leading to a low average reward per episode.

Figure 8.2(b) shows the average reward per episode for three different values of the discount factor. The highest average reward is achieved for $\gamma = 0.9$, which implies that the system needs to be optimized over a long horizon. In fact, a big value of γ makes the agent looking for a long-term high reward by giving more importance to future rewards.

Finally, Figure 8.2(c) shows the average reward per episode achieved at convergence for different values of the learning rate α and discount factor γ. The highest average reward is achieved for $\alpha = 0.3$ and $\gamma = 0.9$.

8.3.4 Deep Q-Learning Training

An analysis of the influence of the learning parameters to the training performance is presented in this section for the DQL algorithm. In particular, Figure 8.3 reports the performance achieved for different values of the learning rate and the batch size.

Figure 8.3(a) shows the average reward per episode achieved by the DQL agent for three different values of the learning rate. The discount factor and the batch size are fixed to $\gamma = 0.9$ and $l = 20$, respectively. The value at convergence and the number of episodes needed to converge are influenced by the learning rate. In particular, big values of α make rapidly achieve a stable average reward per episode, whereas smaller learning rates lead to a slower convergence, but a higher average reward per episode in the long run.

Figure 8.3(b) shows how the training performance is affected by the batch size l. The average reward per episode shows high oscillations and a slow convergence if experience replay is not used (i.e., $l = 1$). Instead, enabling it allows to reduce the

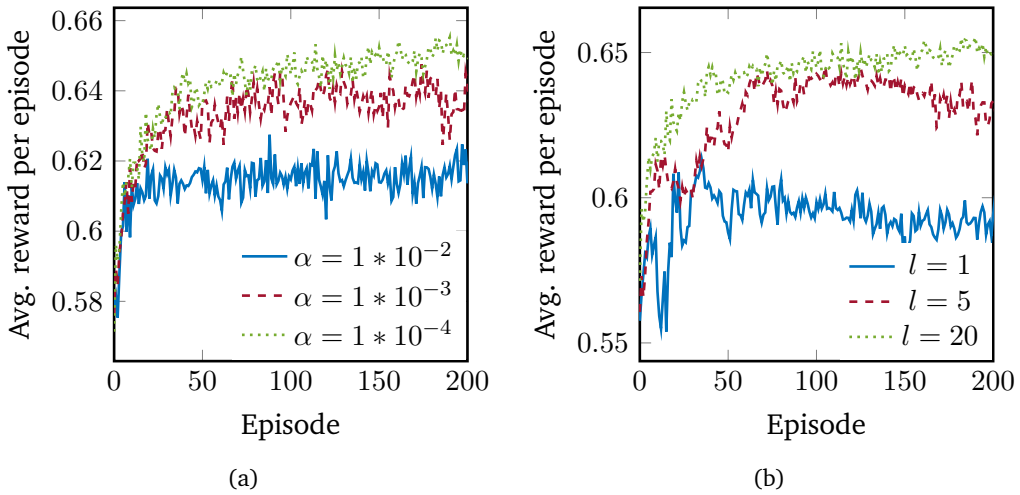


Figure 8.3: Average reward per episode of the DQL algorithm when adopting different values of learning rate (a) and batch size (b). The dimension of the batch is $l = 20$ in (a) whereas the learning rate is $\alpha = 1 \cdot 10^{-4}$ in (b). The discount factor is $\gamma = 0.9$.

magnitude of the oscillations and the number of epochs needed to converge. We highlight here that this does not come for free in terms of execution time. In fact, the running time per episode increases linearly with the dimension of the batch size l, since the number of forward and backward passes through the ANN scales linearly with l.

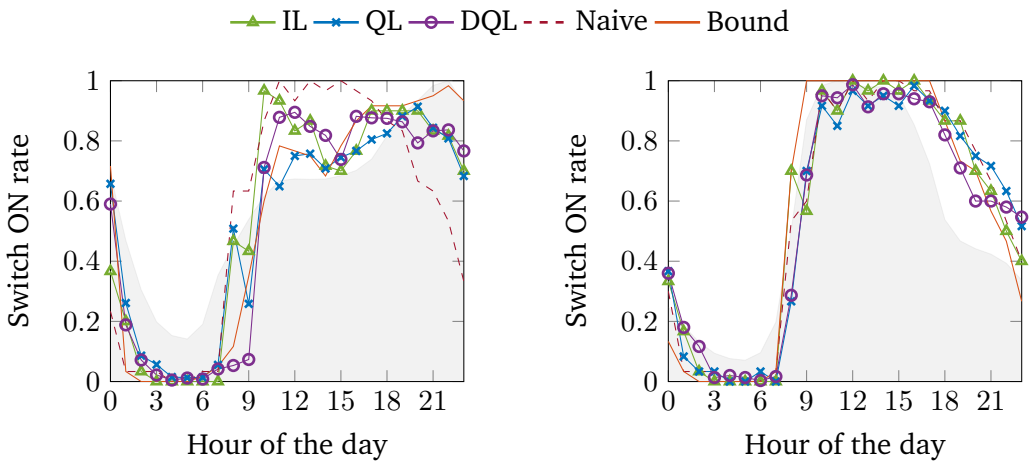
Finally, the average reward per episode achieved varying the numbers of neurons per layer and the number of layers is reported in Table 8.3. The average reward per episode increases with the dimension of the ANN till a performance limit. Increasing the dimension of the ANN only leads to longer training times without any positive influence on the average reward per episode. The best average reward per episode is achieved with an ANN architecture composed of two layers, both consisting of 50 neurons. Moreover, the best results have been obtained using the linear activation function (i.e., $f_a(x) = x$) for the output layer and the ReLU activation function for the input and hidden layers. We experimentally noticed that the property of ReLU of turning any negative argument into zero helps the convergence speed and the average reward per episode at convergence.

8.3.5 ON-OFF Policies

In this section, we analyze the daily average SBSs switch ON rate experienced by the three different implementations of the agent in a cluster of 3 SBSs. In particular, the switch ON rates experienced in a residential and an office area are reported in Figure 8.4 and Figure 8.5, respectively, for the months of December and July.

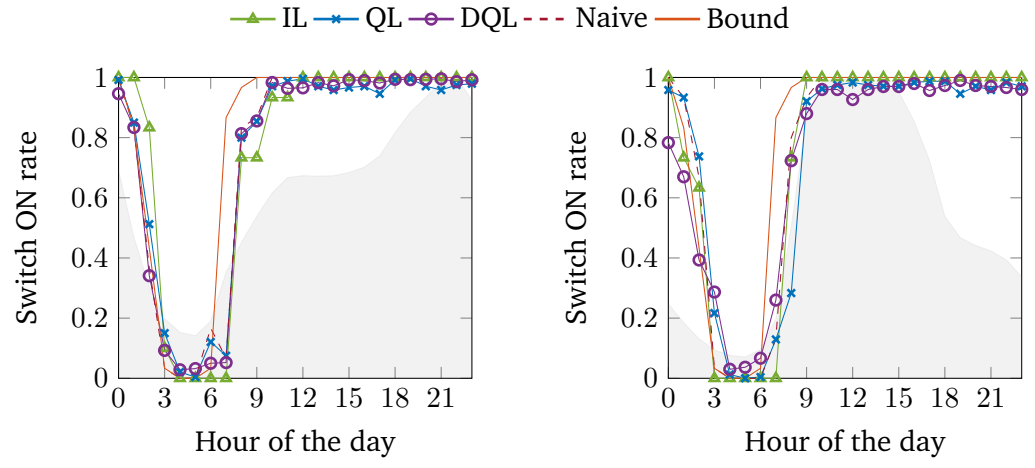
Table 8.3: Average reward per episode for different ANN architectures

Number of neurons			Avg. reward per episode
1st layer	2nd layer	3rd layer	
10	-	-	0.63
20	-	-	0.64
20	20	-	0.64
50	-	-	0.65
50	50	-	0.66
50	50	50	0.65
100	-	-	0.65
100	100	-	0.65



(a) Residential area (b) Office area

Figure 8.4: Average switch ON rate of the SBSs when adopting different policies in a residential (a) and office (b) area in the months of December. The shaded area represents the shape of the traffic demand.



(a) Residential area (b) Office area

Figure 8.5: Average switch ON rate of the SBSs when adopting different policies in a residential (a) and office (b) area in the months of July. The shaded area represents the shape of the traffic demand.

Those months represent the worst and the best case in terms of harvested energy, respectively. The shape of the traffic demand profiles is indicated by the shaded areas. The three agent implementations are compared with a naive approach that operates by turning OFF the SBSs when their battery levels go below the threshold B_{th} and switching them ON when the level is above it. The performance bound using the offline optimization is also reported in the figures as a reference. We refer to this policy as *Bound* policy.

In general, the policy works so that the SBSs save energy during nights, when the traffic is low, and provides service during the high traffic demand hours. The length of the night switch OFF period depends on the energy availability and, thus, on the season of the year. In particular, longer switch OFF periods are generally observed in December for both the traffic areas.

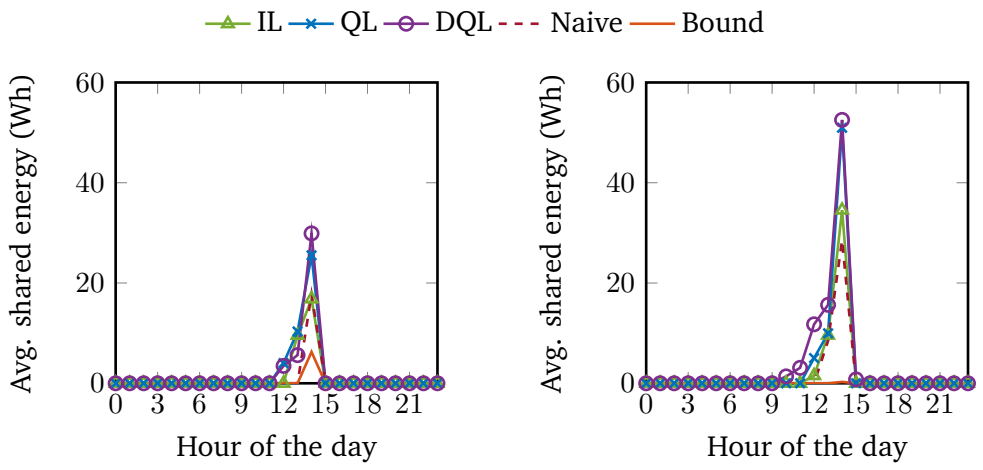
Figure 8.4(a) shows the switch ON rate for a residential area in the month of December.

The bound policy is characterized by a high switch ON rate of 0.94 in average during the traffic peak, i.e., from 6 pm to midnight, whereas an average switch ON rate of 0.75 is experienced from 10 am to 3 pm. This means that the bound policy partially switches OFF the SBSs during those hours to save energy needed to provide service during the peak hours.

The naive policy switches ON the SBSs as soon as sufficient energy is harvested in the morning. Therefore, high switch ON rate is measured from 10 am to 3 pm. i.e., in correspondence with the peak of the harvesting process. Then, as soon as the energy reserves fall below the threshold B_{th}, the switch ON rate decreases and reaches low values during the peak of the traffic demand, leading to high traffic drop.

The QL and DQL policies have an average switch ON rate of 0.7 and 0.8, respectively, between 10 am and 3 pm. Those values highlight that the adopted policies follow the behavior of the bound policy: they partially switch OFF the SBSs during the daytime and maintain the switch ON rate to an average of 0.8 during the peak of the traffic demand. The same behavior is reproduced by the IL algorithm. We highlight here that the behavior of the QL, DQL and IL policies follows the behavior of the bound policy with some differences due to the stochastic nature of the energy and traffic processes.

Figure 8.4(b) shows the switch ON rate for an office area in the month of December. The office traffic profile is easier to manage, since the peak of the traffic demand coincides with the period of more intensive energy arrivals (i.e., around midday). In



(a) Residential area

(b) Office area

Figure 8.6: Average amount of shared energy used by the MBS in December when adopting different policies in a residential (a) and office (b) area.

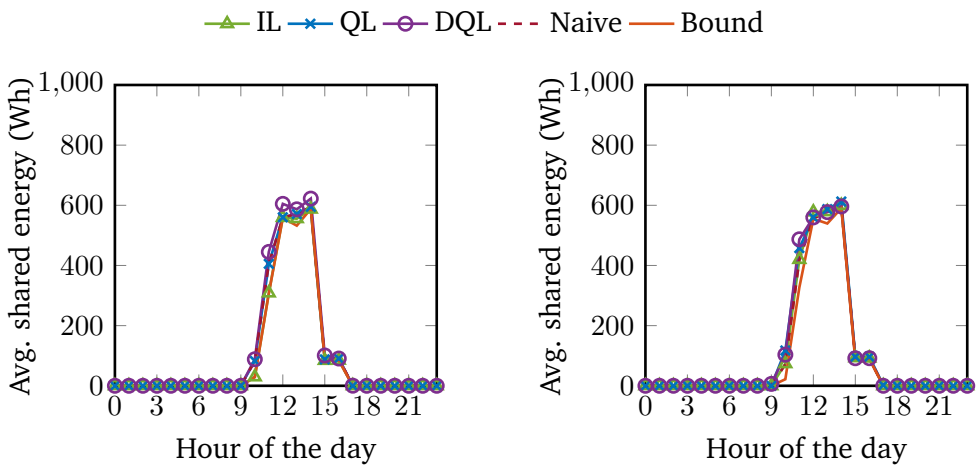
all the studied cases, the SBSs are switched ON after the night sleep starting from 7 am, and a high switch ON rate is experienced from 10 am to 5 pm for all the policies (i.e., during the peak of the traffic demand).

Finally, Figure 8.5(a) and Figure 8.5(b) show the switch ON rate in the month of July for the residential and the office areas, respectively. In both the cases, bound policies show an increase of the switch ON rate after 6 am, whereas in the case of the naive and learning agents, it starts raising from 7 am. No other particular difference from the studied algorithms is noticed due to the high availability of harvested energy in summer, which allows the agent to keep the SBSs ON until midnight in both the traffic areas regardless the specific implementation.

8.3.6 Shared Energy Assessment

In this section, we analyze the behavior of the different implementations of the agent in terms of the daily energy shared between SBSs and MBS. The average amount of energy shared with the MBS and used for its operations is reported in Figure 8.6 and Figure 8.7 for the month of December and July, respectively. Results are provided for both a residential and an office area.

In general, the energy is shared around noon during a period of time whose duration depends on the season of the year. The energy in December is shared between 11 am and 3 pm, whereas in July is shared from 10 am to 4 pm, which is longer since during summer the SBSs harvest more solar energy.



(a) Residential area

(b) Office area

Figure 8.7: Average amount of shared energy used by the MBS in July when adopting different policies in a residential (a) and office (b) area.

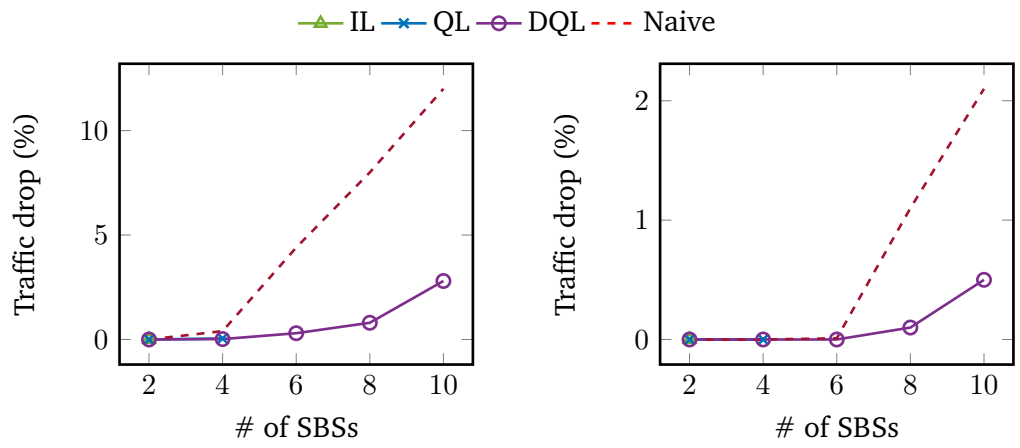
The bound policy shares less energy with respect to the other policies in all the considered scenarios. This is motivated by the fact the bound policy maximizes the amount of traffic which is offloaded to the SBSs and, thus, the local use of the harvested energy. This behavior is confirmed by the high switch ON rates in Figure 8.4 and Figure 8.5, and is made possible by the prior knowledge of the temporal variation of the energy and traffic processes.

The naive policy shares less energy than the three implementations based on ML. In fact, the naive policy maintains the SBSs active as long as their energy is available. This policy reduces the probability of filling the batteries and, thus, of sharing the excess energy.

The IL, QL and DQL policies show similar behaviors. In particular, the DQL agent is that leading to higher energy sharing with an average of 39 Wh and 89 Wh in a residential and office area, respectively, for a day of December. Higher values are measured in the month of July, where DQL is sharing an average of 2.54 kWh and 2.51 kWh per day in the residential and office area, respectively.

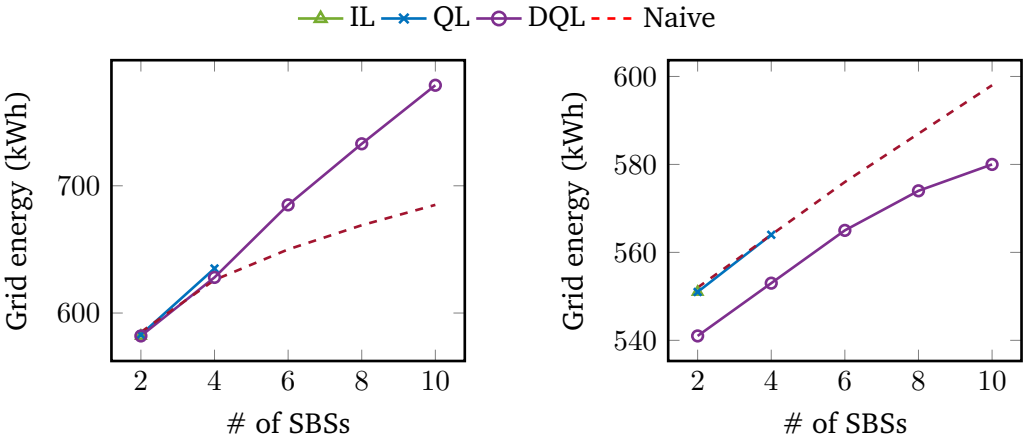
8.3.7 Network Performance

In this section, we compare the three learning algorithms in terms of grid energy consumption and traffic drop. The results have been obtained by setting each algorithm with the best configuration of the training parameters, as discussed in Sections 8.2-4. In particular, we focus on the performance achieved in the month of December (i.e., the worst in terms of harvested energy).



(a) Residential area (b) Office area

Figure 8.8: Percentage of dropped traffic experienced in the month of December, when adopting the naive policy and the policy obtained by the proposed learning approaches.



(a) Residential area (b) Office area

Figure 8.9: Grid energy consumption experienced in the month of December, when adopting the naive policy and the policy obtained by the proposed learning approaches.

Figure 8.8 and Figure 8.9 show the traffic drop and grid energy consumption performance achieved when considering different cluster sizes in a residential and office area, respectively. In particular, we consider a scenario with 100 UEs deployed in the area of each SBSs, such that the traffic demand linearly increases with the number of deployed SBSs in the cluster.

The IL approach cannot be adopted in scenarios with more than 3 SBSs due to the high computational complexity and the amount of memory required by the optimal algorithm and the resulting unavailability of training data. Similarly, the QL approach cannot be adopted in scenarios with more than 4 SBSs, due to the high amount of memory required to store the Q-table, which increases exponentially with the number of SBSs (i.e., $\mid \hat{B}_i^{(t)} \mid^N \times \mid h^{(t)} \mid \times 2^N$). On the contrary, DQL limits this

problem thanks to the introduction of the ANN function approximation, in which the number of weights that needs to be estimated is lower than the number of Q-table elements estimated by QL. This makes the memory footprint of the DQL controller much lower, and allows its operations for a network with a higher number of SBSs in the cluster.

Figure 8.8 shows the traffic drop experienced in a residential and an office area. The ML agents reach the same performance in the scenario with 2 SBSs, whereas DQL reduces the drop by 60% with respect to QL in the scenario with 4 SBSs in a residential area. This is mainly due to the possibility of using continuous state variables as input of the ANN and avoid the quantitation errors of QL.

IL, QL and DQL algorithms lead to lower traffic drop than the naive approach for all the considered cluster dimensions. In the case of DQL, this aspect is more evident for deployments with more than 4 SBSs in the residential area and 6 SBSs in the office area. In those cases, the traffic drop experienced by the naive policy linearly increases with the number of SBSs. We highlight here that DQL reaches significant savings up to 90% with respect to the naive algorithm for both traffic profiles.

The amount of grid energy consumed by the MBS in the month of December is reported in Figure 8.9, for the residential and the office traffic profiles. DQL saves up to 3% and 1% of the grid energy than IL, QL for all the considered cluster dimensions in office and residential area, respectively. This translates into a high energy efficiency, since DQL is able to serve more traffic. We highlight that the naive policy consumes less grid energy when considering a cluster with more than 4 SBSs in a residential area. This is motivated by the high amount of traffic that is not served by this policy.

8.3.8 Energy Savings and Cost Analysis

In this section, we provide an analysis of the grid energy consumed in one year and monetary costs experienced after five and ten years for different cluster dimensions that an operator may achieve through the implementation of the proposed DQL algorithm. They are reported in Table 8.4 considering two different traffic areas, namely residential and office. Note that in this analysis a trained version of the DQL agent is implemented in a new environment, characterized by instances of the energy arrival and traffic demand processes that are different from those used during the training phase. In such a way, we generalize the behavior of the agent and get performance as close as a real implementation in an operative network.

Table 8.4: Energy and cost analysis

SBSs	Grid Energy (kWh)		Cost at 5 years ($)		Cost at 10 years ($)	
	Grid-only	DQL	Grid-only	DQL	Grid-only	DQL
Residential						
2	8,312	5,753 (-31%)	8,728	7,735 (-11%)	17,455	13,775 (-21%)
4	10,580	5,932 (-44%)	11,109	9,617 (-13%)	22,218	15,845 (-29%)
6	12,848	6,292 (-51%)	13,490	11,689 (-13%)	26,981	18,295 (-32%)
8	15,116	6,560 (-57%)	15,872	13,664 (-14%)	31,744	20,552 (-35%)
10	17,384	7,236 (-58%)	18,253	16,068 (-12%)	36,506	23,666 (-35%)
Office						
2	8,301	5,500 (-34%)	8,716	7,469 (-14%)	17,432	13,244 (-24%)
4	10,554	5,512 (-48%)	11,081	9,617 (-17%)	22,163	14,963 (-32%)
6	12,806	5,536 (-57%)	13,447	10,895 (-19%)	26,893	16,708 (-38%)
8	15,059	5,504 (-63%)	15,812	12,555 (-21%)	31,624	18,334 (-42%)
10	17,312	5,576 (-68%)	18,178	14,325 (-21%)	36,355	20,180 (-44%)

The performance achieved by DQL are compared with a network scenario in which all the BSs are supplied by the power grid, also referred to as *grid-only*. We consider a cost of 1.17 $/W for the solar panel (which also includes the installation cost) and 131 $/kWh for the battery. Moreover, the grid energy has a cost of 0.21 $/kWh.

The harvesting/storage hardware jointly with the centralized network control based on DQL allows to reduce the grid energy consumption in both the traffic areas. The savings increase as the network deployment is more dense (i.e., more SBSs in the cluster): from 31% to 58% higher than the grid-only case in the residential area and from 34% to 68% in the office area. OPEX is decreased accordingly. Costs after 10 years are significantly reduced up to 35% and 44% in the residential and the office area, respectively.

8.4 Discussion on the Environmental Modeling

A key issue in RL is to define a proper model of the environment sensed by the agents, so that they may accurately capture the system dynamics and learn how to optimally interact with it. In this section, we extend the analysis of the DQL algorithm by investigating on different representations of the environmental state. In particular, we provide an analysis of the state representations effect to the RL training phase, its policies and to the resulting system performance.

We considered the following representations of the environment:

- *B*: $\boldsymbol{x} = [\boldsymbol{B}]$
- *HBL*: $\boldsymbol{x} = [\boldsymbol{H}, \boldsymbol{B}, \boldsymbol{L}]$

- *Bh*: $\boldsymbol{x} = [\boldsymbol{B}, h]$

- *HBLh*: $\boldsymbol{x} = [\boldsymbol{H}, \boldsymbol{B}, \boldsymbol{L}, h]$

where $h \in [0, 23]$ represents the hour of the day in which the measurements are collected. The rationale behind these choices is to model the environment with an incremental number of variables that may (or may not) lead to a better representation of the dynamic processes that characterize the system under study.

8.4.1 Training Performance

In this section, we analyze the influence of the environment state representation on the algorithm training when considering a cluster of 5 SBSs. In particular, the agent parameters have been set to the values that provide the best average reward per episode according to simulations. In details, the learning rate is $\alpha = 10^{-4}$, the discount factor is $\gamma = 0.9$ and the size of the experience replay batch is $l = 20$. The exploration rate is set to $\epsilon = 0.9$ at the beginning of the training and discounted by 10% at every training episode, until reaching the minimum value of $\epsilon_{\min} = 0.05$.

Figure 8.10 shows the average reward per episode for the different representations. The highest is achieved by the Bh and HBLh representations: both of them reach 0.546 on average after 160 episodes. The EBL representation reaches 0.542 on average after 120 episodes, whereas the B representation reaches 0.540 on average after 30 episodes. Therefore, we can note that: i) increasing the number of state variable does not necessarily improve the average reward per episode, and ii) including the hour of the day in the environmental model increases the average reward, at the expenses of a longer training phase.

8.4.2 ON-OFF Policies

In this section, we analyze the policy selected by the agent when considering the different representations. We adopt the switch ON rate as a metric to describe the behaviors of the SBSs, and we focus on the month of December since it represents the most challenging month for learning a policy due to the scarce availability of the harvested energy.

The daily average switch ON rate of the SBSs is reported in Figure 8.11. The shape of the residential traffic profile is also depicted. A common behavior can be observed regardless the environmental model adopted. The SBSs switch OFF during night hours, when the traffic is low, to save the stored energy. Then, the SBSs are gradually

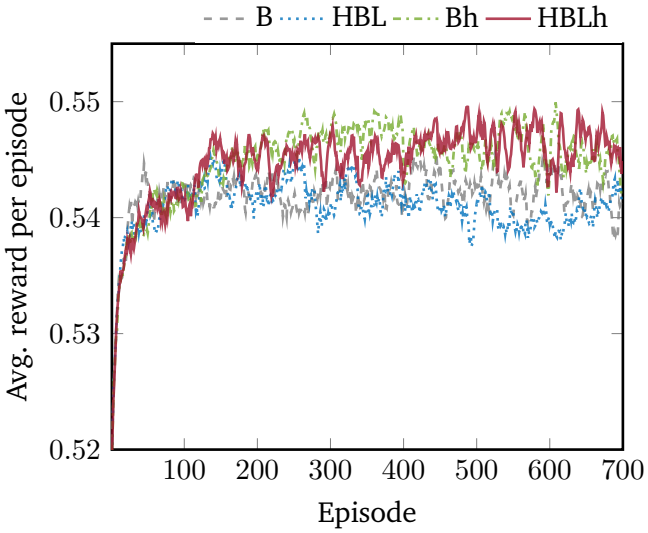


Figure 8.10: Average reward per episode when adopting different representations of the environment state. The learning rate is $\alpha = 10^{-4}$, the discount factor is $\gamma = 0.9$ and the dimension of the batch is $l = 20$.

switched ON, starting from 7 am to provide service during high traffic demand hours. However, the B and HBL policies show a higher average switch ON rate between 1 am and 7 am than the Bh and HBLh policies. Moreover, the B policy maintains the switch ON rate to an average of 0.9 between midday and 6 pm. The HBL, Bh and HBLh policies maintain the switch ON rate to an average of 0.80, 0.79 and 0.78 in the same period, respectively. In particular, a local minimum is experienced at 5 pm. Finally, the different policies show different average switch ON rates also during the peak of the traffic (i.e., between 7 pm and midnight). In details, the B and HBL policies maintain an average switch ON rate of 0.65 and 0.67, respectively. On the other hand, the Bh and HBLh reach higher values, respectively 0.74 and 0.76.

8.4.3 Energy Consumption and Network Performance

The effects of the learned policies on the network performance are discussed in this section. In particular, the grid energy consumption and the traffic drop experienced in the month of December are reported in Table 8.5, considering the different representations of the environment.

The worst performance in terms of traffic drop is achieved by the B representation. The HBL representations leads to 6% less drop than the B representation, at the price of a small increase in the grid energy consumption. The best performance is achieved by the HBLh representation, which leads to the smallest values of grid energy and traffic drop. The Bh representation returns very similar values of grid energy and traffic drop. This suggests that a simple model including the battery

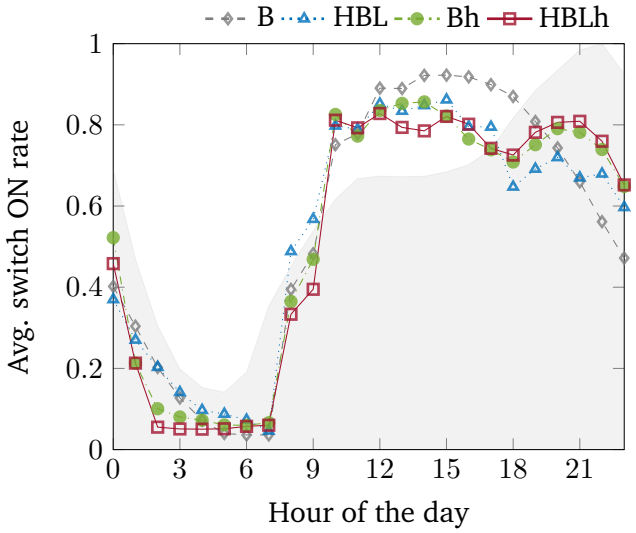


Figure 8.11: Average switch ON rate of the SBSs when adopting different policies in the month of December. The shaded area represents the shape of the traffic demand.

Table 8.5: Grid energy consumption and traffic drop performance achieved at convergence when using different representations of the environment

State definition	Grid energy [kWh]	Traffic drop [GB]
$[\boldsymbol{B}]$	651.4	108.6
$[\boldsymbol{H}, \boldsymbol{B}, \boldsymbol{L}]$	652.1	101.1
$[\boldsymbol{B}, h]$	647.9	72.2
$[\boldsymbol{H}, \boldsymbol{B}, \boldsymbol{L}, h]$	647.6	70.1

level and the hour of the day is sufficiently good to capture the dynamics of the environment and return high performance.

8.5 Conclusions

In this chapter, we have proposed a RAN setup in which a hierarchical cell structure is deployed within the same geographical area with BSs of different scale factors, transmission power, computational capabilities and coverage areas. This federation of BSs together with the distributed harvesters and storage devices at SBSs sites form a micro-grid, whose operations are managed by an energy management system in charge of controlling the intermittent and erratic energy budget from the RESs. We have focused on the design of online algorithms capable of jointly control sleep mode and energy sharing within the micro-grid. Three different implementations of ML models are proposed for the central agent, namely Imitation Learning, Q-Learning and Deep Q-Learning. We have discussed on the achieved performance, complexity and feasibility of those different ML approaches.

The DQL algorithm tailored for our scenario presents higher performance in terms of energy saving and system outage, and is able to control highly dense scenarios, where the other approaches fail due to complexity and memory issues. The energy and cost analysis provides an insight on the greater savings that can be achieved by an operator implementing the proposed DQL algorithm in the central controller of the RAN setup with energy harvesting capabilities.

Finally, we analyzed the performance of the DQL approach when considering different representations of the environment. Numerical results show that the adopted representations affect both the number of training episodes needed to converge and the asymptotic values of convergence. Moreover, the state representations may produce different operative policies, since they change the way the agent senses the environment. The performance of the system under study show that including the hour of the day in the state representation is fundamental to efficiently reduce both the energy consumption and the traffic drop.

9 Conclusions

In this final chapter, which conclude the dissertation, we provide a summary of the achieved results and what we consider to be the most promising directions for the related future works.

The chapter is organized as follows. In Section 9.1, we discuss the results achieved in this dissertation. Then, in Section 9.2 we introduce the related future works.

9.1 Summary of the Results

The goal of this dissertation is to contribute on the design of sustainable mobile networks. In particular, we focused on the analysis of the performance of optimal and online control algorithms for HetNets with energy harvesting capabilities. Moreover, we investigated on the possibility of intelligently routing energy in a micro-grid of interconnected conventional/renewable energy sources and loads.

The dissertation is divided into one introductory part and 4 main parts:

1. Chapter 1, Chapter 2, Chapter 3, Chapter 4 introduce the investigated problem and provide a discussion on the state-of-the-art literature. Moreover, the mathematical models of the system under studying is proposed and an introduction on the adopted mathematical framework is given.

2. Chapter 5 presents the first technical contribution part on the characterization of the solar energy source.

3. Chapter 6, presents the second technical contribution part on the design of an optimal load control solution for SBSs powered with solar panels.

4. Chapter 7 presents the third technical contribution part on the definition of a optimal load control and energy sharing solution for SBSs powered by solar panels.

5. Chapter 8 presents the fourth technical contribution part on the design of online ML algorithms capable of jointly control sleep mode and energy sharing within a micro-grid of interconnected BSs.

In the next sections, we summarize the contributions and the conclusions of the fourth main technical parts (i.e., items 2-5).

9.1.1 Characterization of the Solar Energy

The first technical section of this dissertation proposes a method based on unsupervised learning to extract hidden features of the solar energy generation. Moreover, clustering performed on the extracted features allows grouping geographical locations according to their similarities in terms of solar energy generation.

The results analyzed in the chapter show that clustering the extracted features provides a valid support for learning latent variables of the solar energy generation process. This approach can therefore be adopted to have a more detailed characterization of different locations with respect to state-of-the-art solar maps that relies on a single metric (e.g., annual solar radiation).

9.1.2 Optimal Direct Load Control

Current trends anticipate that future mobile networks will be composed of dense deployments of heterogeneous BSs that will allow meeting the foreseen huge traffic demand. New generation networks are also going to be very demanding in terms of energy consumption from the power grid. The high power requirements suggests that RESs can be adopted to reduce the environmental impact of mobile networks while also enabling cost savings from the MNO perspective. However, the adoption of RESs entails higher management complexity do to the inherently erratic and intermittent nature of this sources, which may cause fluctuating energy inflow and produce system outage.

In the second technical part we have therefore introduced an optimal DLC of a two-tier mobile network based on DP. We have provided a graphical representation of the problem and introduced an algorithm based on the shortest path method to find the optimal ON/OFF policy for the SBSs.

The analysis of the results achieved by the proposed algorithm provided a first insight on the characteristics of the optimal ON/OFF policies and how those are affected by the energy and traffic processes. In particular, we analyzed the ON/OFF

policies for a winter and a summer month in three different traffic areas. Then we compared the achieved performance with the one of a naive algorithm in terms of grid energy consumption and system traffic drop. In particular, a further analysis on the exceeding energy available at the SBS sites in summer months suggested that this energy may be shared with the MBS to further reduce its grid energy consumption and therefore increase the mobile network self-sustainability.

9.1.3 Optimal Direct Load Control plus Energy Sharing

In the third technical part, we extended our analysis by investigating a scenario in which BSs within the same geographical area are grouped in a micro-grid. The micro-grid enables the efficient use of the renewable energy coming from distributed sources and facilitates the off-grid operation of the RAN.

In particular, we targeted the design of an optimal load control method with energy sharing within the micro-grid, and we solved the optimization problem by using the approach introduced in the previous technical part.

We studied the dimensioning of the energy harvesting and storage devices, and we analyzed how the dimension of the adopted deployment affects the optimal policies, the network performance and the monetary cost of a MNO. The numerical results demonstrate that the intelligent renewable energy management is essential to reduce the dimensions of the harvesting/storage devices and that it also leads to high energy and cost savings for a MNO.

9.1.4 Online Direct Load Control plus Energy Sharing

In the last technical part, we investigated the control of sleep modes and energy sharing among the BSs by a centralized agent which implements ML techniques and is capable of learning how to efficiently operate the network. We proposed three different implementations of ML models based on supervised and reinforcement learning, and we analyzed their performance, complexity and feasibility.

The proposed DQL algorithm achieved better performance in terms of energy saving and system outage and was able to control dense scenarios, where other approaches failed due to memory and complexity issues.

Moreover, we extended the analysis of the DQL algorithm by investigating how the representation of the environment affects the training, the policies and the performance of this approach. Numerical results showed that including the hour of

the day in the environment state representation is fundamental for the design of a control algorithm that allows efficiently reducing both the energy consumption and the traffic drop.

9.2 Future Works

The sustainability of future mobile networks still represents an important challenge for the coming years. In this dissertation, we proposed and analyzed new approaches that were not covered in the literature. In the following, we provide some open issues that we identified while working on this topic. In particular, we focus on what we consider being the most promising research lines that should be investigated in the following years.

9.2.1 Characterization of the Solar Energy Generation

In Chapter 5, we introduced an approach based on feature extraction to learn hidden features of the solar energy generation process. This method allowed obtaining a classification of geographical locations that consider different characteristics of the solar energy generation process, going beyond state-of-the-art solar maps.

The input of the adopted feature extraction methods consisted of daily traces of solar energy generation. We believe that the adoption of additional input variables may allow obtaining a more detailed characterization of the process. In particular, the predictability of the solar generation process is an important factor to consider in the future, due to the development of control algorithms that relies on the prediction of the energy harvesting process. Weather forecast data could be included as an input of the adopted feature extraction method. In this way, cities with the same solar energy characteristics (according to what reported in Chapter 5) could be classified into different clusters based on the different level of predictability (e.g., cities in which the weather conditions are mostly stable and cities with high variations would fall into two different clusters).

Finally, the application of our proposed method to characterize other RESs (e.g., wind) needs to be investigated. Moreover, feature extraction performed on datasets merging data from different types of RES could provide a more complete characterization of the cities, based on their overall harvesting capabilities. This classification is expected to be useful when considering the deployment of heterogeneous RESs.

9.2.2 Updated and Flexible Power Consumption Models

In the last decade, Cloud Computing has emerged as a new paradigm characterized by the idea of a centralization of computing, storage and network management implemented in the Clouds (i.e., datacenters, core networks, etc.). However, in recent years, a new trend has emerged with cloud functions moving towards the edges of the network. This new paradigm is named Mobile Edge Computing (MEC) and its main feature is to push computing functions and storage to the network edges (i.e., BSs and access points) to allow intensive computation and low latency [65]. Therefore, the processing component of future BSs power consumption is expected to be more relevant and it needs to be accurately modeled. Moreover, new models should be versatile, allowing the estimation of the power consumed by the developed network control algorithms (e.g., our proposed direct load control centralized controller).

9.2.3 Uncoupled BSs - RESs

In Chapter 6, we highlighted how energy sharing among BSs represents a valid method to reduce the mobile network grid energy consumption. We believe that this approach introduces advantages that need to be further investigated. In particular, the introduction of energy sharing capabilities allows removing the requirement of having a solar panel located at every SBS site. We believe that in some scenarios (e.g., SBSs with low traffic demand), having one solar panel per SBS may be less energy and cost effective than having less solar panels of bigger dimensions. Therefore, an analysis of the costs and benefits experienced when considering different amounts and dimensions of the harvesting devices could be performed similarly to what has been done in Chapter 7. Furthermore, the harvesting devices could also be uncoupled from the SBSs and deployed in the proximity of them, taking advantage of bigger spaces.

9.2.4 Grid of Heterogeneous Sources and Loads

The integration of heterogeneous sources and loads may allow reaching high energy savings thanks to the cooperation between different elements of the micro-grid.

In particular, the deployment of heterogeneous RESs (e.g., solar panels and wind turbines) needs to be investigated since this approach is expected to increase the micro-grid self-sufficiency, taking benefit from the low correlation between the different energy processes.

On the other hand, different types of load can be introduced into the framework to be managed by the proposed centralized controller. We consider, for instance, the emerging concept of intelligent street lighting[56, 74] based on the deployment of smart lampposts. These devices are going to implement presence detectors, to reduce the lamp power consumption when light is not needed. Therefore, the lamp load process is going to be stochastic and dependent from different factors (e.g., available natural light, people presence, etc.). The management of new loads entails a higher complexity of the controller, but it open new possibilities for a more efficient use of the harvested energy.

9.2.5 Learning Agents and Complexity

In Chapter 8, we proposed three centralized controller implementations based on supervised and reinforcement learning. Different learning algorithms and neural networks architectures need to be investigated since they could provide an improvement in terms of performance and complexity. In particular, the application of recurrent neural networks as function approximators for the DQL approach needs to be studied since it may allow taking advantage of the sequential characteristics of the energy and traffic processes.

The clusterization of the mobile network introduced in our reference model allowed to scale the complexity of the centralized controller. However, the augmented number of loads in the cluster may lead to big action spaces, unmanageable for the learning algorithms. An ability to generalize over the set of actions is necessary to handle scenarios with increased number of loads. Some recent literature investigated this problem in the case of large discrete action spaces [24] and continuous control [59], by considering approaches based on the actor-critic method [92] and deep RL. We believe these new methods may allow better scaling our proposed architecture, allowing dealing with more complex scenarios. However, the implementation of such methods into real-world control problems needs to be further investigated.

Bibliography

[1] 3GPP. *TS 36.942.; ; E-U; Radio equency (RF) system scenarios v14*". 2007 (cit. on p. 31).

[2] E. U. T. R. Access. „Further advancements for E-UTRA physical layer aspects, 3GPP TS 36.814“. In: *V9. 0.0, Mar* (2010) (cit. on p. 94).

[3] M. Agiwal, A. Roy, and N. Saxena. „Next Generation 5G Wireless Networks: A Comprehensive Survey“. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 1617–1655 (cit. on p. 2).

[4] A. S. Andrae and T. Edler. „On global electricity usage of communication technology: trends to 2030“. In: *MDPI Challenges* 6.1 (2015), pp. 117–157 (cit. on p. 1).

[5] J. J. G. Andrews, S. Buzzi, W. Choi, et al. „What will 5G be?“ In: *IEEE Journal on Selected Areas in Communications* 32.6 (2014), pp. 1065–1082 (cit. on p. 2).

[6] G Auer, O Blume, V Giannini, et al. *EARTH Deliverable D2.3: Energy efficiency analysis of the reference systems, areas of improvements and target breakdown*. Project Deliverable D2.3, `www.ict-earth.eu`. 2013 (cit. on pp. 3, 16, 30–33).

[7] G. Auer, V. Giannini, C. Desset, et al. „How much energy is needed to run a wireless network?“ In: *IEEE Wireless Communications* 18.5 (2011), pp. 40–49 (cit. on pp. 31, 64).

[8] Y. Bengio, A. Courville, and P. Vincent. „Representation Learning: A Review and New Perspectives“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828 (cit. on p. 47).

[9] D. P. Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Athena scientific Belmont, MA, 1995 (cit. on pp. 37, 40).

[10] L. Bonati, A. F. Gambin, and M. Rossi. „Wireless power transfer under the spotlight: Charging terminals amid dense cellular networks“. In: *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. Macau, China, 2017, pp. 1–9 (cit. on p. 18).

[11] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis. „Game-Theoretic Infrastructure Sharing in Multioperator Cellular Networks“. In: *IEEE Transactions on Vehicular Technology* 65.5 (2016), pp. 3326–3341 (cit. on p. 14).

[12] A. Bousia, A. Antonopoulos, L. Alonso, and C. Verikoukis. „Green distance-aware base station sleeping algorithm in LTE-Advanced“. In: *IEEE International Conference on Communications (ICC)*. 2012, pp. 1347–1351 (cit. on p. 13).

[13] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis. „Multiobjective Auction-Based Switching-Off Scheme in Heterogeneous Networks: To Bid or Not to Bid?“ In: *IEEE Transactions on Vehicular Technology* 65.11 (2016), pp. 9168–9180 (cit. on p. 14).

[14] A. Bousia, E. Kartsakli, L. Alonso, and C. Verikoukis. „Energy efficient base station maximization switch off scheme for LTE-advanced“. In: *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. Barcelona, Spain, 2012, pp. 256–260 (cit. on p. 12).

[15] S. Cai, L. Xiao, H. Yang, J. Wang, and S. Zhou. „A cross-layer optimization of the joint macro and picocell deployment with sleep mode for green communications“. In: *IEEE Wireless and Optical Communication Conference (WOCC)*. Chongqing, China, 2013, pp. 225–230 (cit. on p. 12).

[16] Y. K. Chia, S. Sun, and R. Zhang. „Energy Cooperation in Cellular Networks with Renewable Powered Base Stations“. In: *IEEE Transactions on Wireless Communications* 13.12 (2014), pp. 6996–7010 (cit. on pp. 16, 17).

[17] Y.-L. Chung. „An energy-saving small-cell zooming scheme for two-tier hybrid cellular networks“. In: *International Conference on Information Networking (ICOIN)*. Siem Reap, Cambodia, 2015, pp. 148–152 (cit. on p. 14).

[18] G. Cili, H. Yanikomeroglu, and F. R. Yu. „Cell switch off technique combined with coordinated multi-point (CoMP) transmission for energy efficiency in beyond-LTE cellular networks“. In: *IEEE International Conference on Communications (ICC)*. Ottawa, Canada, 2012, pp. 5931–5935 (cit. on p. 15).

[19] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021*. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ visual-networking-index-vni/mobile-white-paper-c11-520862.html. 2017 (accessed May, 2017) (cit. on pp. 1, 2).

[20] K. Davaslioglu and E. Ayanoglu. „Quantifying Potential Energy Efficiency Gain in Green Cellular Wireless Networks“. In: *IEEE Communications Surveys & Tutorials* 16.4 (2014), pp. 2065–2091 (cit. on p. 11).

[21] J. V. Dave, P. Halpern, and H. J. Myers. „Computation of Incident Solar Energy“. In: *IBM Journal of Research and Development* 19.6 (1975), pp. 539–549 (cit. on p. 25).

[22] M. Deruyck, D. De Vulder, W. Joseph, and L. Martens. „Modelling the power consumption in femtocell networks“. In: *Wireless Communications and Networking Conference Workshops (WCNCW)*. Paris, France, 2012, pp. 30–35 (cit. on pp. 3, 31).

[23] H. S. Dhillon, Y. Li, P. Nuggehalli, Z. Pi, and J. G. Andrews. „Fundamentals of heterogeneous cellular networks with energy harvesting“. In: *IEEE Transactions on Wireless Communications* 13.5 (2014), pp. 2782–2797 (cit. on p. 19).

[24] G. Dulac-Arnold, R. Evans, H. van Hasselt, et al. „Deep reinforcement learning in large discrete action spaces“. In: *arXiv preprint arXiv:1512.07679* (2015) (cit. on p. 116).

[25] S.-E. Elayoubi, L. Saker, and T. Chahed. „Optimal control for base station sleep mode in energy efficient radio access networks“. In: *IEEE International Conference on Computer Communications (INFOCOM)*. Shanghai, China, 2011, pp. 106–110 (cit. on p. 16).

[26] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. „Hierarchical Clustering“. In: *Cluster Analysis*. Wiley-Blackwell, 2011. Chap. 4, pp. 71–110 (cit. on p. 58).

[27]A. Fehske, J. Malmodin, G. Biczók, and G. Fettweis. „The Global Footprint of Mobile Communications: The Ecological and Economic Perspective“. In: *IEEE Communications Magazine, issue on Green Communications* 49.8 (2011), pp. 55–62 (cit. on p. 2).

[28]U. Forum. „Mobile Traffic Forecasts 2010-2020“. In: *ONF White Paper* Report no. 44 (2011) (cit. on p. 30).

[29]M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller. „auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks“. In: *arXiv preprint arXiv:1712.04382* (2017) (cit. on p. 47).

[30]O. N. Fundation. „Software-Defined Networking: The New Norm for Networks“. In: *ONF White Paper* 2 (2012), pp. 2–6 (cit. on p. 2).

[31]E. Gelenbe and E. T. Ceran. „Energy Packet Networks With Energy Harvesting“. In: *IEEE Access* 4 (2016), pp. 1321–1331 (cit. on p. 18).

[32]H. Ghazzai, M. J. Farooq, A. Alsharoa, et al. „Green Networking in Cellular HetNets: A Unified Radio Resource Management Framework With Base Station ON/OFF Switching“. In: *IEEE Transactions on Vehicular Technology* 66.7 (2017), pp. 5879–5893 (cit. on p. 12).

[33]I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016 (cit. on p. 48).

[34]J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. „Internet of Things (IoT): A vision, architectural elements, and future directions“. In: *Future generation computer systems* 29.7 (2013), pp. 1645–1660 (cit. on p. 2).

[35]B. Gurakan, O. Ozel, J. Yang, and S. Ulukus. „Energy cooperation in energy harvesting communications“. In: *IEEE Transactions on Communications* 61.12 (2013), pp. 4884–4898 (cit. on p. 18).

[36]F. Han, S. Zhao, L. Zhang, and J. Wu. „Survey of Strategies for Switching Off Base Stations in Heterogeneous Networks for Greener 5G Systems“. In: *IEEE Access* 4 (2016), pp. 4959–4973 (cit. on pp. 4, 12).

[37]F. Han, Z. Safar, and K. R. Liu. „Energy-efficient base-station cooperative operation with guaranteed QoS“. In: *IEEE Transactions on Communications* 61.8 (2013), pp. 3505–3517 (cit. on p. 13).

[38]Z. Hasan, H. Boostanimehr, and V. K. Bhargava. „Green cellular networks: A survey, some research issues and challenges“. In: *IEEE Communications Surveys & Tutorials* 13.4 (2011), pp. 524–540 (cit. on p. 31).

[39]H. A. H. Hassan, L. Nuaymi, and A. Pelov. „Renewable energy in cellular networks: A survey“. In: *2013 IEEE Online Conference on Green Communications (OnlineGreenComm)*. 2013, pp. 1–7 (cit. on p. 3).

[40]M. Hata. „Empirical formula for propagation loss in land mobile radio services“. In: *IEEE Transactions on Vehicular Technology* 29.3 (1980), pp. 317–325 (cit. on pp. 69, 81, 94).

[41]S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994 (cit. on p. 90).

[42] A. He, D. Liu, Y. Chen, and T. Zhang. „Stochastic geometry analysis of energy efficiency in HetNets with combined CoMP and BS sleeping“. In: *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. Washington DC, USA, 2014, pp. 1798–1802 (cit. on pp. 13, 15).

[43] Z. Hu, Y. Wei, X. Wang, and M. Song. „Green relay station assisted cell zooming scheme for cellular networks“. In: *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. Changsha, China, 2016, pp. 2030–2035 (cit. on p. 15).

[44] S.-C. Hung, H. Hsu, S.-Y. Lien, and K.-C. Chen. „Architecture Harmonization Between Cloud Radio Access Networks and Fog Networks“. In: *IEEE Access* 3 (2015), pp. 3019–3034 (cit. on p. 2).

[45] S. Hur, T. Kim, D. J. Love, et al. „Millimeter wave beamforming for wireless backhaul and access in small cell networks“. In: *IEEE Transactions on Communications* 61.10 (2013), pp. 4391–4403 (cit. on p. 17).

[46] International Telecommunication Union. *ICT Facts & Figures 2016*. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf. 2016 (accessed May, 2017) (cit. on p. 1).

[47] K. A. H. Ismaiil, B. Assaf, M. Ghantous, and M. Nahas. „Reducing power consumption of cellular networks by using various cell types and cell zooming“. In: *International Conference on e-Technologies and Networks for Development (ICeND)*. Beirut, Lebanon, 2014, pp. 33–38 (cit. on p. 14).

[48] C. Jia and T. J. Lim. „Resource partitioning and user association with sleep-mode base stations in heterogeneous cellular networks“. In: *IEEE Transactions on Wireless Communications* 14.7 (2015), pp. 3780–3793 (cit. on p. 13).

[49] I. T. Jolliffe. *Principal components in regression analysis*. Springer, 2002, pp. 167–198 (cit. on p. 47).

[50] M. Kamel, W. Hamouda, and A. Youssef. „Ultra-Dense Networks: A Survey“. In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2522–2545 (cit. on p. 2).

[51] S. Khalid, T. Khalil, and S. Nasreen. „A survey of feature selection and feature extraction techniques in machine learning“. In: *2014 Science and Information Conference*. IEEE. 2014, pp. 372–378 (cit. on p. 47).

[52] T. Khatib, A. Mohamed, and K. Sopian. „A review of solar energy modeling techniques“. In: *Renewable and Sustainable Energy Reviews* 16.5 (2012), pp. 2864 –2869 (cit. on p. 25).

[53] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. eprint: 1412.6980 (cit. on p. 95).

[54] B. Kreiner and J. Reeves. *Packetized Power*. US Patent App. 14/010,674. 2014 (cit. on p. 18).

[55] L. B. Le. „QoS-aware BS switching and cell zooming design for OFDMA green cellular networks“. In: *IEEE Global Communications Conference (GLOBECOM)*. Anaheim, CA, USA, 2012, pp. 1544–1549 (cit. on p. 14).

[56]F. Leccese and Z. Leonowicz. „Intelligent wireless street lighting system“. In: *2012 11th International Conference on Environment and Electrical Engineering*. 2012, pp. 958–961 (cit. on p. 116).

[57]G. Lee, W. Saad, M. Bennis, A. Mehbodniya, and F. Adachi. „Online Ski Rental for ON/OFF Scheduling of Energy Harvesting Base Stations“. In: *IEEE Transactions on Wireless Communications* 16.5 (2017), pp. 2976–2990 (cit. on p. 20).

[58]Y. Li and M. Chen. „Software-Defined Network Function Virtualization: A Survey“. In: *IEEE Access* 3 (2015), pp. 2542–2553 (cit. on p. 2).

[59]T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al. „Continuous control with deep reinforcement learning“. In: *arXiv preprint arXiv:1509.02971* (2015) (cit. on p. 116).

[60]B. Lindemark and G. Oberg. „Solar power for radio base station (RBS) sites applications including system dimensioning, cell planning and operation“. In: *Twenty-Third International Telecommunications Energy Conference INTELEC 2001*. 2001, pp. 587–590 (cit. on p. 5).

[61]C. Liu, Y. Wan, L. Tian, Y. Zhou, and J. Shi. „Base Station Sleeping Control with Energy-Stability Tradeoff in Centralized Radio Access Networks“. In: *IEEE Global Communications Conference (GLOBECOM)*. San Diego, CA, USA, 2015, pp. 1–6 (cit. on p. 13).

[62]D. López-Pérez, M. Ding, H. Claussen, and A. H. Jafari. „Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments“. In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2078–2101 (cit. on pp. 2, 3).

[63]J. Lorincz, T. Garma, and G. Petrovic. „Measurements and modelling of base station power consumption under real traffic loads“. In: *Sensors* 12.4 (2012), pp. 4281–4310 (cit. on p. 31).

[64]R. Mahapatra, Y. Nijsure, G. Kaddoum, N. U. Hassan, and C. Yuen. „Energy Efficiency Tradeoff Mechanism Towards Wireless Green Communication: A Survey“. In: *IEEE Communications Surveys & Tutorials* 18.1 (2016), pp. 686–705 (cit. on p. 11).

[65]Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. „A Survey on Mobile Edge Computing: The Communication Perspective“. In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2322–2358 (cit. on p. 115).

[66]M. A. Marsan, G. Bucalo, A. D. Caro, M. Meo, and Y. Zhang. „Towards zero grid electricity networking: Powering BSs with renewable energy sources“. In: *IEEE International Conference on Communications Workshops (ICC)*. Budapest, Hungary, 2013, pp. 596–601 (cit. on p. 20).

[67]M. Mendil, A. D. Domenico, V. Heiries, R. Caire, and N. Hadj-said. „Fuzzy Q-Learning based energy management of small cells powered by the smart grid“. In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Valencia, Spain, 2016, pp. 1–6 (cit. on p. 65).

[68]M. Mezzavilla, M. Miozzo, M. Rossi, N. Baldo, and M. Zorzi. „A Lightweight and Accurate Link Abstraction Model for the Simulation of LTE Networks in Ns-3“. In: *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWiM '12. Paphos, Cyprus: ACM, 2012, pp. 55–60 (cit. on p. 65).

[69]M. P. Mills. „The cloud begins with coal: Big data, big networks, big infrastructure, and big power“. In: *Digital Power Group* (2013) (cit. on p. 1).

[70]M. Miozzo, N. Piovesan, and P. Dini. „Coordinated Load Control of Renewable Powered Small Base Stations through Layered Learning“. In: *IEEE Transactions on Green Communications and Networking* (2019), pp. 1–1 (cit. on p. 20).

[71]M. Miozzo, L. Giupponi, M. Rossi, and P. Dini. „Distributed Q-learning for energy harvesting Heterogeneous Networks“. In: *IEEE International Conference on Communication Workshop (ICCW)*. London, UK, 2015, pp. 2006–2011 (cit. on p. 20).

[72]M. Miozzo, D. Zordan, P. Dini, and M. Rossi. „SolarStat: Modeling photovoltaic sources through stochastic Markov processes“. In: *IEEE International Energy Conference (ENERGYCON)*. Cavtat, Croatia, 2014, pp. 688–695 (cit. on pp. 20, 55).

[73]V. Mnih, K. Kavukcuoglu, D. Silver, et al. „Playing atari with deep reinforcement learning“. In: *arXiv preprint arXiv:1312.5602* (2013) (cit. on pp. 52, 92).

[74]R. Müllner and A. Riener. „An energy efficient pedestrian aware Smart Street Lighting system“. In: *International Journal of Pervasive Computing and Communications* 7.2 (2011), pp. 147–161 (cit. on p. 116).

[75]Z. Niu, Y. Wu, J. Gong, and Z. Yang. „Cell zooming for cost-efficient green cellular networks“. In: *IEEE Communications Magazine* 48.11 (2010), pp. 74–79 (cit. on p. 14).

[76]M. Oikonomakou, A. Antonopoulos, L. Alonso, and C. Verikoukis. „Fairness in multi-operator energy sharing“. In: *IEEE International Conference on Communications (ICC)*. Paris, France, 2017, pp. 1–6 (cit. on p. 19).

[77]M. Oikonomakou, A. Antonopoulos, L. Alonso, and C. Verikoukis. „Evaluating Cost Allocation Imposed by Cooperative Switching Off in Multi-Operator Shared HetNets“. In: *IEEE Transactions on Vehicular Technology* (2017) (cit. on p. 14).

[78]T. Osa, J. Pajarinen, G. Neumann, et al. „An algorithmic perspective on imitation learning“. In: *Foundations and Trends® in Robotics* 7.1-2 (2018), pp. 1–179 (cit. on p. 89).

[79]M. Peng, S. Yan, K. Zhang, and C. Wang. „Fog computing based radio access networks: issues and challenges“. In: *IEEE Network* 30.4 (2016), pp. 46–53 (cit. on p. 2).

[80]M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang. „Recent Advances in Cloud Radio Access Networks: System Architectures, Key Techniques, and Open Issues“. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2282–2308 (cit. on p. 2).

[81]N. Piovesan and P. Dini. „Optimal direct load control of renewable powered small cells: A shortest path approach“. In: *Internet Technology Letters* (2017). e7, e7–n/a (cit. on p. 56).

[82]G. Piro, M. Miozzo, G. Forte, et al. „HetNets Powered by Renewable Energy Sources: Sustainable Next-Generation Cellular Networks“. In: *IEEE Internet Computing* 17.1 (2013), pp. 32–39 (cit. on p. 3).

[83]P. J. Rousseeuw. „Silhouettes: a graphical aid to the interpretation and validation of cluster analysis“. In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65 (cit. on p. 58).

[84]D. Sabella, D. Rapone, M. Fodrini, et al. „Energy Management in Mobile Networks Towards 5G“. In: *Energy Management in Wireless Cellular and Ad-hoc Networks*. Ed. by M. Z. Shakir, M. A. Imran, K. A. Qaraqe, M.-S. Alouini, and A. V. Vasilakos. Springer International Publishing, 2016, pp. 397–427 (cit. on pp. 3, 31, 32).

[85]S. Samarakoon, M. Bennis, W. Saad, and M. Latva-aho. „Dynamic Clustering and ON/OFF Strategies for Wireless Small Cell Networks“. In: *IEEE Transactions on Wireless Communications* 15.3 (2016), pp. 2164–2178 (cit. on p. 12).

[86]A. Sarveniazi. „An actual survey of dimensionality reduction“. In: *American Journal of Computational Mathematics* 4.02 (2014), p. 55 (cit. on p. 46).

[87]S. Sesia, M. Baker, and I. Toufik. *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011 (cit. on p. 64).

[88]S. Shin, B.-Y. Choi, and S. Song. „Siesta: Software-Defined Energy Efficient Base Station Control for Green Cellular Networks“. In: *International Conference on Computer Communication and Networks (ICCCN)*. Las Vegas, Nevada, USA, 2015, pp. 1–8 (cit. on p. 16).

[90]Y. S. Soh, T. Q. Quek, M. Kountouris, and H. Shin. „Energy Efficient Heterogeneous Cellular Networks“. In: *IEEE Journal on Selected Areas in Communications* 31.5 (2013), pp. 840–850 (cit. on p. 13).

[91]K. Son, H. Kim, Y. Yi, and B. Krishnamachari. „Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks“. In: *IEEE Journal on Selected Areas in Communications* 29.8 (2011), pp. 1525–1536 (cit. on p. 13).

[92]R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on pp. 50, 51, 87, 116).

[93]H. Tabassum, U. Siddique, E. Hossain, and M. J. Hossain. „Downlink performance of cellular systems with base station sleeping, user association, and scheduling“. In: *IEEE Transactions on Wireless Communications* 13.10 (2014), pp. 5752–5767 (cit. on p. 13).

[94]R. Tao, J. Zhang, and X. Chu. „An Energy Saving Small Cell Sleeping Mechanism with Cell Expansion in Heterogeneous Networks“. In: *IEEE Vehicular Technology Conference (VTC Spring)*. Porto, Portugal, 2016, pp. 1–5 (cit. on p. 12).

[95]H. D. Trinh, N. Bui, J. Widmer, L. Giupponi, and P. Dini. „Analysis and modeling of mobile traffic using real traces“. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2017, pp. 1–6 (cit. on p. 30).

[96]C. R. Valenta and G. D. Durgin. „Harvesting wireless power: Survey of energy-harvester conversion efficiency in far-field, wireless power transfer systems“. In: *IEEE Microwave Magazine* 15.4 (2014), pp. 108–120 (cit. on p. 18).

[97]M. Vincenzi, A. Antonopoulos, E. Kartsakli, et al. „Cooperation incentives for multi-operator C-RAN energy efficient sharing“. In: *IEEE International Conference on Communications (ICC)*. Paris, France, 2017, pp. 1–6 (cit. on p. 14).

[98]Y. Wang, Y. Zhang, Y. Chen, and R. Wei. „Energy-efficient design of two-tier femtocell networks“. In: *EURASIP Journal on Wireless Communications and Networking* 2015.1 (2015), p. 1 (cit. on p. 13).

[99]C. J. Watkins and P. Dayan. „Q-learning“. In: *Machine learning* 8.3-4 (1992), pp. 279–292 (cit. on p. 51).

[100]C. J. C. H. Watkins. „Learning from delayed rewards“. In: (1989) (cit. on p. 51).

[101]J. Wu, Y. Zhang, M. Zukerman, and E. K.-N. Yung. „Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey“. In: *IEEE Communications Surveys & Tutorials* 17.2 (2015), pp. 803–826 (cit. on p. 16).

[102]F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin. „Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment“. In: *IEEE/ACM Trans. Netw.* 25.2 (Apr. 2017), pp. 1147–1161 (cit. on p. 30).

[103]J. Xu and R. Zhang. „CoMP Meets Smart Grid: A New Communication and Energy Cooperation Paradigm“. In: *IEEE Transactions on Vehicular Technology* 64.6 (2015), pp. 2476–2488 (cit. on p. 19).

[104]J. Xu, L. Duan, and R. Zhang. „Cost-aware green cellular networks with energy and communication cooperation“. In: *IEEE Communications Magazine* 53.5 (2015), pp. 257–263 (cit. on p. 19).

[105]X. Xu, C. Yuan, W. Chen, X. Tao, and Y. Sun. „Adaptive Cell Zooming and Sleeping for Green Heterogeneous Ultra-Dense Networks“. In: *IEEE Transactions on Vehicular Technology* (2017) (cit. on p. 15).

[106]A. K. Yadav and S. Chandel. „Solar radiation prediction using Artificial Neural Network techniques: A review“. In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 772 –781 (cit. on p. 55).

[107]N. Yu, Y. Miao, L. Mu, et al. „Minimizing Energy Cost by Dynamic Switching ON/OFF Base Stations in Cellular Networks“. In: *IEEE Transactions on Wireless Communications* 15.11 (2016), pp. 7457–7469 (cit. on p. 12).

[108]Y. Yuan and P. Gong. „A QoE-orientated base station sleeping strategy for multi-services in cellular networks“. In: *International Conference on Wireless Communications & Signal Processing (WCSP)*. Nanjing, China, 2015, pp. 1–5 (cit. on p. 12).

[109]H. Zhang, J. Cai, and X. Li. „Energy-efficient base station control with dynamic clustering in cellular network“. In: *IEEE International Conference on Communications and Networking (CHINACOM)*. Guilin, China, 2013, pp. 384–388 (cit. on p. 12).

[110]G. Zheng, Z. Ho, E. A. Jorswieck, and B. Ottersten. „Information and energy cooperation in cognitive radio networks“. In: *IEEE Transactions on Signal Processing* 62.9 (2014), pp. 2290–2303 (cit. on p. 18).

[111]Y. Zhu, Z. Zeng, T. Zhang, and D. Liu. „A QoS-Aware Adaptive Access Point Sleeping in Relay Cellular Networks for Energy Efficiency“. In: *IEEE Vehicular Technology Conference (VTC Spring)*. Seoul, Korea, 2014, pp. 1–5 (cit. on p. 12).

[112]Y. Zhu, Z. Zeng, T. Zhang, L. An, and L. Xiao. „An energy efficient user association scheme based on cell sleeping in LTE heterogeneous networks“. In: *International Symposium on Wireless Personal Multimedia Communications (WPMC)*. Sydney, Australia, 2014, pp. 75–79 (cit. on p. 13).

[113]Y. Zhu, T. Kang, T. Zhang, and Z. Zeng. „QoS-aware user association based on cell zooming for energy efficiency in cellular networks“. In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*. London, UK, 2013, pp. 6–10 (cit. on p. 15).

[114]D. Zordan, M. Miozzo, P. Dini, and M. Rossi. „When telecommunications networks meet energy grids: Cellular networks with energy harvesting and trading capabilities“. In: *IEEE Communications Magazine* 53.6 (2015), pp. 117–123 (cit. on pp. 3, 19).

[115]M. Šúri, T. A. Huld, E. D. Dunlop, and H. A. Ossenbrink. „Potential of solar electricity generation in the European Union member states and candidate countries“. In: *Solar Energy* 81.10 (2007), pp. 1295 –1305 (cit. on pp. 6, 55).

Websites

[89]SoDa. *Solar Radiation Data*. URL: `http://www.soda-pro.com/web-services/radiation/helioclim-3-archives-for-free` (visited on May 31, 2018) (cit. on p. 56).

List of Figures

List of Tables