UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Contributions to energy-aware demand-response systems using SDN and NFV for fog computing

Christian José Tipantuña Tenelema

Doctoral Thesis

Contributions to Energy-Aware Demand-Response Systems Using SDN and NFV for Fog Computing

by:

Christian José Tipantuña Tenelema

Ph.D. Advisors:

Dr. Xavier Hesselbach

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Telematics

BAMPLA

(Design and Evaluation of Broadband Networks and Services research group)

Departament of Network Engineering

December 20, 2021

“Nothing is invented, for it is written in nature first.”

Antonio Gaudí

Abstract

Ever-increasing energy consumption, the depletion of non-renewable resources, the climate impact associated with energy generation, and finite energy-production capacity are important concerns worldwide that drive the urgent creation of new energy management and consumption schemes. In this regard, by leveraging the massive connectivity provided by emerging communications such as the 5G systems, this thesis proposes a long-term sustainable Demand-Response solution for the adaptive and efficient management of available energy consumption for Internet of Things (IoT) infrastructures, in which energy utilization is optimized based on the available supply. In the proposed approach, energy management focuses on consumer devices (e.g., appliances such as a light bulb or a screen). In this regard, by proposing that each consumer device be part of an IoT infrastructure, it is feasible to control its respective consumption.

The proposal includes an architecture that uses Network Functions Virtualization (NFV) and Software Defined Networking technologies as enablers to promote the primary use of energy from renewable sources. Associated with architecture, this thesis presents a novel consumption model conditioned on availability in which consumers are part of the management process. To efficiently use the energy from renewable and non-renewable sources, several management strategies are herein proposed, such as the prioritization of the energy supply, workload scheduling using time-shifting capabilities, and quality degradation to decrease the power demanded by consumers if needed. The adaptive energy management solution is modeled as an Integer Linear Programming, and its complexity has been identified to be NP-Hard. To verify the improvements in energy utilization, an optimal algorithmic solution based on a brute force search has been implemented and evaluated.

Because the hardness of the adaptive energy management problem and the non-polynomial growth of its optimal solution, which is limited to energy management for a small number of energy demands (e.g., 10 energy demands) and small values of management mechanisms, several faster suboptimal algorithmic strategies have been proposed and implemented. In this context, at the first stage, we implemented three heuristic strategies: a greedy strategy (GreedyTs), a genetic-algorithm-based solution (GATs), and a dynamic programming approach (DPTs). Then, we incorporated into both the optimal and heuristic strategies a prepartitioning method in which the total set of analyzed services is divided into subsets of smaller size and complexity that are solved iteratively. As a result of the adaptive energy management in this thesis, we present eight strategies,

one optimal and seven heuristic, that when deployed in communications infrastructures such as the NFV domain, seek the best possible scheduling of demands, which lead to efficient energy utilization. The performance of the algorithmic strategies has been validated through extensive simulations in several scenarios, demonstrating improvements in energy consumption and the processing of energy demands. Additionally, the simulation results revealed that the heuristic approaches produce high-quality solutions close to the optimal while executing among two and seven orders of magnitude faster and with applicability to scenarios with thousands and hundreds of thousands of energy demands.

This thesis also explores possible application scenarios of both the proposed architecture for adaptive energy management and algorithmic strategies. In this regard, we present some examples, including adaptive energy management in-home systems and 5G networks slicing, energy-aware management solutions for unmanned aerial vehicles, also known as drones, and applicability for the efficient allocation of spectrum in flex-grid optical networks. Finally, this thesis presents open research problems and discusses other application scenarios and future work.

Acknowledgments

The research work shown in this thesis could not have been conducted without the support of the Escuela Politécnica Nacional (EPN) and the Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) of the República del Ecuador.

During this journey of learning and research, I would like to thank a series of people who have allowed me to fulfill the objectives set at this stage of my life. First of all, I express my unconditional love and deepest gratitude to my mommy Marina Tenelema, who was ahead of us on the road to eternity and who will always live in my mind and heart. My sincere thanks to my family, to my dad Jose, to my siblings Vivi and Saúl, and to my niece Dannita, who have given me their unconditional support, encouragement, and motivation during these years of study. A special thanks to Carlita for her affection, friendship, company, and motivation, which has been of great help to me, especially in adversity.

Regarding the academic field, I want to express my gratitude to my supervisor, Prof. Dr. Xavier Hesselbach, for the continuous support, guidance, commitment, and valuable feedback I received during these years. Thanks to Prof. Hesselbach, I have been involved in different, interesting, and challenging research tasks, projects, and collaborations that have broadened my scientific scope. I also thank the colleagues from University Carlos III of Madrid, mainly Prof. Dr. Francisco Valera and Victor Sanchez-Aguero, with whom I was able to work and provide an applicable approach to my research in the fascinating area of communication systems enabled by unmanned aerial vehicles. Finally, in my quest to extend energy management strategies to larger-scale scenarios, the information received by Prof. Dr. Walter Unger from Aachen University was of great help to me, for which I also want to express my gratitude.

In addition, I would like to thank my friends in Ecuador and abroad, who made my stay abroad more bearable and enriching. I also want to express my gratitude to my laboratory colleagues, with whom I was able to exchange knowledge, ideas, and opinions, and share various cultural and sports activities.

Last but not least, I want to express my eternal gratitude to my Creator for all that he has given me and allowed me to live since, without him, nothing would make sense, and nothing would be possible.

Dedicated to my beloved family, mainly to my mommy Marina Elisa, and to all those of humble origins who struggle against adversity to make their dream come true.

Contents

List of Figures

List of Tables

Acronyms

Acronym	Definition
3GPP	Third Generation Partnership Project
5G	Fifth-generation cellular network
AP	Access Point
BETA	Betweenness centrality heuristic algorithm
BS	Base Station
BAN	Building Area Network
CAPEX	Capital Expenditures
CPU	Central Processing Unit
CS	Critical Services
DC	Data Center
DR	Demand Response
EC	Energy Consumer
EM	Energy Manager
eMBB	Enhanced Mobile Broadband
EPC	Evolved packet core
ES	Energy Supplier
ETSI	European Telecommunications Standards Institute
FANET	Flying Ad Hoc Network
FC	Fog Computing
GCS	Ground Control Station
GEM	Green Energy Manager
GPS	Global Positioning System
HAN	Home Area Network
HEMS	Home Energy Management System
IAN	Industrial Area Network
ICT	Information and Communications Technologies
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IoE	Internet of Energy

IoT	Internet of Things
IRTF	Internet Research Task Force
ISP	Internet Service Provider
ITU	International Telecommunication Union
KPI	Key Performance Indicator
LTE	Long Term Evolution
mIoT	Massive IoT
MU-MIMO	Multi-user MIMO
NCS	Non-Critical Services
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Infrastructure
NFVO	NFV Orchestrator
NSI	Network Slice Instance
MANET	Mobile Ad-hoc Network
MANO	Management and Orchestration
MEC	Multi-access Edge Computing
MMKP	Multi-dimensional Multi-choice knapsack Problem
MNO	Mobile Network Operator
NOMA	Non-Orthogonal Multiple Access
NB-IoT	Narrow Band Internet of Things
NP	Nondeterministic polynomial
ONF	Open Networking Foundation
OPEX	Operating Expenses
OFDMA	Orthogonal Frequency Division Multiplexing Access
PLC	Power Line Communications
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RAN	Radio Access Network
SLA	Service Level Agreement
SFC	Service Function Chain
SC	Service Chain
SCADA	Supervisory Control And Data Acquisition
SCMA	Sparse Code Multiple Access
SDN	Software Defined Network
SUAV	Small Unmanned Aerial Vehicle
SST	Slice/Service Type
UAV	Unmanned Aerial Vehicle

URLLC	Ultra Reliable Low Latency Communications
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
V2X	Vehicle to X (e.g., Vehicle, Infrastructure)
VNF	Virtual Network Function
VNFI	VNF Infrastructure
VNFM	VNF Manager
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

Notations

	Notation	Description
Energy supply	m	Time interval in which energy exists
	P_A	Total available power
	P_{ES}	Power provided by the energy supplier
	P_{NR}	Power from non-renewable energy sources
	P_R	Power from renewable energy sources
	$T_{init}^{P_{ES}}$	Starting time of P_{ES}
	w_R	Weight associated to renewable energy
Consumption and services	i	Service identifier, $i \in \{1, \ldots, N\}$
	j	Priority identifier, $j \in \{1, \cdots, L\}$
	L	Number of priority levels
	N	Total number of services
	t_i	Starting time of service S_i
	d_i	Duration of service S_i
	p_i	Power demanded of service S_i
	q_i	Priority level of service S_i
	u_i	Time shifting value of service S_i
	P_D	Aggregated power demanded
	W	Maximum analysis time horizon
Metrics	AR	Acceptance ratio
	E_{A_U}	Available energy utilization
	P_{LACK}	Missing power
	P_{RES}	Residual power
	$\sigma_{P_{RES}}$	Standard deviation of P_{RES}
	σ_{Ts}	Standard deviation of T_s^k
Strategies	$BETA$	Betweenness centrality heuristic algorithm in UAV-enabled systems
	$disOptTs$	Maximum distance to optimum
	$FastFs$	Heuristic strategy in flex-grid optical networks
	$GATs$	Genetic algorithm based strategy

$GreedyTs$	Greedy strategy
G_T	Relative gain in time
$DPTs$	Dynamic programming Based Strategy
$OptTsCost$	Optimal strategy based on a cost funciton
$OptTsCost$	Optimal strategy in the context of 5G network slicing
$OptFs$	Optimal strategy in flex-grid optical networks
$PHRASE$	Prepartitioning strategy in HEMS
ρ	Approximation ratio

Chapter 1

Introduction

This chapter presents a brief introduction of this Ph.D. thesis, pointing out the motivation, objectives, and main scientific contributions of our research work. Moreover, this chapter presents the outline of this dissertation.

1.1 Introduction and Motivation

Traditionally, worldwide energy provisioning has been dominated by fossil fuels such as petroleum, coal, and natural gas, resulting in an increase in CO_2 emissions and global warming [1]. In the near future, this dependence could potentially lead to an energy crisis due to the risk of depleting fossil fuels, the infeasibility of meeting the ever-growing energy demand, the increased cost of energy production, and the high impact on climate and the environment. To ensure human society's development, a zero-carbon alternative is the use of green energy from renewable energy sources, like solar or wind, into the world energy matrix, which can meet current and future energy demands [2]. It is conjectured that more than 50% of projected global energy needs can be satisfied by utilizing the Earth's green energy [3]. Then, the evolution toward ecosystems completely powered by renewable energy is seen as a very promising approach to tackle sustainability issues and reduce the carbon footprint. In this context, major IT providers, including Google [4], Microsoft[5], and Apple [6], as well as mobile network operators [7], are already promoting all computing and networking infrastructure being fully supported by renewable energy. Despite the multiple benefits that green sources can offer, their intermittent behavior (which may cause instability when they are integrated into conventional energy sources [2]), added to the inefficient use of the generated energy (also from non-renewable sources), requires the development of consumption management solutions to maintain power reliability, continuity, and quality. In this regard, consumer-side participation and Demand-Response (DR) schemes are effective solutions to adapt the energy consumption patterns to energy supply dynamically. Adaptive DR schemes or programs based on agreements between the Energy Supplier (ES) and the Energy Consumers (ECs) promote the exchange of indications and requests between parties to adapt consumption in response to changes in energy generation with the motivation of energy bill reduction and/or free usage periods for end users [8].

Technically, an adaptive, environmentally friendly DR energy management system needs robust and scalable Information and Communications Technologies (ICT) both to facilitate the interaction between the ES and the ECs and to deploy the management strategies that lead to efficient energy utilization [9]. In this regard, modern energy systems (e.g., smart grids) can incorporate diverse ICT and Internet of Things (IoT) technologies to improve control, management, and monitoring tasks and to extend energy management to the end user. Currently, there is an important proliferation of Internet connectivity worldwide, and some studies estimate that by 2022 the number of IoT devices (energy consumers) will surpass 28.5 billion (i.e., 10 billion more devices than in 2017) [10]. This fact reveals both the feasibility of implementing efficient energy management solutions with consumer-side participation and the application scope (e.g., energy management in homes, in buildings, in starships, or even for entire countries). Fig. 1.1[1] summarizes the main energy concerns and alternatives or solutions proposed for achieving adaptive energy management in the context of this Ph.D. thesis.

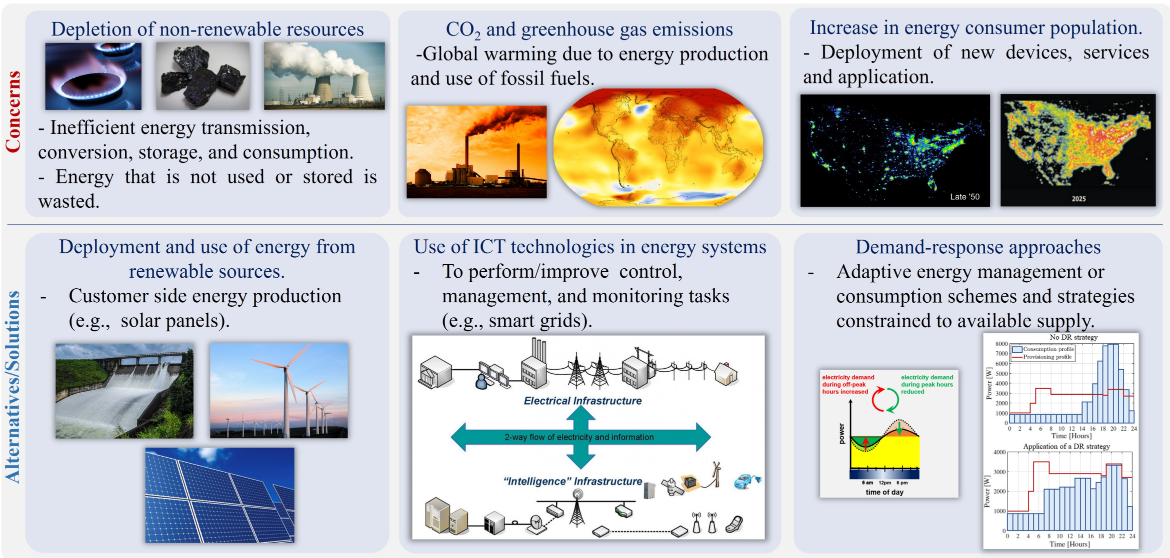


FIGURE 1.1: Summary of concerns and solutions related to the development of this Ph.D. thesis.

A critical feature of adaptive energy consumption systems is the computational capacity (mainly in terms of memory and processing power) needed to execute the management strategies (carried out through algorithmic solutions), which promote the interaction between generation and consumption sides and optimize the use of available energy either from renewable or non-renewable sources. Traditionally, the ES has used operations centers and, recently, Data Centers (DCs) infrastructures in which a variety of management

[1]The author has created all the figures in this thesis. In needed, additional images, as shown in Fig. 1.1, are under the creative commons licenses.

policies have been implemented [11]. However, the dynamic behavior of the generation and consumption ecosystem has prompted these DCs to evolve to cloud computing infrastructures in which sophisticated technologies such as Network Function Virtualization (NFV) can be deployed [9]. A technological solution such as NFV can provide to energy management systems programmability, by facilitating the deployment of different management strategies through virtual components, and the Management and Orchestration (MANO) entities to coordinate all actions between the ES and the ECs and the underlying network (e.g., a Software Defined Networking (SDN) solution). Thus, an ICT-based approach (e.g., based on NFV technology) can manage computational resources on-demand, allowing adaptive energy management to be scalable and carried out in reduced execution times, especially for delay-sensitive applications. Likewise, the massive connectivity, with low latency and high bandwidth, available to end-users due to the proliferation of IoT technologies and development of network systems, such as 5G, makes it possible for potential energy consumers (i.e., devices, appliances, or, in general, IoT infrastructures) to participate in the energy management process [12].

Although an ICT-enabled energy management solution (e.g., based on an NFV approach), deployed on cloud computing infrastructures, can offer programmability, scalability, and high computational capacity, the inherent complexity of the algorithmic management solutions to enable adaptive consumption is still an important, open issue. Existing literature shows that optimal or exact algorithmic solutions for energy management considering features such as the dynamic nature of renewable energy generation [13], added to management mechanisms such as prioritization in energy provisioning if needed, time-shifting capabilities, and quality degradation (i.e., a decrease of power consumption) to adapt the workloads to energy availability, present a complexity NP-Hard [9]. This complexity level imposes limits on the applicability of consumption management to small-scale scenarios (e.g., for units of energy demands) and small values or ranges of management strategies (e.g., using units of time-shifting slots) [14]. Therefore, faster and less complex adaptive energy management approaches (also known as approximate/suboptimal solutions or heuristic) are essential. A variety of techniques or methods, such as greedy strategies and genetic algorithm-based solutions, can be implemented to meet these requirements. In addition, both optimal and heuristic algorithmic strategies need to be evaluated considering a variety of consumption and generation profiles in small and large-scale scenarios (e.g., with thousands of energy demands) to verify their effectiveness in energy utilization and consumption. Apart from demonstrating improvements in energy management of available supply, the suboptimal or heuristic strategies must deliver high-quality solutions (compared to the optimal), feasible running times (e.g., several order of magnitude faster than optimal solutions), and a suitable computational resource usage, that allow their potential deployment not only on cloud computing infrastructures but also on edge or fog computing infrastructures, and even on Home Energy Management Systems (HEMS).

Considering that an ICT-enabled and environmentally friendly energy management solution is a feasible alternative that can offer sustainability for human development, increase in overall energy utilization, and prevention of energy outages while producing cost reductions for the ES and ECs, developing an architectural framework and the algorithmic strategies (both optimal and heuristic) that enable adaptive energy consumption is of the utmost importance. This is the main objective of this Ph.D. thesis. Specifically, this thesis proposes an architecture that enabled by modern ICT solutions (such as NFV, SDN, and IoT) and carry out through algorithmic strategies (optimal and heuristic) enables optimized use of available supply obtained mainly obtained from renewable energy sources and with applicability to small and large-scale IoT infrastructures. In the proposed approach, energy management focuses on consumer devices (e.g., appliances such as a light bulb or a screen). In this regard, by proposing that each consumer device be part of an IoT infrastructure, it is feasible to control its respective consumption. The proposed management solution represents a feasible and promising move to future energy systems, currently referred to as the Internet of Energy (IoE) [15].

1.2 Objectives

The main objective of this Ph.D. thesis is to contribute to the study of the energy efficiency problem by proposing a Demand-Response ecosystem, that based on modern technologies such as SDN and NFV, provides an adaptive energy consumption with applicability to small and large-scale IoT scenarios. In this approach, energy management focuses on consumer devices (e.g., appliances such as a light bulb or a screen). In this regard, by proposing that each consumer device be part of an IoT infrastructure, it is feasible to control its respective consumption. To this end, we have identified a set of objectives that must be achieved and listed below.

- Identify the problem of lack of synchronization of energy generation and consumption (which causes scarcity or waste of energy) and explore the benefits and opportunities of implementing DR systems for adaptive consumption.
- Review research work about ICT-based solutions and customer-side participation for energy management, including the approaches that consider the participation of renewable energy sources. This, as a starting point, to detect possible technologies, mechanisms, schemas, and procedures to be used in the proposed adaptive energy management solution.
- Identify the management mechanisms to be applied on energy demands (e.g., time-shifting capabilities or quality degradation) to adapt the consumption patterns to availability (whether renewable or not).
- Analyze the implications of applying the selected management mechanisms in processing energy demands (e.g., delayed or anticipated execution of demands or rejection). Moreover, analyze the complexity that these management mechanisms can contribute to the overall energy management proposal.

  proposal in order to present a robust architectural framework for the efficient management of energy consumption.
- Analyze the ICT requirements for the deployment of the adaptive energy management solution and appropriate architectural framework. Considering the computational capacity needed for the deployment of management strategies (which are implemented through algorithmic approaches) and the infrastructure to coordinate the notifications, requests, and actions between the ES and the ECs.

- Propose an architecture based on advanced ICT technologies (e.g., NFV and SDN) that enables adapting the consumption patterns to available supply either from renewable and now renewable sources. The architecture proposal must include describing the stakeholders involved, their features, and the interaction level to continuously adapt consumption to generation.
- Characterize the ES and the ECs mathematically, taking into account features such as the type of energy sources (i.e., renewable and non-renewable) and the use of management mechanisms (e.g., time-shifting capabilities), respectively.
- Formulate the energy model mathematically for adaptive consumption (e.g., through an Integer Linear Programming (ILP) formulation), defining the objective function to be optimized and the constraints related, and considering finite energy supply and the application of management mechanisms on energy demands.
- Analyze the complexity level of the adaptive energy management model to determine its classification (e.g., a knapsack-like problem) and the possible strategies or methods to solve it.
- Define performance metrics to determine the utilization of energy capacity, improvements of energy consumption, and increase of processing if energy demands, which in normal conditions (i.e., without the application of management mechanisms) would be rejected.
- Design an algorithmic strategy for solving the energy model for adaptive consumption optimally (e.g., a solution base on a brute-force method). This implementation allows to determine all concerns regarding the adaptive energy management model and can be used as a baseline to compare the performance of suboptimal approaches.
- Analyze the growth rate of the optimal solution to determine its application scope, its limitations, and the possible strategies to tackle the complexity issues, mainly for large-scale scenarios (e.g., for thousands or services) and/or for large values of management mechanisms (e.g., energy management using a 10-time slot for time-shifting).
- Analyze some heuristic strategies in the scope of the problem of adaptive energy management.
- Design faster and heuristic strategies that solve the adaptive energy management model in reduced running times (e.g., several order of magnitude faster), with reduced computational capacity usage (mainly defined in terms of RAM and CPU), with applicability to thousand or hundred of thousands of energy demands, and producing high-quality solutions compared to the optimal values.
- Evaluate the proposed algorithmic strategies, both optimal and heuristic, through extensive simulations in diverse scenarios and for different generation and consumption conditions. The results must be analyzed considering as baseline the no application of management mechanisms. Moreover, the heuristic strategies can be analyzed in offline and online approaches, and their solutions can be compared with the exact results and with solutions of existing related approaches in the literature to validate their effectiveness and benefits in adaptive energy management.

- Identify potential application scenarios in which the proposed architecture or the algorithmic strategies can be used to efficiently manage resources not only for energy but also for other parameters such as the available spectrum.

1.3 Summary of Contributions

The main contributions of this thesis following the proposed objectives are listed below:

- An architecture proposal that, based on modern communications technologies such as 5G, NFV, and SDN, can perform efficient and adaptive management of available energy, whether 100% renewable or not, for IoT infrastructures.
- A novel consumption model subject to availability in which the IoT-enabled consumers (i.e., devices or appliances) are part of the energy management process.
- Several management strategies for the efficient consumption of energy produced by a combination of renewable and non-renewable sources. These include prioritizing energy supply, time-shifting capabilities, quality of degradation, and rejection of energy demands, to adapt the consumption pattern to the available supply.
- The mathematical characterization of ES and ECs considering the nature of energy provisioning (renewable or not) and the management mechanisms that can affect energy demands.
- The mathematical model of adaptive energy consumption based on an ILP formulation and considering finite energy provisioning and the management mechanisms on energy demands.
- The computational complexity estimation of adaptive energy management model.
- An optimal brute-force-search-based algorithmic strategy defined as OptTsCost to solve the ILP optimally.
- Definition of performance metrics such as residual power, energy utilization, acceptance ratio, and missing power to verify the improvements in energy utilization and consumption achieved with the proposed adaptive energy model and algorithmic strategies implemented.
- Discussion of possible algorithms to tackle the hardness of the ILP and the computational complexity of OptTs.
- Three heuristic strategies identified as GreedyTs, GATs, and DPTs to solve the adaptive energy management problem in reduced running time compared to the optimal solution, OptTsCost, and with applicability to IoT scenarios with thousands or hundreds of thousands of energy demands.
- Comparison of the solutions delivered by heuristic strategies with the exact results produced by the optimal algorithmic strategy OptTs.

- The application of a pre-partitioning method inspired in divide-and-conquer approach to scale up the operation of both optimal and heuristic strategies to IoT-enabled scenarios with thousands and hundreds of thousands of energy demands.

- The evaluation of optimal and heuristic strategies in terms of performance metrics, through extensive simulations in different scenarios to confirm the improvements in the management of energy consumption and demand processing.

- The evaluation of optimal and heuristic strategies according to the usage of computational capacity, in terms of Random Access Memory (RAM) and Central Processing Unit (CPU).

- Discussion of open research challenges and possible application scenarios in the context of adaptive energy management.

- Application scenarios in which both the proposed architecture and the algorithmic strategies developed are used to efficiently manage available resources, mainly concerning energy consumption. These application scenarios include: (i) adaptive energy management in HEMS, in which particularly the proposed heuristic strategy denoted as PHRASE is evaluated in offline and online approaches, and their results are compared with the optimal solution and with existing approaches; (ii) adaptive energy management applied to 5G network slicing, in which the architecture and energy model is adapted to the requirements and characteristics of modern mobile communications systems; (iii) energy-aware management in the field of Unmanned Aerial Vehicles (UAVs), in which an optimal solution and a proposed heuristic denoted as BETA are developed to optimize service availability and efficient use of UAVs; and (ii) optimized spectrum utilization in flex-grid optical networks, in which the proposed architecture and prepartitioning method is adapted to improve the allocation of available spectrum.

1.4 Related Publications

In this section, we list the result of the publication of the research done during this thesis. Fig. 1.2 summarizes the development of this Ph.D. thesis according to contributions made and the published and submitted content. As shown in Fig. 1.2, some contributions include a repository with the code used for simulations to facilitate reproducibility of the algorithmic strategies and corresponding results. Most of the contents of this dissertation have been published in the following journals and conferences:

Journal Publications:

J1 **Christian Tipantuña**, Xavier Hesselbach, Victor Sánchez-Aguero, Francisco Valera, Iván Vidal, and Borja Nogales. An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles. *Wireless Communications and Mobile Computing*, 2019. ISSN:1530-8677. Impact factor 2019: 2.336, Q3 [16].

J2 **Christian Tipantuña** and Xavier Hesselbach. NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy. *IEEE Open Journal of the Communications Society*, 1:357–380, 2020. ISSN:2644-125X (new journal since 2020, no impact factor available yet) [9].

J3 Victor Sánchez-Aguero, Francisco Valera, Ivan Vidal, **Christian Tipantuña**, and Xavier Hesselbach. Energy-aware management in multi-uav deployments: Modelling and strategies. *Sensors*, 20(10):2791, 2020. ISSN:1424-8220. Impact factor 2020: 3.576, Q1 [17].

J4 **Christian Tipantuña** and Xavier Hesselbach. Adaptive energy management in 5G network slicing: Requirements, architecture, and strategies. *Energies*, 13(15):3984, 2020. ISSN:1996-1073. Impact factor 2020: 3.004, Q3 [18].

J5 **Christian Tipantuña** and Xavier Hesselbach. NFV-enabled efficient renewable and non-renewable energy management: Requirements and algorithms. *Future Internet*, 12(10):171, 2020. ISSN:1999-5903. CiteScore 2020: 4.1, Q3 [19].

J6 **Christian Tipantuña** and Xavier Hesselbach. IoT-enabled proposal for adaptive self-powered renewable energy management in home systems. *IEEE Access*, 9:64808–64827, 2021. doi: 10.1109/ACCESS.2021.3073638. ISSN:2169-3536. Impact factor 2020: 3.671, Q1 [20].

J7 **Christian Tipantuña**, Xavier Hesselbach, and Walter Unger. Heuristic Strategies for NFV-Enabled Renewable and Non-renewable Energy Management in the Future IoT World. doi: 10.1109/ACCESS.2021.3110246. Impact factor 2020: 3.671, Q1 [21].

J8 **Christian Tipantuña** and Xavier Hesselbach. Network Technologies for Future Green Energy Management: A Comprehensive Survey. To be submitted to *Elsevier Renewable and Sustainable Energy Transition. In preparation.*

Conference Publications:

C1 **Christian Tipantuña** and Paúl Yanchapaxi. Network Functions Virtualization: An Overview and Open-Source Projects. In2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), pages 1–6, Salinas, Ecuador, 2017. ISBN 9781538638941. doi: 10.1109/ETCM.2017.8247541 [22].

C2 **Christian Tipantuña**, Carla Parra, Jorge Carvajal, and Ricardo Llugsi. SDN: Layers, architecture and perspectives of a key technology for the future Internet. XII Jornadas en Ingeniería Eléctrica y Electrónica, FIEE, Escuela Politécnica Nacional, Quito, Ecuador, 2018 [23].

C3 **Christian Tipantuña** and Xavier Hesselbach. Demand-Response power Management Strategy Using Time Shifting Capabilities. In ACM Proceedings of the Ninth International Conference on Future Energy Systems, pages 480–485. ACM, 2018. ISBN 9781450357678. doi: 10.1145/3208903.3213519 [14].

C4 **Christian Tipantuña** and Xavier Hesselbach. NFV-enabled optimal spectrum allocation in flex-grid optical networks. In 2020 22nd International Conference on Transparent Optical Networks (ICTON), pages 1–7, 2020. ISBN 9781728184234. doi: 10.1109/ICTON 51198.2020.9203329 [24].

Research projects:

The research work in this thesis has been done within the framework of several research projects, which are listed below.

- Project 5G-City (2017-2019), TEC2016-76795-C6-1-R and AEI/FEDER, UE, funded by the Ministerio de Economía y Competitividad of the Spanish Government.
- Project SGR under Grant 2017 SGR 397 from the Generalitat de Catalunya.
- Project TRUE5G (2020-2023), PID2019-108713RB-C51/AEI/10.13039/501100011033, funded by the Agencia Estatal de Investigación of Spain.

1.5 Outline of this Thesis

This thesis is organized into seven chapters, as summarized in Fig. 1.3 and described below.

Chapter 2: Background Technologies
This chapter introduces the main concepts and ICT enabling technologies involved in the proposal for adaptive energy management, including a brief overview of 5G systems. In particular, the operational features and architectural frameworks of SDN and NFV technologies are presented in this chapter. At the end of this chapter, a brief description of DR schemas is also introduced.

Chapter 3: Problem Statement and Literature Review
This chapter describes the problem of inefficient use of available supply and desynchronization with the demand, which causes scarcity or waste of energy and extra production and pollution-related. In addition, this chapter discusses related work regarding the use of ICT technologies for energy management, including IoT and NFV technologies, and the transition to sustainable systems powered entirely or primarily by green energy.

Chapter 4: NFV/SDN Enabled Architecture for Adaptive Energy Management in IoT Scenarios
This chapter describes the architecture proposal for adaptive energy management, including the stakeholders, the global NFV-enabled ecosystem, the consumption model, the management strategies to adapt consumption to available generation,

and the mathematical characterization of the ES and the ECs. The proposed architecture considers the capabilities of current consumers (massive connectivity), advanced existing communications technologies such as SDN and NFV (that are enablers of 5G systems), and a finite capacity for energy production.

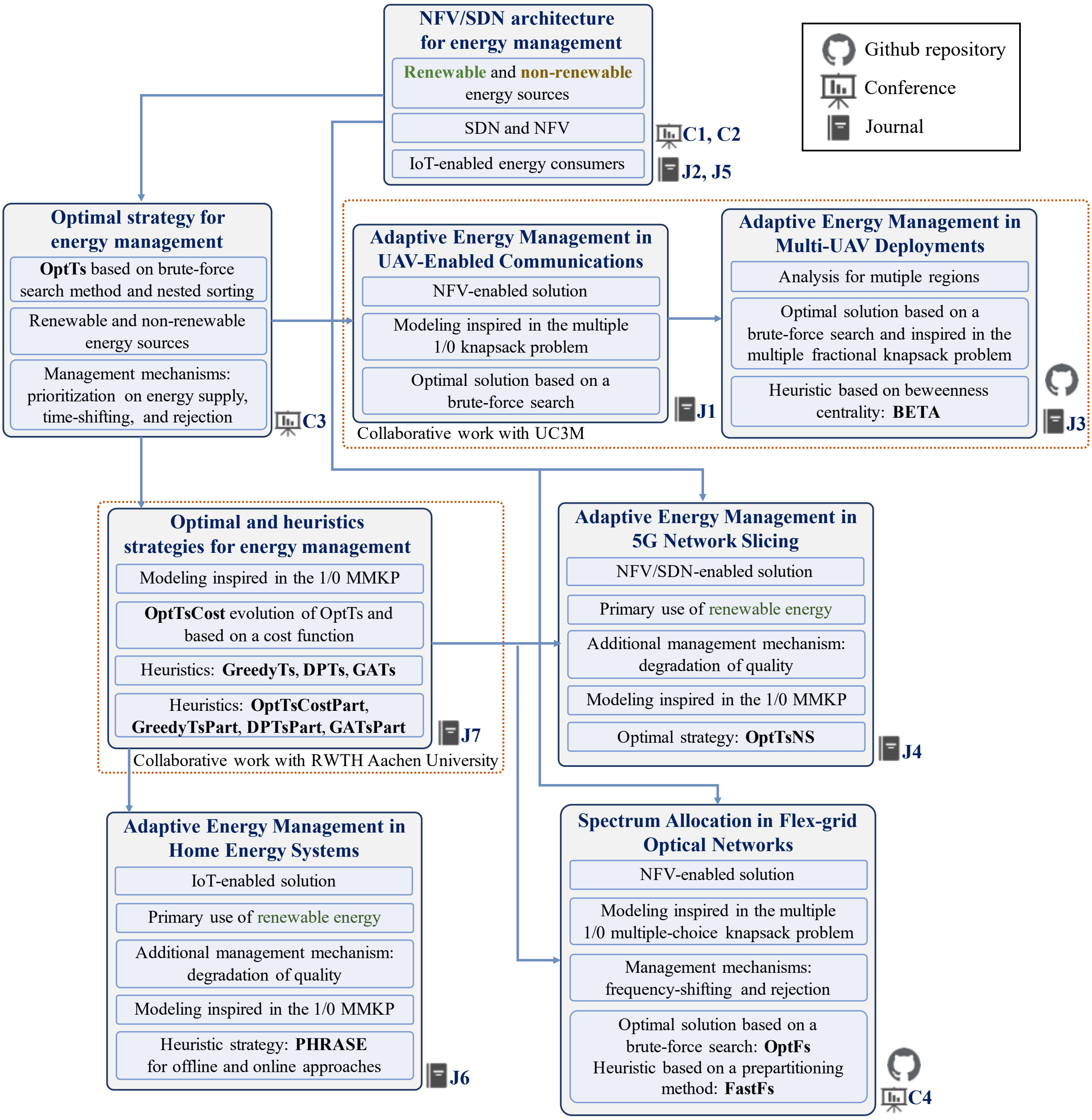


FIGURE 1.2: Thesis overview according the published and submitter content.

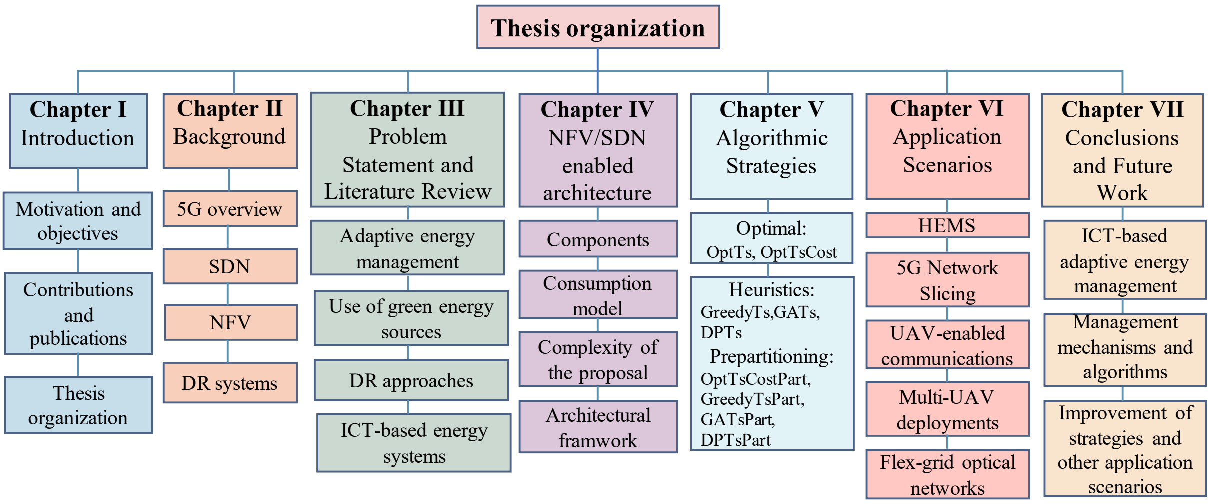


FIGURE 1.3: Thesis organization.

Chapter 5: Algorithmic Strategies for Adaptive Energy Management
This chapter presents the mathematical model related to adaptive energy consumption using an ILP formulation. The mathematical model defines the objective function and constraints related and considers finite available supply and management mechanisms such as prioritization of energy provisioning, time-shifting capabilities, quality degradation, and possible rejection of energy demands. To solve the ILP model, this chapter describes algorithmic strategies, optimal and heuristic, that, when deployed in ICT infrastructures enabled by modern technologies such as NFV, allow adaptive consumption constrained to availability. The optimal algorithmic strategy is defined as OptTsCost and is based of a brute-force search method. Instead, the heuristic solutions denoted as GreedyTs, GATs, and DPTs are based on a constructive algorithm, a genetic algorithm-based method, and a dynamic programming approach, respectively. To further scale up the operation of both the optimal and heuristic strategies, this chapter also presents a prepartitioning strategy based on a divide-and-conquer method. When it is adapted to algorithmic strategies, it enables adaptive energy management to be applied to thousands or hundreds of thousands of energy demands. The performance evaluation of the algorithmic solutions is carried out through extensive simulations in several scenarios, and the simulation results demonstrate that applying the adaptive energy management model through the proposed algorithmic strategies (optimal and heuristics) produces an improved overall performance of the generation and consumption ecosystem, and leads to efficient utilization of available power.

Chapter 6: Application Scenarios
This chapter discusses possible application scenarios of both the proposed architecture and algorithmic strategies developed, including adaptive energy management in home energy systems, 5G network sling, and in the field UAVs, also known as drones. The application for efficient spectrum allocation in flex-grid optical networks is also explored a the end of this chapter.

Chapter 7: Conclusions and future work
This chapter draws conclusions, describes possible improvements in the proposed architecture, adaptive energy model, and algorithmic strategies. Moreover, this chapter discusses other possible application scenarios and future work.

Chapter 2

Background Technologies

This chapter presents the background technologies involved in the design of the proposed architecture for adaptive energy management. This chapter also includes a brief overview of 5G systems and the general characteristics of DR schemas to promote the adaptation of consumption to availability.

The topics that are covered in this chapter are as follows:

- 5G technology description.
- SDN and NFV fundamentals and architectural frameworks.
- An overview of DR schemas for energy management.

This chapter is based on:

C1 **Christian Tipantuña** and Paúl Yanchapaxi. Network Functions Virtualization: An Overview and Open-Source Projects. In2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), pages 1–6, Salinas, Ecuador, 2017.

C2 **Christian Tipantuña**, Carla Parra, Jorge Carvajal, and Ricardo Llugsi. SDN: Layers, architecture and perspectives of a key technology for the future Internet. XII Jornadas en Ingeniería Eléctrica y Electrónica, FIEE, Escuela Politécnica Nacional, Quito, Ecuador, 2018.

C3 **Christian Tipantuña** and Xavier Hesselbach. Demand-Response power Management Strategy Using Time Shifting Capabilities. In ACM Proceedings of the Ninth International Conference on Future Energy Systems, pages 480–485. ACM, 2018.

2.1 5G Systems

Because 5G networks can be seen as a potential enabling technology for deploying different applications or verticals, this section describes the most relevant aspects of this technology.

The global deployment and adoption of Long Term Evolution (LTE) technology have promulgated mobile data with a high data rate and low latency. This trend, as well as the massive proliferation of mobile devices connected to the Internet [25], have contributed to the growth of connectivity and traffic volume, thus, to meet the current and future requirements, due to the emergence of new services and applications, several organizations and standardization bodies have conceived the development of a new architecture for mobile networks. This new generation of mobile networks, the fifth (5G), brings together the best advances in communications and network systems and has been conceived as a technology to mainly provide: (i) high data rate expected to be like around 5Gbps (fiber-like access data rate); (ii) a low latency user experience (latency less than 1 milliseconds); (iii) connectivity for more than 100 billion devices, (iv) improvements of energy, spectrum and cost efficiency; (v) faster response time and high capacity of the system; (vii) more software options to upgrade or evolve; (viii) and more data rate at cell edges [26]. To achieve these requirements, a set of technologies are focused on the different portions of the network; the main enabling technologies are briefly described below [27].

- *Millimeters wave communications*: Use of high-frequency spectrum, with a higher channel bandwidth of 1 to 2 GHz.
- *Multiple access schemes (links)*: Use of different schemas to increase the data rate, such as Sparse Code Multiple Access (SCMA), Non-Orthogonal Multiple Access (NOMA), and Orthogonal Frequency Division Multiplexing Access (OFDMA).
- *Massive MIMO*: Increase in the number of transmissions using massive MIMO or Multi-user MIMO (MU-MIMO) schemes.
- Heterogeneous networks (HetNets): Networks comprised of different types of cells such as femtocells, picocells, metro cells, microcells, macrocells, relays, or enterprise cells. These cells should support 3G, 4G, and Wi-Fi traffic.
- *SDN and NFV*: These technologies can provide dynamic network behavior, agility, and flexibility in deploying network services and applications.

5G networks are expected to provide network solutions for various public and private sectors, such as energy, agriculture, city management, health care, manufacturing, and transport. Apart from the enormous number of connections and the deployment of IoT applications, some application scenarios and use cases include: Mobile Internet with ultra-high traffic volume and mobility, ultra-high-definition multimedia augmented reality, virtual reality, 3D video, mobile health, desktop cloud, online gaming, deployment of vehicular networks, interconnection in smart homes, and improved smart grids [27].

2.2 Software Defined Networking

SDN refers to a networking technology that provides an abstraction of physical resources and network programmability by separating the control plane and the data plane, enabling innovation and simplified network management. This technology has revolutionized the world of networked systems, has become an important networking research topic in academia and industry during the past couple of years, and nowadays is considered as a key technology for the deployment of current and future communication systems and as an enabler for the future Internet. In this regard, this section addresses the concepts, fundamentals, the architectural framework, as defined by the Internet Research Task Force (IRTF) in the RFC7426, and the advantages of the SDN paradigm over traditional network infrastructures.

2.2.1 Introduction to Software Defined Networking

The networking technology used in our world is mostly built on the concepts introduced by the first networking research projects, such as the Advanced Research Projects Agency Network (ARPANET) [28]. Over the years, it is evident that the evolution of computer networks has been a great success, starting from a research project to exchange information between universities and becoming a global communications infrastructure, which serves more and more users and increases in size and complexity. Although the network technologies used in these decades (traditional technologies) have been beneficial and have worked properly, the techniques, requirements, and demands from both industry and users and the hardware and the software have evolved. Thus, for computer networks to fit current and future needs, innovations have been developed mainly in the upper layers of the underlying network and the end devices. However, network architecture has been the same over the years. Originally, computer networks were designed to provide interconnection between network devices considering principles such as heterogeneity, interconnection, and sharing. Still, other aspects such as availability, mobility, scaling, security, quality of service (QoS), quality of experience (QoE), and QoX (where X stands for service, network economics, energy, etc.), among other features, have been left out, because they were not important in the era that technology was born and because nobody thought the impact that would have the computer networks in our current society.

Vendors and operators have developed networking solutions, and although these solutions have allowed global connectivity, they have certain features that have not given way to a more rapid technological evolution. The operation of a network device is closely linked to the design and degree of innovation of vendors (i.e., it is a closed device or a black box). Every vendor has its own software solution, which is included in the hardware and specific network interfaces, protocols, and management systems. Nowadays, managing a multi-vendor network is challenging and complex, not to mention a slow standardization of protocols and delay in introducing new features. Moreover, the current networks must transfer an increasingly large amount of information (big data) [25]. To cope with the aforementioned challenges, meet the requirements of the new services, and promote innovation and open ecosystems (white-box devices), both the academy and industry have considered upgrading or redesign the network architecture. These changes have not been taken at random, on the contrary, they have been the result of decades of research and a collaborative effort of many institutions and organizations [29]. A key point in the evolution of networked systems is the advent of virtualization technologies because they have changed the landscape in the compute domain. The idea of abstracting the software from the underlying hardware and the capacity to produce changes in the behavior of a system (virtual machine) based on changes in software components was the starting point to the development of programmable networks, and later the birth of SDN technology [30].

2.2.2 Traditional Networks

A traditional network device or a network element has three well-defined planes: data, control, and management planes [31], as shown in Fig.2.1.These planes are described as follows.

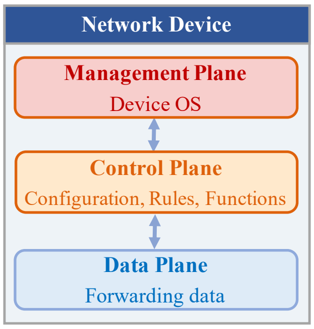


FIGURE 2.1: Data, control, and management planes in a non-SDN network device.

- *Data Plane:* The data plane is also called the data path or the forwarding plane. The data plane processes traffic or data streams (data packets); it is responsible for handling and applying actions to them based on programmed rules into lookup tables (forwarding tables). Examples of protocols in this plane are Ethernet and MPLS.

- *Control Plane:* The control plane is connected to the data plane to update the logic that should be implemented on the data streams. This plane is tasked with calculating and programming actions for the data plane, i.e., how and where to send the network traffic. The control plane is where the forwarding decisions are made and where other functions such as QoS, Virtual Local Area Networks (VLANs), etc., are implemented. Examples of protocols in this plane are OSPF, IS-IS, and BGP.

- *Management Plane:* The management plane is where it is possible to configure and monitor the network device (switch or router) through a command prompt (shell) or a web interface. The management plane usually runs on the same processor as the control plane.

Fig.2.2 shows a traditional communications network composed of network devices and physical and logical connections. Every network element contains the management, control, and data planes. However, the software is tied to the hardware where it runs, and the control and modification of the data (which is forwarding across the network) work within a proprietary framework (which means that the user is directly and indirectly linked to the solutions provided by the vendor). As a result, adopting a new solution, protocol or feature can be difficult, expensive, or take a long time.

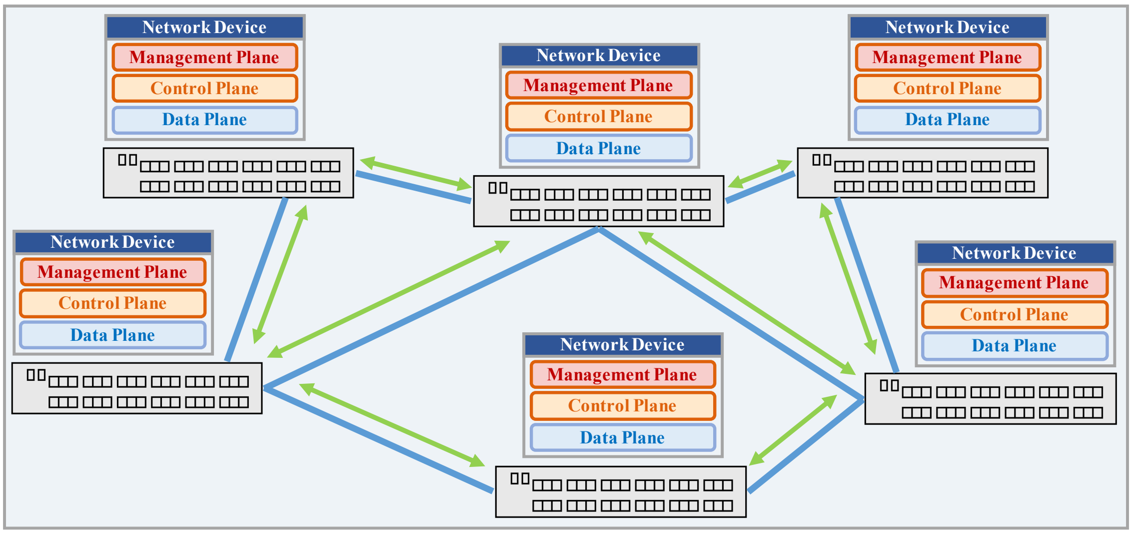


FIGURE 2.2: Traditional network topology.

The control plane within each device fulfills functions such as network device configuration, management and exchange of routing information tables, and collection of information about network topologies and status. The information used by the control plane to make decisions about where to send the traffic can be learned through the use of static routing or the use distributed routing protocols (e.g., OSPF, IS-IS, BGP). Routing techniques allow any router to exchange information with each other device within the network (distributed algorithms running between neighbors) to build routing tables used by the data plane. Then, based on the status of the network topology, the information of all devices is updated in a distributed manner, and any device can take decisions or perform actions depending on the requirements or goals to be achieved. However, because all devices have the same level, it is not easy to understand their state, history and decide which decision or action is best and who should execute it. Moreover, in the traditional network approach, the distributed routing protocols, in many cases, do not take optimal decisions because all devices have the same perspective and no device has a complete view of the network; this lack of information does not allow to take end-to-end optimal decisions. For solving the aforementioned drawbacks, SDN technology has emerged.

2.2.3 SDN Networks

SDN decouples control and forwarding functions by separating the control and data plane [30]. The separation of planes allows that the control plane becomes directly programmable and accessible by an application layer through an application program interface. As a result, the underlying network layer can be abstracted from application and network services. Thus, SDN technology changes the paradigm of networks, unifying the behavior of network devices (routers and switches) and turning them into generic forwarding devices (SDN switches or white boxes), which can be implemented using open standards and vendor-neutral solutions. The functionality of the entire network can be programmed rather than individual devices because the application is independent of the underlying hardware. The programmability offered by the SDN technology makes it possible to make changes in the network, ranging from tasks related to the routing or even a complete change in the behavior of the network. All kinds of network applications can now be programmed, for instance: change the configuration of network devices, give priorities to network traffic flow, give access or permission to the network properties, etc. Definitely, the programmability of the network introduced by an SDN enables innovation and accelerates the introduction of new features and services. SDN transforms the network perspective from specialized hardware with protocols and applications implemented for each network device (switch, router) to an open ecosystem that promotes an independent evolution of hardware and software solutions.

2.2.3.1 Abstractions in SDN

One of the most important characteristics of SDN technology is the abstraction levels. The application does not send codes directly to the network devices; instead, it communicates with the controller, and thanks to the abstraction, the network infrastructure is transparent to the application. The main levels of abstractions in an SDN network are listed below.

- *Network abstractions:* A global view of the network is available from the controller. The behavior of the entire network based on the requirements of the applications and the decisions and actions are taken by the controller.
- *Control plane abstractions:* The controller must be compatible with any hardware and software solution that is part of the infrastructure layer. The controller makes decisions based on requests by the applications and considers the network status and topology.
- *Data plane abstractions:* The network infrastructure is independent of software and hardware solutions. The configuration of the network devices can be based on flows, as proposed by the OpenFlow protocol [32], i.e., the traffic of the data will be based on flow-based forwarding instead of destination-based forwarding. In this context, a flow is a set of packet field values acting as a match rule and a set of actions to operate on all packets belonging to the same flow.

2.2.3.2 Elements of an SDN Network

To understand the changes in the network architecture, the features, and the advantages that could bring the adoption of SDN technology, it is suitable to analyze the structure and operation of the SDN-compatible network devices. Fig.2.3 shows the structure of SDN devices (a controller and a switch). Unlike a traditional network element where the control and data planes are implemented in the hardware platform as an integrated system, see Fig.2.1, in an SDN network, there is a separation between the control and the data planes, which is given by using an SDN controller and an SDN switch, respectively. Although these devices have different functionality than traditional ones, this does not limit that the network devices can be upgraded and compatible with SDN networks. There are currently network devices that can work in conventional and SDN environments. In addition to the SDN controller and the SDN switch, another component that the SDN architecture introduces is the network applications (services), as shown in Fig.2.4. A description of the main features of the application, the controller and the switch is given below.

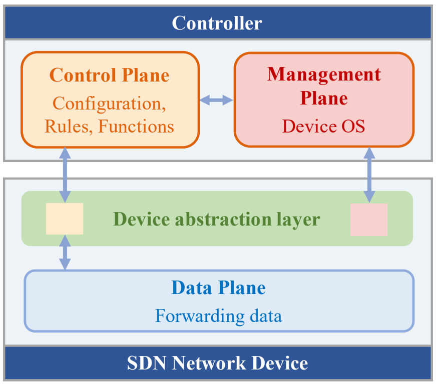


Figure 2.3: SDN network devices: controller and switch.

- *SDN switch:* The switch can be physical or virtual (network device) and does not have to implement all protocols or features, it has an abstraction layer, which allows performing minimum control and management functionalities, and which will be used in the communication with the controller through the southbound interface, namely using a control-switch protocol, such as OpenFlow [32]. The control and management functionalities in the switch are also needed for cold-start operation and configuration. The SDN switches are also called white boxes because they can be developed based on open-source standards and solutions.

- *SDN controller:* The decisions are made in the control plane, which is part of the controller, which offers central management of all devices within the network and which has an entire view of the network, as shown in Fig.2.4. The SDN network controller hides the network complexity and understands the constraints in the network. The controller software communicates with both the network application and the underlying network through interfaces, northbound and southbound interfaces. When the control plane is separated from individual network devices (network infrastructure), and it is centralized in a single device (SDN controller), it can perform better management of multiple network elements (SDN devices), and it can provide underlying network abstraction, which facilitates the deployment of network applications (programmability). The SDN controller has the following characteristics:

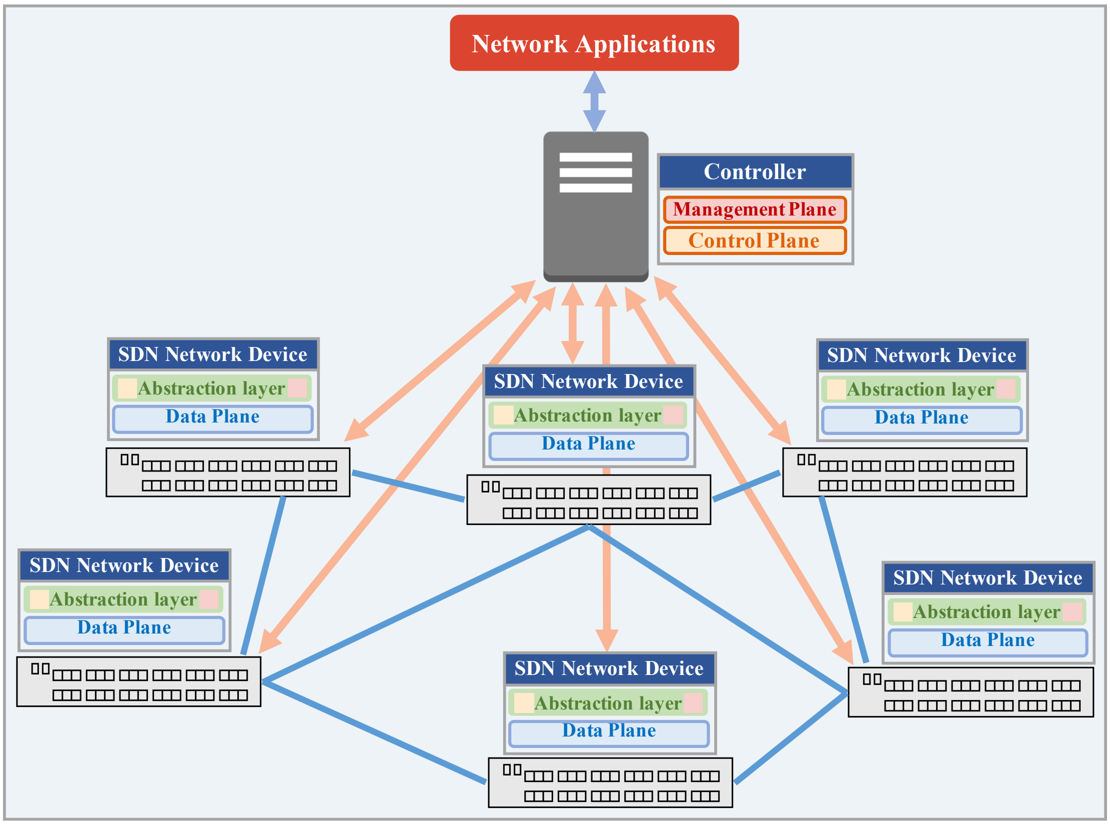


FIGURE 2.4: An SDN network with one controller and multiple SDN-switches.

- Logically centralized software platform. It offers a unique abstract view of the network state (e.g., topology).
- The SDN controller is sometimes referred to as the Network Operating System (NOS) [33], because it acts as such. Network applications run on the SDN controller, similar to computer applications running on a computer operating system.
- For scalability and reliability (e.g., to support very large and complex networks), the controller can be distributed in different servers.
- There are commercial and open-source solutions. Examples of open-source controllers are Beacon, Floodlight, Trema, NOX, POX, Ryu, and OpenDaylight [34], the latter being one of the most relevant projects today. The controller should truly interoperable and multivendor, and its operation cannot be tied to a specific network element.
- The controller can dynamically adapt the data flows.
- The controller has APIs (open) to interact with both applications and the data plane.

• *Network applications:* The network applications running on the controller can be diverse and can range from routing tasks to the implementation of network services such as network access control, load balancers, firewalls, etc. The application communicates with the controller using the northbound interface; REST API and Java API are examples of this interface.

2.2.3.3 SDN Benefits

There are significant benefits related to SDN technology; among the most important ones are abstraction capability and programmability. The control plane becomes directly programmable through APIs, and the underlying network can be abstracted from applications and network services, allowing a programmatically control of network resources. Other benefits of SDN technology include:

- Dynamic network behavior and a central view for running network services and applications.
- Better utilization of network resources (physical or virtual).
- Improvements in network performance, making the network more manageable, cost-effective, and adaptable to ongoing dynamic requirements.

2.2.4 SDN Architecture

The Open Networking Foundation (ONF) was founded in 2011 to promote SDN by developing open standards, such as OpenFlow. Moreover, ONF provides an architectural framework [35], defining the roles of application, control, and data planes and the interfaces to interconnect them. Although in [35] the description of the layers is presented in a friendly way, over time and with the increasing popularity of SDN, a bit of confusion was created regarding the layers, interfaces, and the architecture itself, so that in 2015 as a response to clarify the SDN concepts and terminology, the Internet Research Task Force (IRTF) through its Software-Defined Networking Research Group worked intensively in a proposal, resulting in the RFC7426 [36], which defines in a more detailed way the SDN layers and architecture. Fig.2.5 depicts the architectural framework defined by the IRTF.

The SDN architecture gives the applications information about the state of the entire network from the controller instead of traditional networks where the network is application-aware. In addition, the SDN architecture is: (i) directly programmable, because of forwarding functions decoupling; (ii) agile, due to the abstraction that lets networks administrators dynamically adjust network-wide traffic flow to meet changing needs; (iii) centrally managed given that the controllers maintain a global view of the network; (iv) and open standards-based and vendor-neutral because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols. A description of the planes, the interfaces, and the abstraction layers that make up the architecture are presented below.

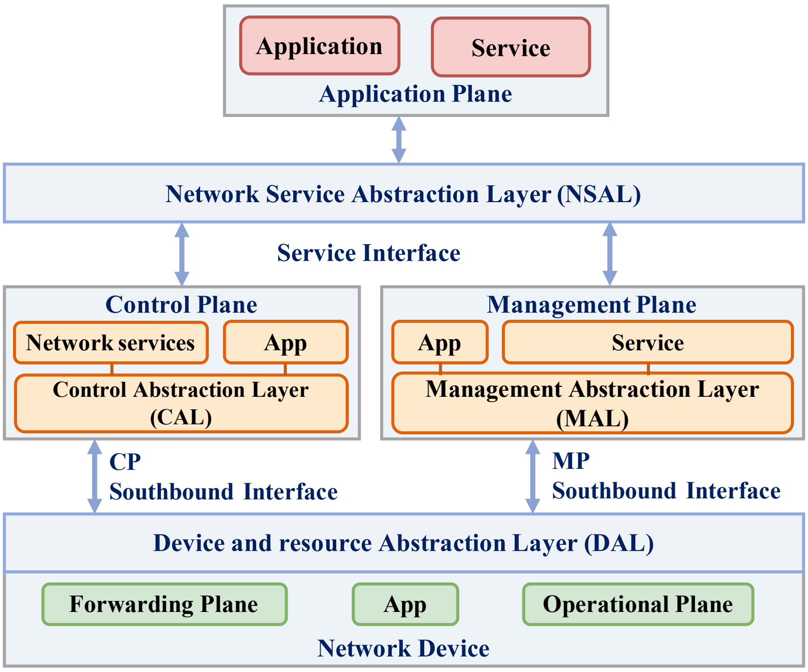


FIGURE 2.5: SDN architecture, adapted from [36].

- *Application plane:* This plane represents the applications and network services. SDN Applications are programs that communicate behaviors and needed resources to the SDN Controller via APIs. The applications can build an abstracted view of the network, and they interact with the controller collecting information and requesting certain requirements for decision-making purposes. Network applications can be very varied, and they could include networking management, analytics, or business applications.

- *Network services abstraction layer:* This layer provides access to services of the control and management planes to other services and applications.

- *Northbound interface:* Also called service interfaces is a set of APIs available to network administrators (application developers), for example, CLI (Command Line Interface), GUI (Graphical User Interface) REST API, and JAVA API, which offer the ability to pragmatically shape traffic and launch services. These APIs can be used to facilitate innovation and enable efficient orchestration and automation of the network. The northbound interface is used to communicate between the SDN Controller and the services and applications running over the network; this interface abstracts the low-level instructions to program forwarding devices needed to develop the network applications.

- *Control plane:* The control plane is composed of the controller, which acts as the brain of the network and provides an entire view of the network, allowing network administrators to instruct network devices (switches and routers) on how the network traffic should be forwarded. The controller is a logical entity that retransmits the instructions and requirements received from the applications to the underlying network. The information communicated to the applications includes statistics and events about what is happening within the network.

- *Control abstraction Layer:* Abstracts the control plane southbound interface and DAL from the applications and services of the control plane.

- *Management plane:* The management plane is in charge of monitoring, configuration, and maintenance of network devices. This plane interacts mostly with the operational plane than with the forwarding plane.

  - *Management abstraction layer:* Abstracts the management plane southbound interface and the Device and Resource Abstraction Layer (DAL) from the applications and services of the management plane.

- *Southbound interface:* The southbound interface allows the controller to communicate with the connectivity devices (data plane). This interface facilitates efficient control over the network, enabling the SDN controller to dynamically make changes according to real-time requirements, demands, and needs. The southbound APIs can be open or proprietary. The most well-known open-source southbound interface is OpenFlow (promoted by the ONF) and recently the P4 programming language [37], which is widely spread for this organization.

- *Device and resource abstraction layer (DAL):* Abstracts the resources of forwarding and operational planes of the device to the control and management planes. The services that the rest of the planes provide depend on this abstraction layer.

- *Forwarding plane:* Also known as the infrastructure layer or data plane, it is composed of connectivity devices, physical or virtual. This plane is responsible for handling data packets according to the instructions provided by the controller. Some actions that are performed in this plane are forwarding and dropping packets.

- *Operational plane:* This plane manages the operational state of the network device, i.e., the activity or inactivity, the status and number of ports, the computational capacity (processor and memory), among others parameters.

2.2.5 SDN Perspectives

There is a large number of fields and applications where SDN technology can be developed and applied. Some prominent areas and topics are:

1. *Open-source solutions and ecosystems*: Since its conception, SDN has promoted the development and the use of open-source interfaces and ecosystems. There is a large number of active projects, as classification is available at [22].

2. *NFV*: SDN may provide a dynamic network behavior and ensure the delivery and quality of the network traffic between virtualized functions [38].

3. *5G networks*: SDN is considered a key enabler for deploying 5G networks and working together with NFV; these technologies can offer dynamic network behavior, agility, and flexibility in deploying network services and applications [39].

4. *Transport networks*: Used in transport networks, SDN offers full-fledged virtual networks to the customers. An example is the first SDN WAN implementation deployed by Google in 2013 [40].

5. *IoT*: SDN is considered a promising architecture to meet the scalability and heterogeneity demanded by IoT devices and services [41].

6. *Smart grids*: SDN provides the power grids with better network monitoring and improved network management and programmability, which allows the evolution of energy systems and sectors [42].

2.3 Network Functions Virtualization

NFV has emerged as a networking technology from the telecom industry to provide agility and flexibility in the deployment of network services and to reduce the Capital Expenditures (CAPEX) and the Operating Expenses (OPEX) by leveraging virtualization and cloud technologies. NFV decouples the software implementation of network functions from the underlying hardware. It provides an abstraction of network functions such as firewalls, deep packet inspectors, load balancers, among others, via software components that can run on general-purpose devices that can be located in a variety of telecom infrastructure, including data centers, network nodes, and end-user facilities. These Virtual Network Functions (VNFs) can easily be created, moved, or migrated from one equipment to another without installing new specialized hardware, allowing faster deployment of the services and providing innovation and a great number of opportunities for the world of networked systems. This section presents an overview of NFV technology, describing its characteristics, enabling technologies, benefits, and architectural framework.

2.3.1 Introduction to Network Functions Virtualization

Currently, networking technology is experiencing a software revolution; the network functions and services are moving from hardware mode to software mode, providing the opportunity to have programmable networks and services [43]. Telecoms networks contain an increasing variety of proprietary hardware appliances. To launch a new network service is required not only for another appliance but also space, power consumption, technical skills, and a big effort to integrate and deploy the service [43]. The challenges experienced by operators include: shorter life-cycle of devices, the software bundled with hardware, vendor-specific interfaces, slow protocol standardization, a long delay in introducing new features, long and complex upgrade cycle, complex configurations in some cases, services are mostly static and dedicated management systems. Networks operators are addressing with increases in CAPEX (investment needed, e.g., dedicated and expensive proprietary hardware) and OPEX (costs of operation and maintenance, e.g., high maintenance cost,

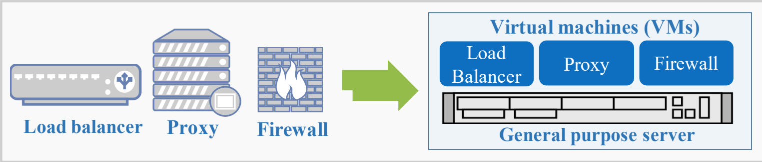


FIGURE 2.6: Vision for NFV.

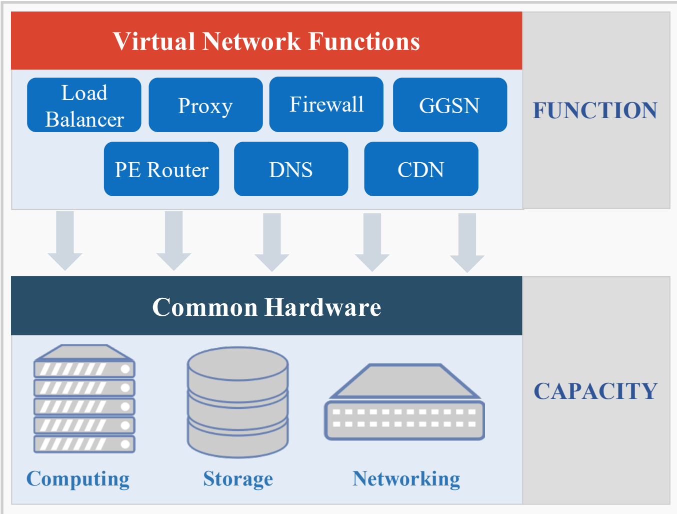


FIGURE 2.7: NFV separation of functionality from capacity, adapted from [45].

power consumption, training), and they are facing a reduction in the return on investment and constraints on innovation [44]. The network operators noticed that the market requirements would not be achieved in the long term due to network capabilities. For these reasons and those mentioned above, and to deal with current and new services, capacity, and functionality requirements, NFV technology has been conceived. NFV faces these problems by implementing network functions in software, i.e., moving network functions to software, by leveraging standard IT virtualization and cloud technology, consolidating many network equipment (functions) onto industry standard high volume servers [44].

NFV is focused on virtualizing network functions such as proxies, load balancers, firewalls, gateways, i.e., any network function, running in specialized hardware and migrating them to software-based devices running on virtual machines (VMs). These VMs (VNFs) can run in general purposes servers, and all hardware resources (servers, storage, and networking devices) are managed as a common resource pool. Thus, they can be moved or instantiated in various locations in the network as required without installing new equipment. Fig.2.6 shows a pictorial representation of the NFV vision. NFV separates functionality from capacity, i.e., it decouples network services from the hardware that delivers them; as shown in Fig.2.7, this decoupling increases network elasticity and promotes heterogeneity.

A VNF itself does not provide a service to the end customer; to create a service, it is used the Service Chain concept, a concept created by the European Telecommunications Standards Institute (ETSI) [46]. The Service Chain (SC), also known as Service Chaining

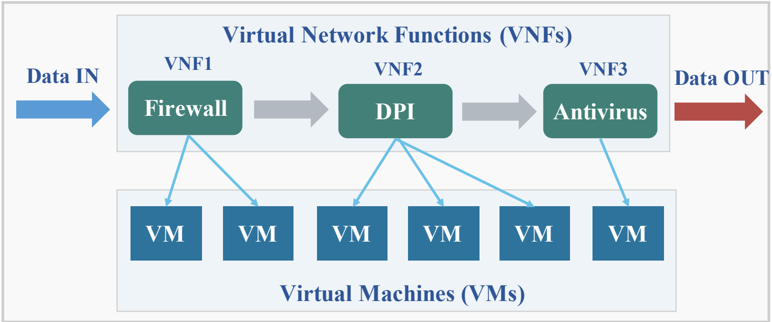


FIGURE 2.8: Example of the deployment of a service through the use of a service chain, adapted from [46].

or Service Function Chaining (SFC), is a sequence of multiple VNFs executed in a given order to deliver a service, a service in the NFV realm is formed by a chain of VNFs. Fig.2.8 shows an example of a security system composed of three VNFs, which can be deployed in one or more VMs. In NFV, network services are built by chaining a set of VNFs that must be allocated on top of the physical network infrastructure (commodity hardware); this problem is the so-called resource allocation, placement, or embedding problem [47].

2.3.2 NFV Enablers

The main enabling technologies for the development of NFV are virtualization, cloud technologies, and the rising economy of scales related to the production of standard servers [43]. Network virtualization technology creates an abstracted virtual network on top of a physical network, allowing many multi-tenant networks to run over a physical network. The services associated with the deployment of these virtual networks can be executed in multiple racks in DCs, in telecommunication nodes also known as Point of Presence (PoP), or even near the user location if necessary. In the first attempts to deploy VNFs, it was conceived that each VNF would require a different server, i.e., one server per VNF. Still, this approach was envisioned as unfruitful because of the excess of unneeded resources, and thanks to the use of virtualization technologies, the VNFS can run on VMs, and hardware resources can be managed as a common resource through the use of a hypervisor layer (i.e., maximizing the use of hardware resources).

The virtualization technology and cloud computing principles provide NFV technology a dynamic operation and on-demand deployment of services. Furthermore, leveraging the economies of scale of the IT industry, specifically with the industry of standard servers, has helped to change the mentality of the networked systems, moving from expensive proprietary purpose-built platforms to low-cost generic platforms. While the technologies described above are essential, it should be remembered that many services could not be implemented without the improved Internet connection speeds provided by ISPs.

2.3.3 SDN and NFV

NFV separates capacity (hardware) and functionality (software); meanwhile, SDN provides programmability to the network, separating the control and data plane. SDN focuses on the virtualization of network devices, and NFV aims to enable the virtualization of network functions and services. Also, SDN is promoted by the Open Networking Foundation (ONF), and NFV is promoted mainly by ETSI and recently by the Internet Engineering Task Force (IETF). SDN and NFV are totally independent technologies. Network functions can be virtualized and deployed without SDN. Likewise, the separation of the control and data plane in SDN can be performed without NFV. Still, these technologies are inherently complementary; in fact, the research in networked systems has allowed the existence of these technologies, foreseeing that working together can provide many benefits and developments in the telecommunications realm. NFV may improve the efficiency and flexibility of SDN control plane services, and SDN may ensure the delivery and quality of the network traffic between NFV's virtualized functions. SDN can play a significant role in the orchestration of the NFV infrastructure resources enabling features such as provisioning and configuration of network connectivity and bandwidth, automation of operations, security, and policy control. The SDN controller can be viewed as a component of the NFV infrastructure, and as such, can efficiently work with orchestration systems and control both physical and virtual resources. On the other hand, the SDN controller could be part of a service chain and other VNFs.

Because the goals of SDN and NFV are similar, i.e., to reduce equipment costs and decrease time to market while attaining scalability, elasticity, and a strong ecosystem, since 2014, the ONF has considered has a part of the SDN architecture the NFV technology [38]. A pictorial representation of this architecture is shown in Fig. 2.9, in which NFV depicts the application layer while SDN provides the underlying layers that facilitate the deployment of network services.

The deployment of NFV through their VNFs (such as firewalls, deep packet inspectors, load balancers, etc.) requires large-scale dynamic network connectivity both in the physical and virtual layers to interconnect VNFs endpoints. An SDN solution (e.g., an Openflow or P4 approach) can provide this connectivity level. To summarize, NFV and SDN are deeply related; together, they can offer a great opportunity to change how to conceive and build networks, increasing profits and reducing complexity, and changing paradigms of traditional networks.

2.3.4 NFV Benefits, use cases, and challenges

2.3.4.1 Benefits

The main benefits that NFV technology brings are reduction in CAPEX and OPEX through reducing equipment costs and related power consumption, reduced time-to-market to deploy new network services and improved return on investment from new

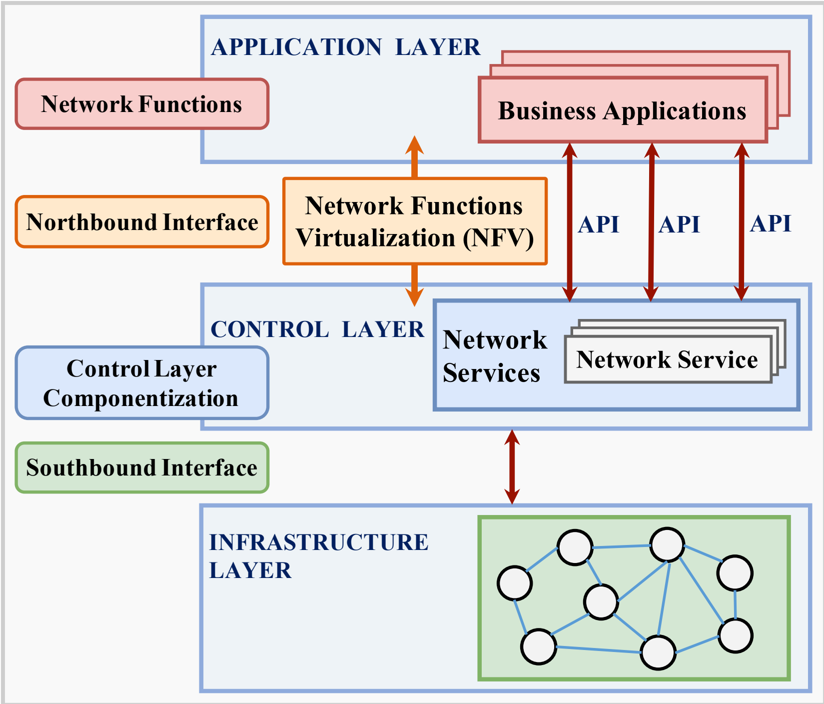


FIGURE 2.9: Integration of NFV/SDN, adapted from [38].

services, greater flexibility to scale up, scale down or evolve network services, and opportunities to deploy new innovative services at lower risk [43]. Below is a list of the NFV benefits related to the reduction in CAPEX and OPEX.

- Reduced CAPEX
  - Leveraging Commercial Off the Shelf (COTS) hardware.
  - Changing from proprietary purpose-built platforms.
- Reduced OPEX
  - Less variety and few numbers of appliances to deploy and maintain.
  - Faster upgrade cycles, both hardware and software components.
  - Faster time to market, enabling service innovation.
  - Adoption of open-source solutions.
  - Support of multi-tenancy, tenants coexisting on the same hardware.

2.3.4.2 Use cases

NFV technology can be applied in various scenarios and in fixed and mobile networks; some examples of use cases are listed below [46].

- *Switching elements:* Routers, broadband network gateways, etc.

- *Mobile network nodes:* IP multimedia subsystem, home location register, radio network controller, node B, eNode B, etc.
- *Tunneling gateway elements:* IPSec/SSL virtual private network gateways.
- *Traffic analysis:* Deep packet inspectors, QoS, Quality of Experience (QoE) measurements, traffic monitoring and Service Level Agreement (SLA) monitoring.
- *NGN signalling:* Session border controller, signalling gateways, etc.
- *Application-level:* Content delivery networks, cache servers, load balancers, and application accelerators.
- *Security functions:* Firewalls, virus scanners, intrusion detection systems, etc.
- *Virtualization of home devices:* Set top boxes, home routers and switches.

2.3.4.3 NFV Challenges

Despite the multiple benefits that NFV provides, some challenges have to be taken into account and addressed for the deployment of network services [48]. These challenges are listed below.

- *Management:* Perfect integration with different hardware vendors, end-to-end automation and orchestration, virtualized network platforms will be simpler to operate than those that exist today.
- *Performance:* Comparable performance with Physical Network Functions (PNFs), i.e., with traditional network functions.
- *Reliability and stability:* Availability of services in carrier-grade networked systems. The stability of the network must not be affected when managing and orchestrating a large number of VNFs from different hardware vendors and hypervisors.
- *Security and resilience:* Suitable agreements between multiple tenants and network operators to manage and control the physical and the virtual infrastructure and the automation and orchestration processes, a VNF should be as secure as a PNF. Also, VNFs must be recreated on-demand after a failure.
- *Portability/Interoperability:* Ability to load and execute VNFs in different but standardized data center environments provided by different vendors and different operators.
- *Migration and co-existence with legacy platforms:* NFV must work in a hybrid network composed of classical PNFs and VNFs.
- *Management, orchestration, and automation:* A consistent management and orchestration architecture is required to leverage the flexibility of VNFs in a virtualization environment.

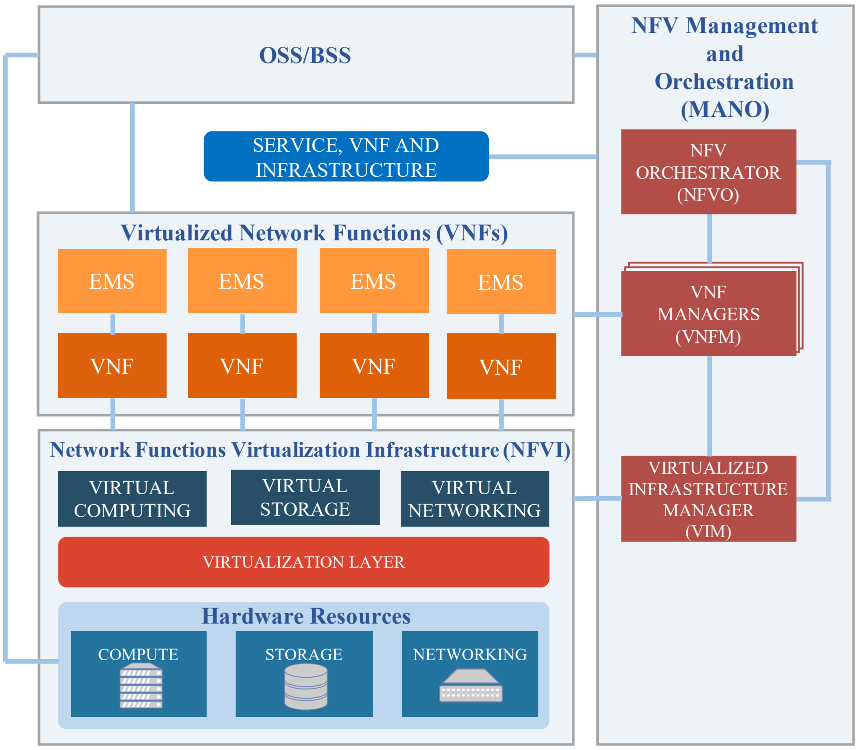


FIGURE 2.10: NFV architecture, adapted from [44].

- *Minimizing energy consumption:* Minimization of energy consumption through consolidation, shifting, or migration techniques.
- *Standardization:* Analyze requirements for technical specifications and standards.

2.3.5 NFV ETSI

ETSI NFV is an initiative started in October 2012 when a group of vendors and operators created a new Industry Specification Group (NFV ISG) and published a white paper describing the objectives, the motivations, and the use cases [43]. Nowadays, there exists substantial literature about NFV on the Internet, like the one available at `https://www.etsi.org/technologies/nfv`.

2.3.5.1 NFV Architecture

NFV architecture comprises three main components: VNFs, VNF Infrastructure (VNFI), and Management and Orchestration (MANO), as depicted in Fig.2.10.These components are described below.

- *Virtualized Network Functions (VNFs):* A VNF is a software implementation of a network function that is capable of running over the NFVI. A VNF can run on one or more VMs, and it is managed by an Element Management System (EMS), which responsible for its creation, configuration, monitoring, performance, and security. The EMS provides fundamental information required by Operations Support System (OSS) in the environment of a service provider. The EMS performs management functionalities for one or several VNFs.

- *Network Functions Virtualization Infrastructure (NFVI):* Physical and software resources, and the virtualization layer, on top of which VNFs are executed. The NFVI-PoPs include processing, storage, and networking resources. The NFVI is implemented as a distributed set of NFVI nodes deployed in various NFVI PoPs as required to support the locality and latency objectives of the different use cases and fields of application. Virtualization is an important element in the NFVI domain because it abstracts the hardware resources and decouples the VNF software from the underlying hardware, thus ensuring a hardware-independent lifecycle of VNFs.

- *Management and Orchestration (MANO):* MANO includes all the management and orchestration functions (management of physical and/or software resources) required for managing the lifecycle of VNFs on top of the NFVI. In addition, the NFV MANO also interacts with the (NFV external) OSS/BSS landscape, allowing NFV to be integrated into an existing network-wide management landscape. The NFV MANO is composed of three key functional blocks: NFV Orchestrator (NFVO), VNF Manager (VNFM), and Virtualized Infrastructure Manager (VIM).

  - *NFV Orchestrator (NFVO):* Performs orchestration functions of NFVI resources across multiple VIMs, instantiates VNF Managers and performs the lifecycle management of network services. The NFVO interacts with the OSS/BSS for provisioning, configuration, capacity management, and policy-based management. NFVO also manages the network service deployment templates and VNF packages. There is usually only one orchestrator that oversees the creation of a network service.
  - *VNF Manager (VNFM):* Performs orchestration and management functions of VNFs. The VNFM interacts with the EMS and the VNF for provisioning, configuration, and fault and alarm management. The VNFM is in charge of managing the lifecycle of VNF instances, it is responsible for: initialize, update, query, scale and terminate VNF instances. Each VNF instance must be associated exclusively with a VNFM.
  - *Virtualized Infrastructure Manager (VIM):* Performs orchestration and management functions of NFVI resources. The VIM is responsible for controlling and managing the NFVI resources, including compute, storage, and network resources. VIM provides functionalities for allocating, upgrading, and releasing NFVI resources, and it manages the association of the virtualized resources. In addition, it is in charge of managing VNF Forwarding Graphs (service chains) to create and maintain virtual links, virtual networks, subnet, and ports. Multiple VIMs instances may be deployed in a communications network.

- *Operations Support System (OSS) and Business Support Systems (BSS):* OSS is the general management system that, together with BSS, helps providers to deploy and manage various end-to-end telecommunications services (such as orders, billing, troubleshooting, etc.). OSS deals with: network, fault, configuration, service, and element management; meanwhile, BSS deals with: customer, operations, order, billing, and revenue management.

2.3.6 NFV projects and research

There are a large numbers of open-source solutions, not only for NFV but also for SDN, among the most well know are: Openflow [32], Mininet [49], OpenDaylight [34], OpenStack [50], OpenMano [51], Open Source Mano (OSM) [52] and Open Platform for NFV (OPNFV)[53]. A classification of more than 170 open-source SDN/NFV projects is available in [22]. Regarding the research in the NFV domain, there are various research areas that can be explored; some examples are listed below.

- Service chaining algorithms.
- NFV orchestration algorithms.
- Abstractions for carrier-grade networks and services.
- Performance studies: scheduling, optimization, portability, and reliability.
- Security of NFV infrastructure.
- Impacts of data plane workloads on computer systems architectures.
- Performance monitoring and reliability of network services.
- Energy-efficient NFV architectures.
- New network topologies and architectures.
- Tools and simulation platforms.

2.4 Demand Response Systems

Nowadays, modern power grids or smart grids are becoming critical infrastructures that require a stable state/equilibrium (in energy terms) to allow the coexistence of different energy providers or suppliers, consumers, and storage systems. However, with the increase of energy consumed, the proliferation and adoption of renewable energy sources, a challenge that is insight is to keep stable the performance of the power grids despite the weather conditions or the fluctuations of the energy demands. A perfect match between power generation and the power consumed by the users is a goal to be achieved. Under this approach, DR mechanisms have been proposed [54].

The DR schemes, mechanisms, or programs consist of a set of requests and actions exchanged between the ES and the ECs with the aim of promoting consumer participation in energy management by allowing the modification of consumption according to energy provisioning [8]. The conventional strategies in power grids ensure their performance and reliability through the generation of an excess of power. Modern systems that use DR technology instead provide a solution in which the ECs play a significant and active role in energy management by modifying their consumption levels, i.e., by increasing, reducing, or shifting their energy usage, according to the amount of available energy and the operating conditions of the system [8]. There are several benefits associated with DR technology, some include: increased system stability and efficiency, reduced CAPEX and OPEX for peak load demand, reduced average power generation costs and lower electricity tariffs for customers [55].

The DR schemes seek to adapt the consumption profile to energy generation conditions through voluntary collaboration between the ES and the ECs. This collaboration can be carried out based on agreements or contracts. Methods of engaging users in DR initiatives can be given in terms of time (free periods) or price agreements (reduced electricity bills), being the latter the most adopted option [54]. Besides, other kind schemes can be considered, for instance, initiatives based on auctions, or programs that stipulate penalties if one of the parties does not cooperate with the other [56]. An example of contractual terms and agreements between energy suppliers and consumers is available in [11]. In the DR scope, the price of the electricity is not fixed, on the contrary, the tariffs are flexible, and they change over time according to the availability of energy in a given time. Depending on the agreement or contract between the supplier and the consumer and the amount of energy, different strategies, and management mechanisms can be implemented. For instance, if the energy supplier has a surplus of energy, it can stimulate consumption (e.g., execution of tasks or jobs in advance, the execution of concurrent/parallel tasks, maintenance jobs, or offer a better quality/performance for services and applications) by offering low-price energy. On the other hand, if the energy supplier experiences an energy shortage, it can motivate deferral energy consumption, the no execution of services, or higher-price energy.

The DR programs can be implemented using ICT infrastructures, such as DCs [57]. Then, these ICT-based infrastructures, as part of DR and acting as an Energy Manager (EM) entity, can execute strategies such as workload scheduling to coordinate and adjust energy provisioning and consumption. Moreover, ICT-based infrastructures can be enabled by sophisticated communications technologies, such as NFV [43] and then offer smarter programmable energy management solutions. Based on all the aforementioned characteristics, the DR approach has been used as the basis for the development of our proposal for adaptive energy management.

Chapter 3

Problem Statement and Literature Review

This chapter describes the problem of the lack of synchronization between energy consumption and generation, which causes energy scarcity, waste, or inefficient utilization. Research work related to energy management through the use of ICT-enabled solutions is also discussed in this chapter. The topics that are covered in this chapter are as follows:

- Description of the problem of inefficient use of available supply and desynchronization with the demand.

- Research work related regarding the use of ICT technologies for energy management.

This chapter is based on:

J2 **Christian Tipantuña** and Xavier Hesselbach. NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy. *IEEE Open Journal of the Communications Society*, 1:357–380, 2020.

J6 **Christian Tipantuña** and Xavier Hesselbach. IoT-enabled proposal for adaptive self-powered renewable energy management in home systems. *IEEE Access*, 9:64808–64827, 2021.

J7 **Christian Tipantuña**, Xavier Hesselbach, and Walter Unger. Heuristic Strategies for NFV-Enabled Renewable and Non-renewable Energy Management in the Future IoT World. *IEEE Access*. doi: 10.1109/ACCESS.2021.3110246.

C3 **Christian Tipantuña** and Xavier Hesselbach. Demand-Response power Management Strategy Using Time Shifting Capabilities. In ACM Proceedings of the Ninth International Conference on Future Energy Systems, pages 480–485. ACM, 2018.

3.1 Problem Statement

Because the energy production is finite and the power demanded by the users is not always synchronized with the power generated (a condition presented in conventional power grids), two situations occur periodically: (i) the energy is wasted because it cannot be consumed or stored (i.e., periods of energy surplus), and (ii) the demands of users cannot be processed due to lack of energy (i.e., periods of energy scarcity). This inefficient use of available energy translates in turn into partial or total energy shortage during certain periods (peak load hours) or, in certain places, an increase in tariffs for consumers, among others. Moreover, the inherent operation condition of conventional energy systems is affected by the proliferation of non-traditional renewable energy sources such as solar, wind, which, when integrated into energy systems, deliver an intermittent resource conditioned by geographical position or environmental conditions. Likewise, customer-side renewable energy generation (e.g., through photovoltaic installations) increases the difficulty (complexity) in managing demand and distributing the available energy resource.

To face the shortage or lack of energy, a possible solution is to use a greater amount of energy resources (i.e., extra energy generation). However, this solution is not sustainable since the resources (such as gas, fuel, or coal) used in energy production are limited, non-renewable, and produce pollution when processed (i.e., they directly increase the carbon footprint). In fact, the design and operation of current energy systems does not guarantee universal energy access. A study carried out in 2016 [58] reported that nearly one-fifth of the world population (7.2 billion in 2016) is deprived of electricity. This situation will be of even more critical concern in the near future due to the increasing number of users, devices, and services. In addition, higher energy production requires additional investment in infrastructure for the conversion and distribution of this resource to consumers, which directly impacts investment and maintenance costs and tariffs for end-users.

On the other hand, to solve the problem of energy waste, an alternative is using storage elements such as battery units. Unfortunately, the amount of energy stored in these devices is small compared to the energy resource produced or wasted. Although the technology for manufacturing batteries has evolved in recent years, mainly due to the development of electric vehicles [59], their lifetime is limited (e.g., a couple of years), their replacement is mandatory, and, once they have reached their lifetime, these elements can be sources of pollution (e.g., the release of heavy metals or chemicals into the water) [60]. For these reasons, massive energy storage to avoid energy misuse and waste is not a viable solution. Currently, energy storage solutions such as battery units are only used as backup units (e.g., for a few minutes) in the event of energy shortage periods or as elements that help the transition to other energy sources (e.g., during the activation of diesel generators to power cellular stations after an energy outage). In addition, nowadays, several initiatives promote the minimum or zero use of batteries in communication systems, such as battery-free IoT networks [61].

Because extra energy production to cope with scarcity or massive storage to deal with the abundance or energy waste are unsustainable and inefficient alternatives, there is a pressing need to develop new supply and consumption management solutions. In response to these limitations presented by traditional energy systems (in terms of energy management), adaptive management or consumption systems emerge (through DR schemes) as prominent solutions, allowing interaction between the supplier and consumer to adapt the consumption patterns to the available supply [?]. In this context, this research work aims to propose an adaptive energy management solution focused on the efficient (optimal) consumption of available power, whether or not it is renewable. In summary, the proposed solution encompasses: (i) an ICT-enabled architecture that based on advanced technologies such as SDN and NFV allows the interaction between the supplier and consumers (i.e., consumer-sider participation) and offers the computational capacity needed to deploy adaptive energy management strategies; and (ii) different management mechanisms (e.g., workload scheduling using time-shifting capabilities) that implemented as algorithmic solutions and running on ICT infrastructures leads to efficient (optimal) adaptation of consumption patterns to available generation profiles. The proposed solution can prioritize the use of green energy and the distribution of the available supply. Moreover, it can work in offline and online approaches, can be used to improve (optimize) energy planning (e.g., avoiding power peaks) and management, and can be applied to a wide range of IoT-enabled infrastructures such as HEMS, UAVs, electric vehicles, building, neighborhoods, small cities or locations, starships, among other application fields.

3.2 Literature Review

This section reviews the related work. Section 3.2.1 describes the ICT participation in energy management systems, including IoT and NFV-based approaches. Then, Section 3.2.2 summarizes the key features and contributions of our proposal and differentiators with existing approaches.

3.2.1 ICT-Based Energy Management Systems

In the last decades, with the deployment of smart grids, several proposals have analyzed the impact on energy use and consumption when communications systems work together with energy systems [62]. From this perspective, the technological term of the IoE has been introduced to refer to a complex and sophisticated Internet-type network for the next generation of power grids [15, 63]. The IoE paradigm promises robustness and reliability in energy systems and is the result of the integration of advanced ICT infrastructures (*e.g.* IoT) into the power grids to carry out automation, monitoring, and management tasks, taking in to account generation, distribution, storage, and consumption factors[63]. Unfortunately, the existing energy infrastructure is not immediately ready to offer an IoE, and several operational changes and functionalities must be introduced in current energy systems. The most relevant requirements for the deployment of IoE solutions are presented in what follows [15, 62, 63].

- *ICT infrastructures between the ES and the ECs:* A robust and scalable communications infrastructure to support the bidirectional information flow between the ES and the ECs for energy management is an essential requirement for implementing the IoE. In this regard, the ICT infrastructure deployed must allow a hierarchical and scalable operation if necessary, in order to guarantee optimum performance of the energy system and use of resources (communications and energy). From the consumer side, the deployed IoT infrastructures are seen as key participants in future energy management systems. In addition, because future energy systems will consist of a tremendous amount of interconnected components in generation (distributed power plants, substations, energy storage components, and metering systems) and consumption (ECs with different requirements), it is necessary that the ICT infrastructure used for the IoE have high-performance computing capacity/resources to carry out all monitoring and management processes for the stakeholders (specially for ECs). Then, with the information of generation and consumption, different analytics can be performed and planning or forecasting actions [63].

- *Demand side management and efficient use of produced energy:* Traditionally, all energy management actions are carried out only on the ES premises. Modern power grids require the participation of the ECs in the energy management process. Enabled by ICT infrastructures (*e.g.*, IoT deployments), the ECs can, for example take part in DR programs through an intermediary or directly with the energy utility, with the objective of effectively managing load peaks, load reduction, and the fluctuation of energy generation from renewable sources [15]. The interoperability between the ES and the ECs and the actions carried out by the latter can lead to efficient energy utilization, with reduced or minimal waste, which is of paramount importance for future energy systems [64].

- *Use of renewable energy sources and transition to systems powered entirely with green energy:* A important requirement for a sustainable energy ecosystem is the continuous penetration of renewable energy. Future energy systems will aim to operate primarily with renewable energy sources and with the capability of managing distributed energy production from ECs (*e.g.*, energy generated by photovoltaic installations in household) [62, 65]. Thus, IoE architectures must be designed to operate partially or completely with green energy sources [15].

- *Flexibility and adaptability in energy management:* Future energy systems will demand flexibility and adaptability in operation and pricing. Regarding the operation, IoE architectures require that modifications of functionalities or incorporation of new features be quickly introduced and deployed throughout the entire power grid. As for tariff systems, the active participation of ECs and adaptive consumption leads to new dynamic and variable payment schemes for energy use (*e.g.*, payment incentives, or penalties), which can be modified in time, even in real-time, according to generation conditions and agreements/negotiations established between parties [63].

- *Regulation and standardization:* The progression towards smart and efficient energy systems involves a discussion on regulation and standardization of operational and economic aspects of both the energy and ICT sectors, to guarantee a consistent evolution of the energy ecosystem and the correct interoperability of the involved

> stakeholders. For the successful development of future energy systems, a standardization of protocols and interfaces is needed, which can be based on existing standards, or might require the creation of new platforms or procedures [62].

In the literature there exist several architectural candidates for the IoE. An example is shown in [3], where Huang *et al.* propose an architecture for using distributed renewable energy and distributed energy storage devices at the residential and industrial levels. In this approach, produced green energy is integrated into the power grid to meet power demands. The proposed system considers the integration and participation of consumers (devices) through a simplified communication network, that is composed of energy routers integrated into the power grid to manage the energy flows. The proposal promotes the need for a communications infrastructure, but it does not provide detailed information about communication protocols/technologies nor energy management strategies, and the analysis is mostly done from the supplier side.

Regarding the use of advanced ICT solutions, some research works analyze the potential of different technologies such as IoT, NFV or SDN integrated into energy systems. For example, in [66], the authors present a substation network architecture enabled by SDN that provides simplified management and reliable communication between the intelligent electronic devices used to monitor the state of the electricity infrastructure. The authors also analyze the virtualization of some components of the power grid and the incorporation of ICT infrastructure to deploy improved management mechanisms. A discussion of some IoT and NFV-based approaches for energy management is presented in Section 3.2.1.1 and Section 3.2.1.2, respectively.

3.2.1.1 IoT-Based Energy Management Approaches

Many studies have analyzed energy efficiency in IoT systems from different points of view, but the integration of IoT infrastructures in the operation of energy systems is still under development and it is a hot research topic at this time [12]. Moreover, the literature reveals that IoT technologies are seen as essential enablers in modern energy systems (such as the IoE approaches) by offering improvements in monitoring, control, management, and automation processes [62]. In this context, several research works have been proposed to encourage the use of ICT-based architectures, mechanisms, and strategies in energy systems. For instance, in [12], the authors survey the features, specifications, communications interfaces, and challenges in the design and deployment of IoT-based systems for energy management purposes in different application environments, such as smart homes, smart power grids, and smart cities.

Other studies have demonstrated that ICT infrastructures, such as IoT and DCs, can be considered as potential enablers for the development of DR programs. For example, in [67], Wei *et al.* propose an IoT-based common information model and communication framework with existing ICT protocols (*e.g.*, Ethernet, IP, and IoT protocols), to deploy an DR energy management system for industrial consumers. The proposal mostly analyzes the operation from the facility side (*i.e.*, from the ECs side), and experimental results demonstrate that the interoperability of entities in industrial facilities, enabled by

ICT systems, allows for the rapid and low-cost implementation of an integrated management system for controlling electrical loads depending on the generation sources, which not only produces improvements in energy efficiency, but also a reduction in the energy cost for the consumer side.

Because demand-side management is an important concern in energy systems and because IoT infrastructures and technologies are already deployed in homes as part of automated systems, DR schemes have been proposed for these kinds of environments [65]. In [64], the authors present an intelligent home energy management system, which uses sensors, actuators, smart meters, and devices connected through a wireless local area network using standard communications protocols *e.g., Ethernet, IP, TCP*. The system integrates local renewable-energy production (from photovoltaic panels) and includes a central hub for monitoring energy consumption and executing the DR strategies (there is no details about the implementation) for controlling of loads. The results show that the integration of IoT technologies and renewable energy in the housing sector optimizes energy performance but is also a sustainable practice to reduce carbon emission.

Regarding algorithmic strategies for adaptive energy consumption (e.g., in HEMS), existing research works have mainly focused on PAR reduction (by reshaping the demand profile) [68], user utility maximization [69], consumption cost minimization [70], and incorporation of renewable energy [71]. Energy management problems are modeled through optimization techniques such as ILP [72], or other approaches, such as game theory [68]. Offline [73] and online [74] algorithms are used to solve these models. Also, heuristic solutions for rapid convergence and simple steps are proposed to reach efficient energy management [71]. For instance, in [72], the authors propose an ILP model that allows maximizing consumer utility (or minimizing energy cost) by adjusting the hourly load level of a given consumer in response to hourly electricity prices. The approach shown in [72] is centralized, and all the decisions are taken entirely by the ES. Due to the complexity of centralized schemes (especially if they produce optimal solutions), distributed models are also proposed. In [68], for instance, the authors propose a demand-side management algorithm using a game-theoretical approach in which each user (player) tries to minimize their consumption.

Due to the complexity of optimal algorithmic strategies for adaptive energy management, identified as non-polynomial, some researchers focus exclusively on heuristic approaches. In [70], for instance, Chavali *et al.* propose an approximate greedy algorithm, in which each EC schedules the consumption of appliances in response to varying electricity prices. The optimization model in [70] is based on minimizing cost functions for each EC. These functions consider the constraints of the appliances and user preferences in the starting consumption time. The results in [70] show that efficient load scheduling results in lower cost for the ECS and the ES, and reduced PAR and load fluctuations. In [73], instead, the authors propose a strategy that computes load scheduling considering photovoltaic availability. The optimization problem is targeted at minimizing the cost of energy and time-based discomfort. Also, an inclining block rate scheme (i.e., a higher rate for each incremental block of consumption) is incorporated into the model to reduce the PAR.

3.2.1.2 Use of NFV Technology for Energy Management

NFV technology has been shown to be an effective platform for deploying management applications, optimization models (mainly based on heuristic approaches), and network services to meet the diverse requirements of customers and vertical markets [75]. Based on network parameters (e.g., traffic load, energy consumption estimations, or active users), the NFV architectural framework can be set to perform actions such as optimized routing of traffic flows, activation of devices, and allocation of resources (both physical and virtual) to achieve desired performance metrics or functionalities (e.g., low-latency requirements [76]). Regarding energy consumption and management in the NFV realm, many studies have focused on the energy-aware operation of VNFs and SFCs, such as in [77] and [78]. Other studies have explored the potential of NFV for resource and energy management in IoT-enabled environments outside the NFV infrastructure. For instance, in [79], Wantamanee *et al.* present an NFV framework that executes an application for real-time synchronization of machine-to-machine sensors nodes, enabling the deployment of a building energy management system.

In mobile communications landscape, NFV-based solutions for energy management have also been analyzed. In these cases, the authors exploit the virtualization and management capabilities of NFV to minimize the energy footprint in different portions of the 5G network infrastructure (i.e., access, transport, and core networks). To achieve energy efficiency, the authors propose linear programming models and optimization algorithms to improve resource utilization in terms of both cost and performance, as shown in [80]. To deal with complexity issues and for real-time applications, the authors also present heuristic strategies (e.g., genetic-based algorithms) as indicated in [81]. The energy management applicability for particular 5G use cases also has been investigated. For instance, in [16], the authors propose an NFV-enabled energy management scheme for a drone fleet with 5G connectivity based on an optimal scheduling algorithm that aims to ensure a given level of service availability.

Considering that NFV-based management policies can be implemented into DCs (cloud computing infrastructures) belonging to the energy utilities [82], the evolution of NFV-based schemes for energy management seems to be a natural process in smart grids. Initial studies demonstrate that NFV technology can be used to virtualize components of power grids (e.g., advanced metering infrastructure), as shown in [83], producing better performance in exchanging information on energy production and consumption. Recent works, instead, analyze the use of NFV and ICT technologies to improve communications among the components of smart grids [84]. For instance, in [85], Yang *et al.* present an NFV/SDN-based model that slices the resources in core networks and coordinates the activation of SFCs to meet end-to-end low latency requirements for mission-critical energy services. Meanwhile, in [86], the authors present an SDN/NFV-based infrastructure that offers optimal placement and dynamic resource allocation of middleboxes, enabling them to meet both cyber-security and low-latency communication requirements in smart grids. The works reviewed in this section demonstrate that NFV is a suitable environment for deploying management solutions in different domains (e.g., 5G networks and power grids).

3.2.2 Contribution, Features, and Differentiators of Our Proposal

Our proposal for adaptive energy management based on NFV and IoT technologies is a feasible alternative and is aligned with the requirements of future energy systems [62]. Even our proposal could be considered as an architecture (solution) on the road to the IoE. This is because of the following features: (i) use of advanced ICT infrastructures (IoT, SND, and NFV) in the energy management process (algorithmic management strategies deployed at NFV domain); (ii) customer-side participation and use of renewable energy sources (the proposed energy management model considers the parameters of IoT-enabled ECs and prioritizes the use of green energy); and (iii) adaptive energy consumption adjusted to the generation (achieved by diverse management mechanisms in the algorithmic strategies). Also, the proposed algorithmic strategies can be applied to scenarios with thousands or hundreds of thousands of energy demands in contrast to most of the existing approaches which, are limited to small-scale scenarios (e.g., in HEMS with up to 10 demands [68] or at most up to 100 demands [70] if heuristic methods are used).

Most studies address energy management and efficiency by minimizing consumption or encouraging energy savings, as shown in [77]. Our proposal instead leverages the dynamic, programmable, and scalable features offered by ICT technologies (such as the NFV technology) to deploy an adaptive energy management solution conditioned on availability (whether renewable or not) and carried out through algorithmic strategies (using optimal but mainly heuristic approaches). The proposed solution also exploits the manageability of the ECs enabled through massive connectivity and IoT technologies for energy management. In this regard, considering that the IoT already connects billions of devices and keeps growing exponentially (e.g., 28.5 billion IoT devices estimated in 2022 [10]), the proposed model and strategies in this thesis could potentially be applied to manage a plethora of IoT-enabled ECs.

To efficiently use the available energy, the proposed solution (architecture) establishes a collaborative energy management environment between the ECs and the ES that is carried out using advanced 5G technologies (NFV and SDN) and aims to adapt consumption according to generation. Unlike existing management approaches, in our proposal, the ECs (devices or services with connectivity capacity and manageable), traditionally seen as an inactive entity, actively participate in negotiating their consumption with the ES. For this, a new consumption model is proposed, in which, before using energy, a two-way handshake is established between the parties. During this process, the ECs send the parameters of services or power demands (*e.g.*, duration, power demanded, priority and initial time) to the ES. Then, this latter using management strategies such as prioritization in energy supply or time-shifting applied to the service execution, send the consumption conditions to the ECs (*i.e.*, the service(s) to be processed and the corresponding execution time).

Our proposed solution (architecture for energy management) is not solely approached from the supplier side, as described in [66], or solely from the customer side, as discussed in [67], or is it exclusively focused on a specific infrastructure (e.g., network resource), as shown in [80], nor is it exclusively focused on the development of algorithmic solutions, as shown in [87]. On the contrary, our proposal presents a complete system for the efficient adaptive management of energy consumption. This proposal presents a complete vision of the architecture, stakeholders, mathematical models related to generation and consumption, and it discusses the complexity associated with the optimal management of the available supply for later present an appropriate architectural framework. In addition, the mathematical model associated with optimal energy use is presented, performance metrics are defined, and an algorithmic solutions (optimal and heuristics) and its numerical evaluation is introduced. The works reviewed in the literature lack the description of any or several of these elements. All information related to our proposal for adaptive energy management constrained to availability is described in detail to motivate future work in this field.

In the proposed solution for adaptive energy consumption, a reliable and scalable communications infrastructure between the ES and the ECs is an essential requirement for efficient energy management. This is vital for realizing the proposal; otherwise, the management of IoT devices (from the view of their activation and consumption) could not be carried out. In addition, the architecture requires robust computational resources on the ES side for the execution of the different management strategies, algorithms, and calculations involved in the optimal utilization of the energy resource. In this context, our proposal uses sophisticated communications technologies such as NFV, SDN and 5G as enablers. Specifically, SDN [30] provides the reliable, dynamic and programmable connectivity necessary for the exchange of information (parameters and consumption conditions) between the ES and the ECs, whereas NFV, deployed in cloud computing infrastructures (*i.e.*, at the DC level), is responsible for the execution of workload scheduling strategies for the adaptation of consumption according to the available energy. In addition, NFV also provides the management entities (management and orchestration functionalities), so all the components of the energy generation and consumption ecosystem (actions and resources between ES and ECs) work in an orchestrated manner [43]. Thus, SDN and NFV are indispensable technologies in the proposed architecture that offer a flexible and scalable ICT infrastructure that can grow proportionally (increase in computing, storage and networking resources) according to the varied requirements of the ECs. These technologies enable efficient, automated, agile, dynamically reconfigurable, and programmable energy management for varied IoT infrastructures.

Chapter 4

NFV/SDN Enabled Architecture for Adaptive Energy Management in IoT Scenarios

This chapter presents our NFV/SDN-enabled architecture proposal for adaptive energy management.

The topics that are covered in this chapter are as follows:

- Overview of the proposed architecture by describing its components and management mechanisms for achieving adaptive energy consumption.
- Consumption model and the complexity of the proposal.
- Description of the proposed architectural framework.
- List of open research challenges and potential application fields.

This chapter is based on:

J2 **Christian Tipantuña** and Xavier Hesselbach. NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy. *IEEE Open Journal of the Communications Society*, 1:357–380, 2020.

C3 **Christian Tipantuña** and Xavier Hesselbach. Demand-Response power Management Strategy Using Time Shifting Capabilities. In ACM Proceedings of the Ninth International Conference on Future Energy Systems, pages 480–485. ACM, 2018.

4.1 Adaptive Energy Management Architecture Proposal

This section presents an overview of an NFV- and IoT-enabled ecosystem for achieving adaptive energy management constrained to availability. Section 4.1.1 describes the entities that compose the energy provisioning and consumption ecosystem. Section 4.1.2 presents the management mechanisms to adapt consumption patterns to availability. Then, Section 4.1.3 presents the mathematical models of generation and consumption sides. Whereas Section 4.1.4 describes the consumption model in the proposed architecture. Section 4.1.5 describes both the computational complexity of the proposal for adaptive energy consumption and the computation capacity needed to deploy the proposal. Finally, Section 4.1.6 presents the proposed architectural framework.

4.1.1 Architecture Proposal Description

The proposed architecture follows the general structure of a DR system and is composed of three entities:(i) an ES, that provides energy from renewable and non-renewable sources; (ii) an NFV-enabled Energy Manager (EM) that is part of the ES and disposes of all ICT infrastructures (e.g., DCs or cloud computing infrastructures) for executing the management strategies (optimal or heuristic algorithms) through diverse management mechanisms (e.g., prioritization of energy supply and workload scheduling using time-shifting capabilities) to adapt consumption (consumption of all services or power demands) to available generation; and (iii) ECs, that represent the end users (*i.e.*, IoT infrastructures) that demand energy. Fig. 4.1 shows a pictorial representation of the proposed architecture for adaptive energy management, and its entities are detailed below.

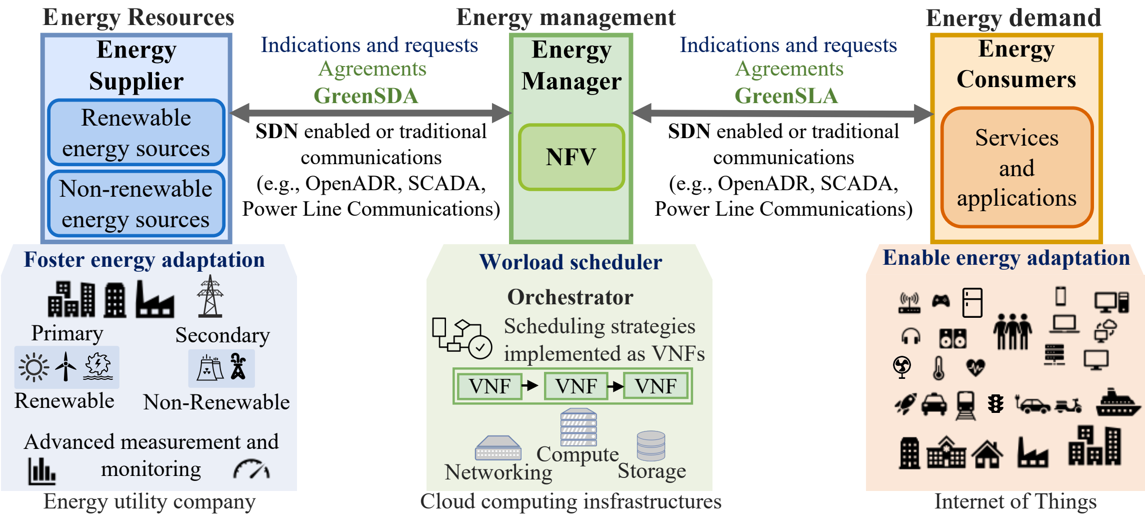


FIGURE 4.1: High-level architectural framework of the proposal for adaptive energy management in IoT-enabled environments.

4.1.1.1 Energy Supplier (ES)

The ES has advanced control, measurement, monitoring, and communications systems, supplies energy (from renewable and nonrenewable sources) to the entire ecosystem and performs energy-mixing process[1]. Different suppliers or sub-suppliers can integrate the ES. However, for analytical simplicity in the proposed energy management model, the ES is regarded as a single entity.

According to the amount of power supplied (P_{ES}) and power demanded (P_D), three different operation states are presented on a periodic basis: normal, shortage and surplus. The *normal operation state* or the regular state refers to a power level in which all the generated power is consumed; i.e., in this state, the power demanded by the ECs is equal to the power supplied by the ES, $P_{ES} = P_D$. In this condition, there is no wasted energy and all services are processed. The *normal operation state* is an ideal scenario; however, the behavior of the ES and the ECs is not flat but it changes over time, which causes shortage or surplus periods. These power levels within the ecosystem are called *shortage operation state* and *surplus operation state*. Thus, a *shortage operation state* represents a scarcity of available power [11], and it can originate from a low supply level (low or zero power generation) or a high demand (demand increase). The ratio between P_{ES} and P_D in a *shortage operation state* can be defined as:

$$0 \leq \frac{P_{ES}}{P_D} < 1 \tag{4.1}$$

During a *shortage operation state* the available power is insufficient to meet all demands (i.e., $P_{ES} < P_D$). By contrast, the *surplus operation state* is defined by an abundance of available power (finite power level) [11]. This state originates from a high supply (for instance, from renewable energy sources) or from a low demand (demand decrease). The ratio between P_{ES} and P_D in a *surplus operation state* is given by:

$$\frac{P_{ES}}{P_D} > 1 \tag{4.2}$$

In a *surplus operation state*, the ES fosters energy consumption, because the available power is greater than the power demanded (i.e., $P_{ES} > P_D$). If after processing all the demands, the system still has energy (i.e., $(P_{ES} - P_D) > 0$), this amount of energy can be stored in battery units for later use. Periodically, changes in generation and consumption cause a transition between the different energy states. This transition goes from a *shortage operation state* to a *surplus operation state* and vice versa, always going through the *normal operation state*, as illustrated in Fig. 4.2.

[1] Process to obtain energy for direct use, combining different primary energy sources [88].

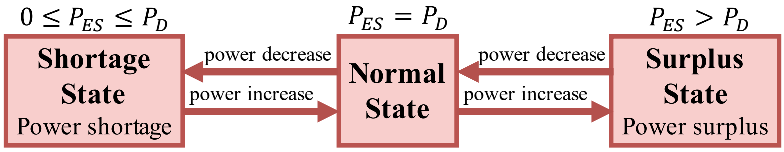


FIGURE 4.2: Interaction between energy operation states.

4.1.1.2 Energy Consumers (ECs)

The ECs are the IoT infrastructures that demand energy to execute tasks, jobs, applications or processes. They have computational resources and are equipped with communications (e.g., SDN-compatible connections), energy (power grid connections), control (integrated or external systems for activation/deactivation of consumption), and even measurement interfaces. Different communications protocols or interfaces (*e.g.*, Ethernet, IP, TCP, SDN, and IoT protocols) can be used by the ECs to exchange energy-management data with the ES (demand-side management). In the energy management process, the ECs are aware and tolerant of the configuration performed by the ES—specifically in the Energy Manager (EM)—to optimize the consumption of available supply. Then, the ECs can activate, deactivate, or modify their energy consumption (e.g., increase or decrease energy use within the minimum and maximum thresholds) based on the conditions established by the ES (workload scheduling). The interaction between the ES and the ECs may be fully automated, or it may include end-user participation, depending on the applicability environment (e.g., HEMS, industrial facilities, or public infrastructures).

The use of IoT technology in the proposed management solution is an ideal alternative because it allows for the use of existing protocols, interfaces, and frameworks (e.g., Ethernet, IP, TCP, SDN, and IoT protocols) used in the exchange of energy-related data (collected from ECs) and for the control of energy resources needed in efficient energy management [15, 64]. In this regard, an environment in which all devices (services and applications) have a communications interface to exchange information (consumption and related parameters) and interact with the rest of the architecture is a completely feasible scenario and not very distant because a growing number of devices are manufactured with embedded communication systems, especially with the proliferation of massive connectivity driven by technologies such as 5G [89]. Moreover, today, there are very affordable platforms (e.g., Arduino or Raspberry platforms) that can be integrated into any device to offer connectivity, management, and control capabilities, converting a traditional device into a smart device (e.g., smart dishwasher).

4.1.1.3 Energy Manager (EM)

The EM is part of the energy utility, and it corresponds to the ICT infrastructure (e.g., cloud computing facilities) in which the NFV technology is deployed. It provides the control, management, and orchestration functions of energy resources and demands. The NFV paradigm gives the EM: (i) reconfigurable behavior by activating or creating VNFs (or complete SFCs) according to the algorithmic management strategy to be implemented and the applicability environment; (ii) scalability, due to the on-demand use of computational capacity; and (iii) the MANO entities (NFV Orchestrator, VNF manager, and Virtualized Infrastructure Manager) so that ES, EC, and the underlying communications systems (*e.g.*, an SDN network) work in an orchestrated manner. These features create a robust energy management solution that can be applied to different scenarios (*e.g.*, HEMs or smart cities).

The EM enables two-way interaction between the ES and the ECs, governs the actions of the two subsystems, ES-EM and EM-ECs, simultaneously. It is responsible for adapting the energy demands to the capacities of the ES (e.g., normal, surplus, or shortage energy states). The cooperation between parties (i.e., ES and ECs) can be supported by contracts or agreements in which technical and economic terms are defined [11]. Technically, through a handshake process, the ECs notify the ES (technically to the EM) of their demands. Then, with the generation and consumption information, the EM runs the management algorithmic strategies (through mechanisms such as time-shifting implemented as VNFs in the NFV domain) to determine the consumption conditions. Specifically, the EM delivers the ECs an optimal (or suboptimal) power scheduling scheme that enables them to adapt the consumption patterns to available supply (i.e., the EM performs adaptive energy management). Thus, the EM can be considered the brain of the architecture, because its operation decides how and when the available power is used. Different communications systems can provide scalable, secure, and reliable connectivity to exchange data on energy management (e.g., parameters of demands and instructions of consumption) between the ES and the ECs. However, SDN technology is one of the best alternatives due to its compatibility with NFV and IoT [90].

4.1.2 Management Mechanisms for Achieving Adaptive Consumption

This section presents different management mechanisms that enable adaptive energy management when incorporated in the algorithmic strategies deployed in the EM (i.e., in the NFV domain). Fig. 4.3 shows an example of the application of these mechanisms, and their description is presented below.

4.1.2.1 Processing of Energy Demands without Management Mechanisms

If the available energy is sufficient to meet all consumption from ECs, which occurs in the normal or surplus power states (i.e., if $P_{ES} \geq P_D$), all energy demands can be processed in their required execution time without being affected by any management mechanism (e.g., displacement in execution due to the application of a time-shifting interval).

4.1.2.2 Use of Time-shifting Capabilities

The time-shifting denoted as u (or Ts) is the finite displacement (forward or backward) on the execution time of an energy demand [14]. This mechanism allows the ES (through

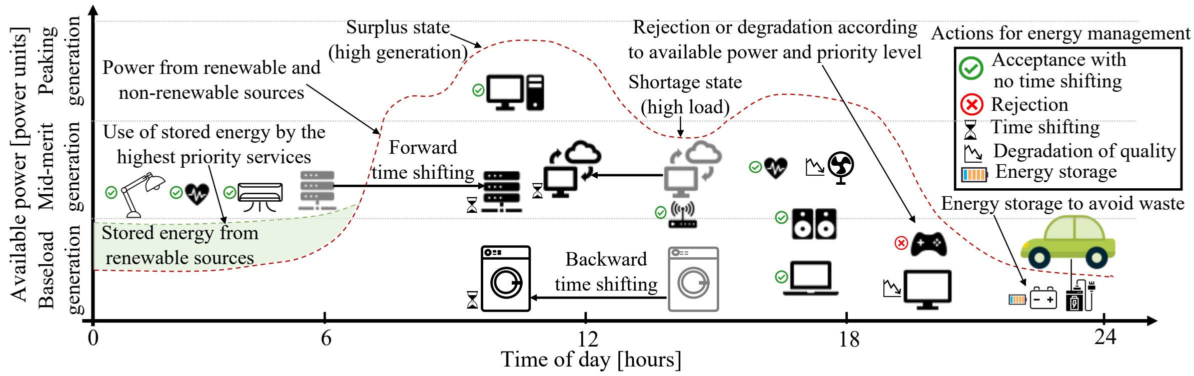


FIGURE 4.3: Example of application of the management strategies for efficient energy consumption [9].

the EM) to increase or decrease consumption by anticipating or delaying the execution of energy demands during periods of energy surplus or energy shortage, respectively. The ES can encourage the ECs to advance or delay their demands by offering a set of reduced energy tariffs depending on the time-shifting performed.

4.1.2.3 Prioritization of Energy Supply

The ES (technically the EM) can use a prioritization scheme to differentiate the energy resource allocation and the application of management mechanisms such as time-shifting and rejection (no energy allocation) on energy demands. The number of priority levels, the priority level for each energy demand, the actions per each priority, and the limits of each management mechanism (e.g., maximum time-shifting interval) are agreed upon between the EM and the ECs (through contracts) [9].

Considering that energy demands correspond to IoT devices' consumption and that these IoT infrastructures offer a service for end-users, hereinafter, the energy demands will be referred to as "services" in the proposed management model. In this regard, the ECs can produce services with multiple priority levels (a service with a single level of priority). According to their priority levels, these services can be categorized into Critical Services (CS) or Non-critical Services (NCS), as described below.

1. *Critical Services (CS):* The CS have the highest priority level, denoted as $j = 1$, with $j \in \{1, \ldots, L\}$, in the proposed energy management model. CS cannot be interrupted or shifted to earlier or later periods, and they cannot be rejected. The ES prioritizes the allocation of the energy resource available to CS. The services in emergency scenarios (e.g., life support devices) and natural disasters (e.g., for search and rescue operations) are examples of CS.

2. *Non-critical services (NCS):* The NCS are shiftable (advanced or delayed execution). However, once operation begins, they cannot be interrupted until the operation completes. Multiple priority levels can be used for the NCS (e.g.,

$j \in \{2, \ldots, L\}$). Thus, after energy allocation for CS, the remaining supply is distributed to the NCS based on their priority level (from $j = 2$ up to $j = L$). The use of different priority levels for NCS seeks the best energy utilization and optimal comfort for the ECS. Examples of services (IoT infrastructures and applications) within this category include entertainment services (non-essential audio and video systems), dishwashers, washing machines, water pumps, water, heaters, and fans.

4.1.2.4 Rejection of Energy Demands

Services are rejected when the available supply is insufficient to cover all demand (due to energy shortage or high load) or when the service(s) cannot be adapted to the energy profile even if time-shifting is used [9]. The rejection criteria are based on the priority level of NCS. Thus, services with a lower priority level (e.g., $l = L$) during periods of energy scarcity are more likely to be rejected.

If all NCS have equal priority, the algorithmic strategies executed by the EM allocate available supply to services whose execution maximizes energy utilization (i.e., minimizes energy waste). Also, the algorithmic strategies are responsible for finding the best management mechanism for each energy demand (i.e., optimal service scheduling) so that executing simultaneous services yields optimal energy consumption.

4.1.2.5 Other Strategies

There are different actions on energy demands that can be incorporated into the proposed architecture and complement the operation of the aforementioned mechanisms. Examples of these mechanisms are consumption variation (e.g., degradation of quality) and energy storage. Regarding the former, in order to adapt P_D to the P_{ES} in both surplus or shortage operation states, a possible strategy, that enables the P_{ES} is not wasted and a greater amount of services can be executed is the variation of power consumption proportional to the P_{ES} value (*e.g.*, a decrease in the brightness level screens of devices, when $P_{ES} < P_D$). The candidates for performing these strategies are the NCS and the ranges of variation are conditioned to the features of each device and the agreements between the ES and the ECs.

Regarding energy storage, although the proposal mainly focuses on the optimal use of available energy and not on its storage, the architecture can leverage the energy storage infrastructure (*i.e.*, the battery units) that is part of the renewable-energy generation, for storing the surplus energy that, if not used, would potentially be wasted. Then, the stored energy could be used to partially or totally meet the demands from ECs. In this context, an important aspect to be addressed in future work is the sizing of storage devices, to ensure, for a finite period of time, the execution of CS, if P_{ES} is insufficient to meet all these demands.

4.1.2.6 Promotion of the Use of Renewable Energy

The architecture encourages the use of renewable energy as a primary source and allows its gradual contribution in the total energy supplied. In this regard, the architecture has an adaptive consumption capacity conditioned to P_{ES}, which is obtained by exploiting the time-shifting capabilities of the services and the optimal service scheduling performed by the ES (EM). In this way, consumption can be adapted to the time intervals where there is an excess of renewable energy. In addition, renewable energy can be stored in battery units and subsequently used for the execution of services, mainly for CS, as shown in the example of Fig. 4.3.

4.1.3 Mathematical Models of Energy Generation and Consumption

This section presents the mathematical representation of the ES and the ECs, considering green energy and management mechanisms in Section 4.1.2.

4.1.3.1 Energy Supplier Modeling

In the proposed energy management model, the ES is characterized by a total power supply capacity denoted as P_{ES}. It is *de facto* the power received at the point of consumption (regardless of losses). The P_{ES} has an initial time defined as $T_{init}^{P_{ES}}$ and a finite duration denoted as m. It is equal to the sum of power coming from renewable (e.g., solar, wind, or hydroelectric) and non-renewable (e.g., coal or natural gas) energy sources, defined as P_R and P_{NR}. Considering that ES can prioritize green energy use, a weight $w_R \in [0, 1]$ is included in the P_{ES} to control the provisioning capacity from non-renewable sources. For sustainable reasons and as a requirement for future energy systems [12], the participation of the P_{NR} in the P_{ES} is expected to be minimal and in the best scenario equal to zero (i.e., when $w_R = 1$). The mathematical expressions that represent the P_{ES}, the P_R, and the P_{NR}, are shown in Eq. 4.3, Eq. 4.4, and Eq. 4.5, respectively.

$$P_{ES} = P_R + P_{NR} \tag{4.3}$$

$$P_R = P_{Es} \cdot w_R \tag{4.4}$$

$$P_{NR} = P_{Es} \cdot (1 - w_R) \tag{4.5}$$

4.1.3.2 Energy Consumers Modeling

In the proposed energy management model, the ECs are characterized by their consumption capacity and the management mechanisms applied to the energy demands. Different ECs with different demands can be considered; however, for analytical simplicity, the model considers only an EC that can produce several services with different consumption parameters. The number of services belonging to an EC can range from units to thousands and hundreds of thousands of services (corresponding to small-and large-scale IoT scenarios). In this regard, the EM must determine the appropriate strategy (optimal or heuristic) to be applied in each scenario.

The consumption model considers a total of N energy demands or services. Each service i, with $i \in \{1, \ldots, N\}$ is identified as S_i. It has an initial time, an interruptible duration, and a fixed power level denoted as t_i, d_i, and p_i, respectively. The power demanded or consumed by all services for which the ES (technically the EM) can allocate energy is denoted as P_D. This value represents the total power consumption. Also, each service S_i has a priority level identified as q_i that can be affected by a time-shifting value (in the case of the NCS) denoted as u_i. The time-shifting can be backward (i.e., when $t_i - u_i$) for the anticipated execution of a service S_i, equal to zero (i.e., $u_i = 0$ or not time-shifting applied) for the normal processing of a service S_i, or forward (i.e., when $t_i + u_i$) for the delayed execution of a service S_i. Normally a service S_i cannot only use a specific value of u_i but move in an interval $\{t_i - u_i, \ldots, t_i, \ldots, t_i + u_i\}$. Table 6.2 summarizes the parameters related to the services. An example of these parameters is illustrated in Fig. 4.4.

TABLE 4.1: Parameters of services or energy demands in the proposed adaptive energy management model.

Parameter	**Description**	**Unit/Comment**
N	Total number of services	Integer number
i	Service identifier	$i \in \{1, \ldots, N\}$
L	Number of priority levels of services	Integer number
j	Priority level identifier	$j \in \{1, \ldots, L\}$
t_i	Starting time of service S_i	Time units
d_i	Duration of service S_i	Time units
p_i	Power demanded of service S_i	Power units
q_i	Priority level of service S_i	Integer number
u_i	Time shifting value of service S_i	Time units

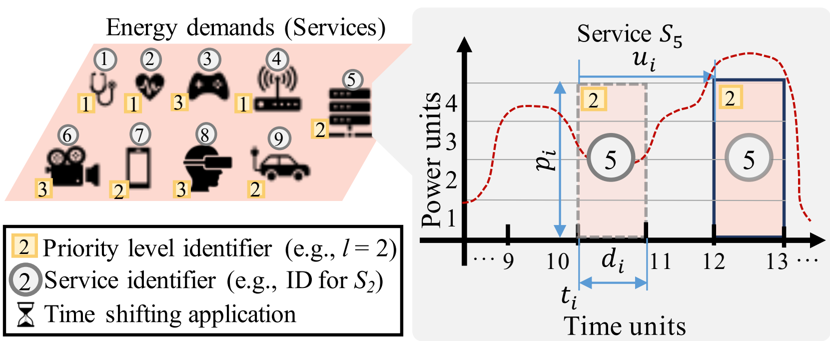


FIGURE 4.4: Graphical representation of a service with time-shifting application. Parameters of S_5: $N = 9$, $L = 3$, t_5 =10, d_5 =1, p_5 =3, q_5 =2, u_5 =+2 (forward).

4.1.4 Consumption Model in the Proposed Architecture

In the proposed architecture, the implementation of management mechanisms is complemented by a consumption model that requires the active participation of the ECs. This section describes the energy negotiation and adaptation between the ES and ECs to use the available power efficiently.

4.1.4.1 Description of the Energy Consumption Model

In contrast to traditional energy systems in which the user is not aware of consumption and immediately uses energy through the activation of services, our proposal establishes a two-way handshake between the ECs and the ES prior to the use of energy. This procedure can be summarized in the following steps: (i) the ECs (devices with connectivity capacity and manageable) send the information about the services (e.g., priority level, duration, power level, possibility or percentage of rejection) to the ES, through a communication network (e.g., SDN); (ii) the ES (specifically the EM) with the information about P_{ES} and the services calculates (through the execution of algorithms) the optimal/efficient scheduling of the services, using management strategies deployed at NFV domain, which enable the efficient energy utilization (i.e., the maximization in the use of the available energy); (iii) the ES (specifically the EM) sends the consumption parameters (services that can be executed and execution times) to the ECs; (iv) the ECs confirm the consumption conditions and request energy for services to be processed; and (v) the energy is allocated, the devices are activated and the ES supplies the demanded energy (P_D) for service execution. Fig. 4.5 shows a general representation of the bidirectional dialog between ES and ECs to enable efficient energy consumption; instead, Fig. 4.6 presents this level of interaction considering the stakeholders and enabling technologies of the proposed architecture.

4.1.4.2 Energy Adaptation

In the previous section, the interaction between ES and EC is summarized. This section presents a brief description of the energy adaptation process (energy consumption model) considering the participation of the three stakeholders. The actions involved in the energy adaptation process are related to the procedures performed in the two subsystems ES-EM and EM-ECs. Fig. 4.7 shows a summary of actions/processes (handshake) carry out between the three stakeholders to offer an efficient energy management solution. In short, whenever there is an energy demand from the ECs, the following processes/actions are performed:

- The ECs enquire the EM if the system has available energy to execute their services. Because the EM is a management entity and not an energy supplier, it sends this inquiry to the ES. When the EM has the information about the status of the ES, it reports to the ECs the amount of available energy (P_{ES}). At this point, the EM cannot specify to the ECs how many demands are going to be accepted, shifted in

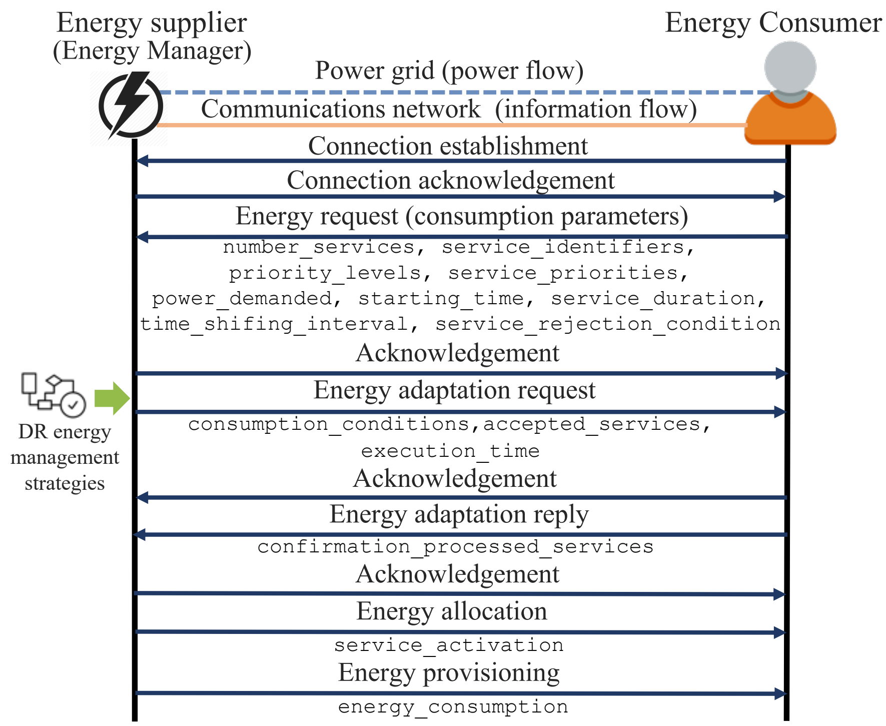


FIGURE 4.5: Summarized handshake process between the ES and the ECs.

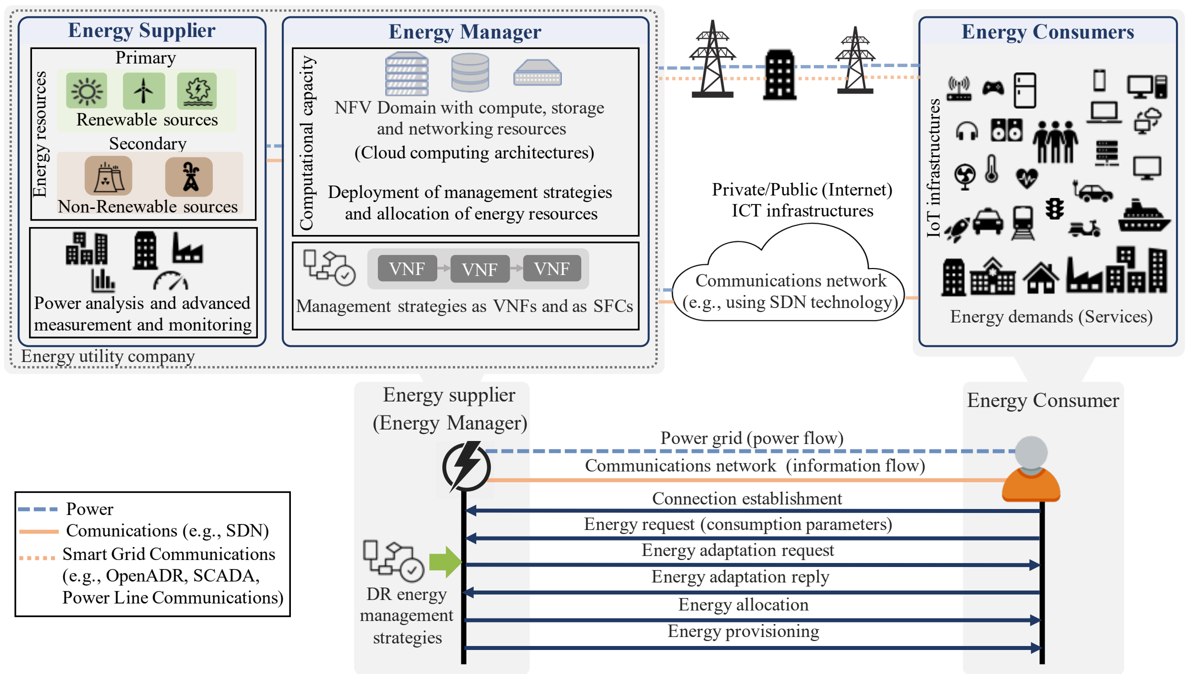


FIGURE 4.6: Overview of an NFV-enabled ecosystem for adaptive energy management, adapted from [9].

time, or rejected because the EM has not yet implemented any strategy (workload scheduling).

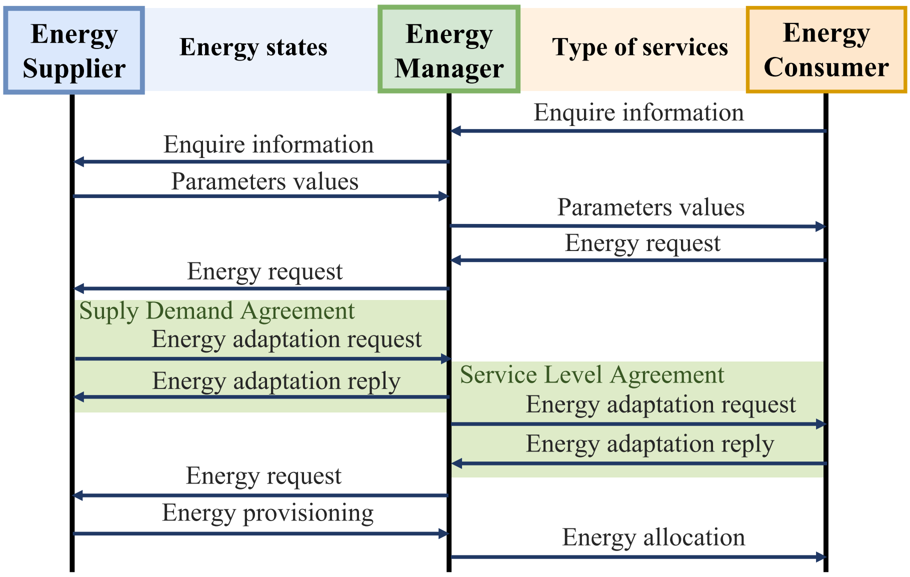


Figure 4.7: General scheme of the energy adaptation process (handshake process between the three stakeholders).

- Subsequently, the ECs formally initiate the energy request by sending the different parameters of the services to be processed (see Table 6.2).

- At this stage, the EM knows the total number of services (N) and the aggregated power demanded (P_D). Thus, the EM proceeds to communicate this requirement to the ES, and, in a bidirectional data flow exchanged between the ES and the EM (GreenSDAs), the ES-EM subsystem chooses the working energy state, which can be either normal, shortage or surplus. Depending on the energy state, the ES can foster energy consumption, deferral in service execution, or service degradation on the basis of actions and indications that the EM gives to the ECs.

- Once the energy state has been established, the EM can use the information about P_{ES}, P_D, and the parameters of the services as input in algorithms (NFV-Based scheduling strategies) to compute the optimal (or near-optimal) energy allocation. This information allows the EM to know which demands are going to be processed and their respective execution times. The workload distribution enables the efficient use of the P_{ES} at each time.

- The scheduling of the demands is communicated to the ECs. Then, thanks to the cooperation established between the EM and the ECs in the subsystem EM-EC (GreenSLAs), the ECs accept the distribution of the tasks performed by the architecture (EM). In this sense, and if the energy is insufficient, the ECs accept that their demands may or may not be processed. As aforementioned, the architecture can prioritize the energy supply for CS execution.

- Finally, on the basis of the information about the services (e.g., services without time-shifting and services with time-shifting), the EM requests from the ES the corresponding amount of energy. Then, the EM sends the activation instructions to the ECs (devices) and performs the allocation of energy resources. Finally, the ES supplies energy to ECs.

4.1.5 Complexity of the Proposal and Related Processing Requirements

Before formally presenting the architectural framework of our proposal, in this section, we discuss the computational complexity and processing requirements related to the efficient management of available power consumption. In addition, we analyze the technology enablers that allow us to meet the architecture requirements to demonstrate that our envisioned energy management solution is entirely achievable with existing technologies.

4.1.5.1 Complexity of the Proposal

In our proposal, a key aspect to efficiently use P_{ES} is the (optimal) workload scheduling by exploiting management mechanisms such as time-shifting capabilities of services. Mathematically the process of selecting the services to be processed (considering those subject to management mechanisms) with the aim of efficiently using a finite P_{ES} is analogous to the objective of the 1/0 Knapsack Problem of placing the most valuable or useful items without overloading the knapsack [91]. In this regard, the literature has proven that this kind of problem has complexity NP-hard [91]. For solving the problem optimally (i.e., optimal energy utilization in the context of our proposal) there are several options, and one of the best-known strategies is the exhaustive search based on the combinatorial analysis of all possible solutions. Translated into the scope of our problem, the exhaustive or brute-force search consists of analyzing all possible combinations of N services considering all possible management mechanisms (e.g., analysis within all time-shifting intervals) to which a service may be subject. In this regard, an optimal algorithmic solution (based on combinatorial analysis) for efficient energy management in the context of DCs has been validated in previous work [14]. Then, the analysis of the algorithmic strategy (optimal or exact) reveals that this method has exponential complexity with a growth rate that depends on the values of services to be processed (i.e., N) and the values of the management mechanisms (e.g., the value or interval of u^i (Ts)). Fig. 4.8 shows an evaluation example of grow rate of the problem (optimal service scheduling) according to the increase of the number of processed services (energy demands).

The example in Fig. 4.8 indicates that an increase in N, u_i, or both lead to an increase in the size of the problem (*i.e.,* in the size of the search space to find the optimal solution), which can potentially demand a greater amount of computational resources and runtime. For example, with $N = 9$ and $u_i = 3$ the number of combinations to be processed is over 40 million. For a computer equipped with a 3.33 GHz x 12 cores Intel Core i7 Extreme processor and 12 GB RAM, the total running time needed to obtain the optimal solution (optimal workload scheduling) is about 90 hours, which can be an expensive computing time especially considering the density of current IoT deployments and the low latency required by modern networks [92].

In summary, the problem of the optimal management of available energy consumption presents a hardness NP-hard and its optimal solution has an exponential complexity. This information reveals the drawbacks and needs of the proposed architecture, in terms of computational resources (processing and memory) for calculations and the development of faster algorithmic solutions (e.g. based on heuristic methods). The following discusses the requirements and enabling technologies to address computational complexity related to the execution of management strategies.

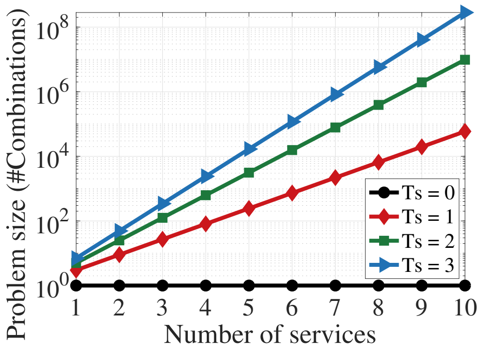


FIGURE 4.8: Example of growth rate associated with the problem of optimal scheduling of services. Growth rate based on N and $u_i(Ts)$.
Parameters: $N = \{ 1, \cdots, 10 \}, max \{ u_i = Ts \} = \{0, \cdots, 3\}$.

4.1.5.2 Processing Requirements

Based on the analysis in Section 4.1.5.1, we highlight two relevant aspects that must be considered in the deployment of the architecture: (i) establishment of maximum values of management mechanisms (e.g., maximum values of u_i) in the agreements between the ECs and the ES, and (ii) use of sophisticated computational resources to execute management strategies. Regarding the first point, although we do not address the contractual terms between ES and ECs, an important parameter that must be considered is the value or ranges of management mechanisms (e.g., the maximum time windows of services) because, as demonstrated in the growth rate in Fig. 4.8 an indiscriminate increase or selection of the parameters has a direct impact on the complexity. As for the second point, the complexity related to calculations for efficient energy utilization reveals the need for a sophisticated infrastructure both for the processing of services (high computational capabilities) and for the interaction of the components in the architecture (efficient network infrastructure) especially for delay-sensitive applications.

In addition, the fact that several algorithmic solutions of different complexity and characteristics (optimal or heuristics) can be executed to meet the varied requirements of services and scenarios, demands that the proposed architecture (specifically in the EM) has reconfigurability and/or programmability capabilities, as well as being agile enough to execute changes on the fly without affecting or degrading the performance of any stakeholder involved in energy management. Related to this requirement, an important aspect is the scalability to cover the dynamic increase/decrease of ECs; so, the architecture must have the capacity to be deployed using hierarchical or distributed infrastructures according to the requirements of the ECs. Other relevant requirements in the proposal are the abstractions between the components through a separation in layers or domains (to promote the transparent evolution of ES, EM, and ECs) and the unified management of the whole architecture.

In summary, the architecture for an efficient use of P_{ES} presents the following requirements:(i) high performance computing infrastructures for management calculations, considering P_{ES} and P_D information, (ii) reliable, flexible and scalable communications network to exchange the management information and instructions between the ES and ECs, (iii) agile deployment of management strategies and reconfigurability and programmability capabilities to implement diverse algorithmic solutions, and (iiii) unified orchestration (coordination and interaction) of all stakeholders to ensure an efficient energy management in scenarios of varied sizes (from residential users to clients that may be companies, populations, smart cities or even countries) and features.

4.1.5.3 Enabling Technologies

The structure of the proposed architecture, as shown in Fig. 4.1 allows for the adoption of different technological enablers for each stakeholder. Existing protocols, interfaces, and mechanisms developed for power grids, DR systems, and IoT infrastructures can be used and adapted to the context of the proposal. For example, the communications network necessary for the exchange of indications, and instructions between ES and ECs could be facilitated by technologies such as OpendADR [93], Supervisory Control And Data Acquisition (SCADA), or Power Line Communications (PLC) [15]. In the proposed energy management solution, however, we have decided to use NFV and SDN technologies to carry out the proposal, because they perfectly meet the functional requirements described in Section 4.1.5.2. In addition, NFV and SDN are already key participants in modern communications systems, such as 5G [89], which facilitates the deployment of the proposed energy management solution. In this context, the architecture presented is open to include new technological enablers as the ICT and energy systems evolve.

In the architecture, NFV deployed at DC level or in general in a cloud computing environment disposes of all the computational resources needed to execute the management strategies (e.g., workload scheduling) and perform all necessary calculations to efficiently adapt the power consumption demanded (P_D) to the availability (P_{ES}). This technology also provides the architecture the management entities so that all components work orchestrated, which is essential to ensure that both the processes of the calculations and the communications and notifications between the stakeholders have a very low latency associated. In the NFV scope, the energy management strategies are represented by VNFs forming SFCs. Then, these VNFs can be created, modified or upgraded, even on the fly, according to the desired functionalities and objectives, which provides programmability and reconfigurability to the proposed architecture. In addition, thanks to NFV technology, the management of the ECS and the underlying network (SDN) is separated (abstracted) from the functionality (i.e., management strategies), a feature that enable the proposal to be applied in different scenarios, such as small households, intelligent transportation systems, and smart cities.

The SDN technology instead provides the architecture reliable, secure, and scalable connectivity necessary for the dynamic interaction and the exchange of information between components (ES-EM-ECs), mainly between the ES and the ECs (parameters and consumption conditions). This technology is also scalable enough to allow the management (connectivity) of both a small and massive number of devices, and agile and flexible enough to allow rapid deployment of indications and changes necessary for energy management. SDN can easily adapt the network infrastructure (underlying network) to the requirements from the EM (NFV realm and management strategies implemented as SFCs).

Regarding the operational and ownership aspects of the ICT systems (i.e., the NFV-enabled EM and the underlying SDN network) needed to carry out the proposal, in the first instance, the ES could lease these infrastructures to ICT and telecom providers, and multiple interoperability and negotiation schemes (contracts) that are outside the scope of this paper could be considered. It is expected, however, that, following the guidelines for future energy systems as described in Section 3.2.1, the energy sector will invest in and incorporate sophisticated communications systems into the power grids in the coming years. A scenario that is very feasible to be achieved, because currently the energy providers/distributors have large operation centers, to monitor and manage energy production and consumption and where NFV can be easily deployed, and also have optical transport networks (e.g., using optical ground wire technology) for data exchange. Then, these infrastructures could serve as a baseline for the deployment of our architecture.

In summary, NFV and SDN are scalable architectures that can grow proportionally (increased use of servers, controllers or switches) according to need and are able to work in distributed environments (e.g., multiple SDN controllers may be used to manage a massive number of ECs), enabling an automated, agile, flexible, scalable, dynamically reconfigurable, and programmable energy management for IoT devices, services, and applications with different requirements. The NFV and SDN technologies can fully interact with power grids networks [94], allowing DR systems to be easily deployable and they can be successfully integrated into a unified architectural framework for the deployment of agile, flexible and scalable services as demonstrated by the European Telecommunications Standards Institute (ETSI) [90]. In this regard, our proposal can be seen as an NFV use case (an energy management optimizer) [43], of rapid deployment that is capable of introducing new features and functionalities in an agile manner as the services and the energy market evolve.

4.1.6 Architectural Framework Proposal

As seen in the previous section the computational complexity of the proposal requires flexible and powerful architecture. Then, the enabling technologies to carry out the efficient management of P_{ES} are NFV and SDN technologies not only because they are inherently complementary, but also because when they work together they can offer great capabilities and benefits in the deployment of applications and services [38, 90]. In our proposal, the NFV/SDN integration adapted to the concept of DR systems gives rise to a robust and sophisticated energy management system that can be used in a wide range of IoT implementations. This solution, as a modular and open ecosystem divided into layers or domains, corresponding to ECs, SDN, NFV and the ES, allows the adoption of technological solutions for each domain (the domains can be developed independently), which can potentially improve the performance of the entire architecture. For example, in our proposal we have considered the possible inclusion of Fog Computing (FC) (see Fig. 4.9). FC is a technology that allows to extend the functionality of cloud computing (processing resources) close to the end user [95]. Thus, in the architecture, FC is seen as an alternative domain to bring the NFV functionality (computations for service scheduling) closer to the ECs. Several studies shown that NFV capabilities placed at the network edge can result in a more efficient network-wide resource utilization, bandwidth, low latency, mobility and heterogeneity [96]. The proposed architecture for efficient energy management is show Fig. 4.1, and its low-level representation including all domains and FC technology is illustrated in Fig. 4.9. The description of the different architectural domains is presented below.

4.1.6.1 Energy Supplier Domain

The ES belongs to this domain and is responsible for feeding the entire ecosystem using renewable and non-renewable energy sources, which have been categorized as primary and secondary sources, respectively. This categorization has been adopted because the proposal has as a collateral objective to promote the majority use of renewable sources because their adoption is envisioned as a promising long-term and environmentally friendly alternative.

4.1.6.2 NFV Domain

NFV, operating at cloud computing level (or alternatively at fog or edge computing levels) and applied within the context of a DR system, is the domain responsible for making decisions/actions to manage the power demands. In the NFV domain, the strategies or algorithms that enable the workload distribution according the availability are implemented through one or more VNFs, as shown in the generic example of Fig. 4.9. These VNFs, running on the NFV Infrastructure (NFVI), i.e., on generic general-purpose servers with processing, storage and networking capabilities, form an SFC and, in turn, a network service, which is intended to optimize the energy consumption. The NFV domain in the scope of the proposal comprises: (i) VNFs that corresponds to the algorithms or subroutines deployed to perform the energy management of services (scheduling strategies); (ii) the NFVI that consists of all hardware and software resources to host and connect VNFs and the entire infrastructure that enable the energy provision, energy management, and connectivity of ECs; and (iii) the Management and Orchestration (MANO) framework that coordinates the necessary resources (resource allocation), tasks, and actions between the three different domains (energy, NFV and SDN) to setup and implement the NFV functionalities.

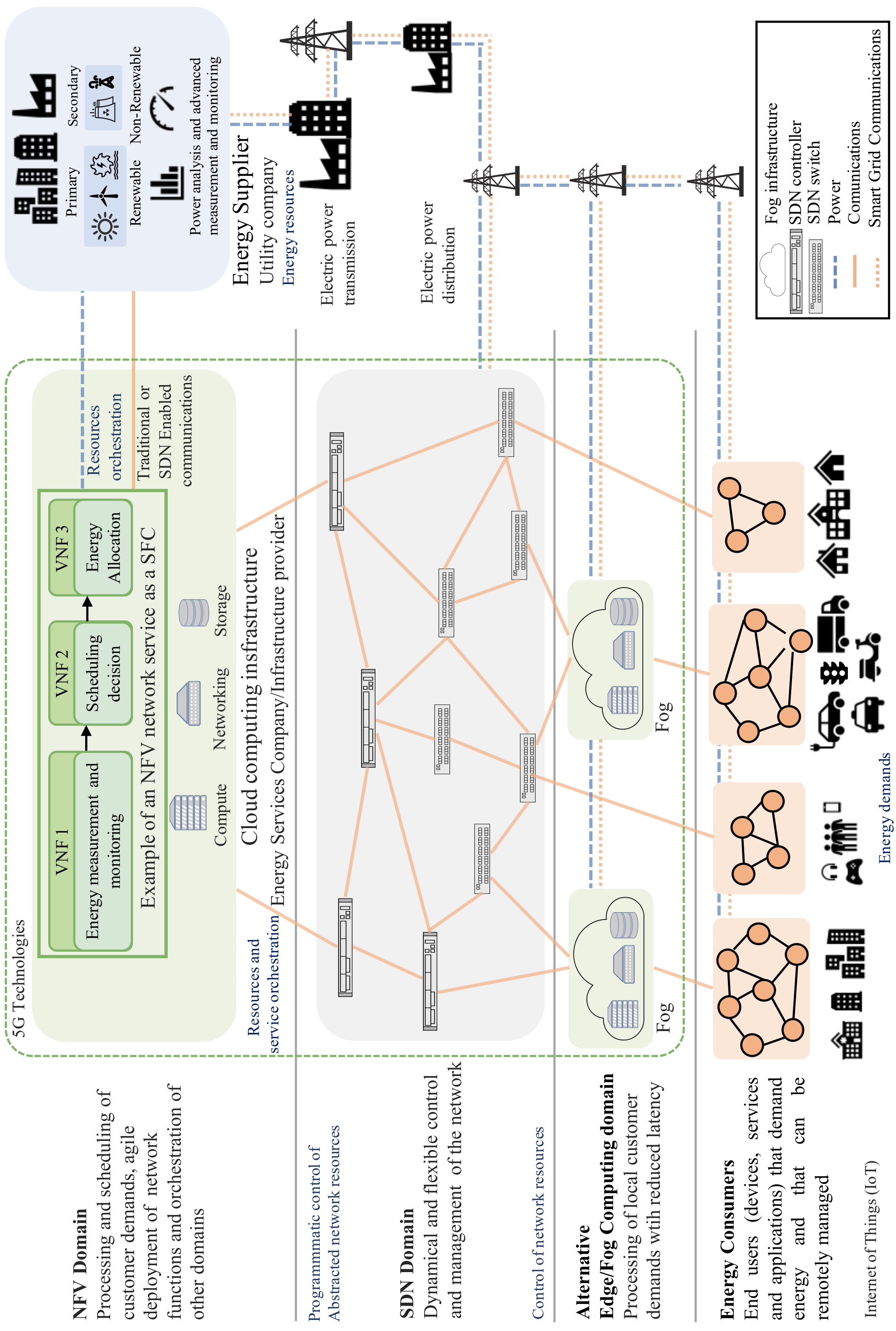


FIGURE 4.9: Low-level schematic representation of the proposed architecture.

From the information of energy provisioning (from ES through the Virtualized Infrastructure Manager (VIM)) and the users demands (from the ECs through the VIM), the NFV Orchestrator (NFVO) determines the allocation of energy resources for those demands that can be processed. The process of resource allocation is made based on calculations and service scheduling algorithms implemented as VNFs and forming SFCs, which are executed in the NFVO and managed by a VNF Manager (VNFM). In this context, the NFVO can decide the number and sequence of activation of the VNFs, it can also determine the use or not of a set of VNFs in order to change the behavior of the energy management on the fly, if necessary. Thus, the NFVO functionality is divided into resource orchestration (NFVI) and service orchestration (VNFs).

The processing of energy demands (from VIM to NFVO) and the allocation of resources (from NFVO to VIM) is constantly carried out, the architecture, therefore, must be able to send the information through its different domains in order to guarantee an efficient management of energy at all times. In summary, once the computation of the energy allocations has been performed, the NFVO uses the VIM entity to notify the ECs (NFVI) of the consumption conditions, i.e., the services that can be processed and their respective execution time. At all times, the energy consumption from the ECs is restricted to the actions performed by the NFVO (energy allocation procedure) and the amount of available energy in the architecture (P_{ES}).

4.1.6.3 SDN Domain

The NFV and Energy domains assume that all necessary connectivity can be dynamically established. To this end, SDN has been considered since it makes it possible to directly program, manage, and orchestrate the network infrastructure (ECs). The SDN paradigm facilitates delivery and operation of notifications and actions/instructions from ES, EM and ECs through the interaction between the NFVO and the VIM entities. In this regard, all changes made by NFV (algorithms executed in the NFVO) are transparently adopted by the underlying infrastructure (ICT infrastructures or power grids infrastructures [66]).

The SDN within the architecture is mainly responsible for two functions: (i) the dynamic connectivity of devices (ECs) and (ii) the energy management tasks. In the first case, the SDN network allows communication between the devices and components of the ecosystem (even the connectivity required for communicating VNFs inside the EM if needed). By means of a controller or group of controllers, which can be managed by the MANO component through the VIM as shown in Fig. 4.9, the SDN-compatible devices (ECs) receive the forwarding (energy consumption) instructions. Thus, the SDN controller interfacing with the device management agents is able to carry out the monitoring, configuration, and control functions of network resources as needed for the provision of data services. Depending on the size of the network (ECs to be managed) the ecosystem can be nested in different levels of controllers and SDN switches. Regarding the second case (the energy flow management), the controller receives the messages from the VIM and notifies the devices to modify their power usage condition, i.e., whether or not to execute their services. In a similar procedure, the controller sends the power requests to the NFVO through the VIM. In this way, the VIM, the SDN controller, and the underlying connectivity infrastructure form a hierarchy for delivering the energy management service throughout the architecture.

Once the architectural framework of the proposal and its respective domains have been described, its most relevant properties are presented below:

- *Stability*: The DR operation enables constant maintenance of the balance between energy provisioning (P_{ES}) and consumption (P_D).
- *Environmentally friendly energy management*: The architecture has been designed to be powered by renewable (P_R) and non-renewable (P_{NR}) energy sources. In a particular case ($w_R = 1$), the system could be powered only by green energy.
- *Scalability and dynamic operation*: The architecture can scale its performance based on the requirements from ECs and the network conditions. The dynamic management of resources (virtual or physical) can also be exploited to achieve additional energy efficiency through consolidation, migration, or the on-demand utilization of resources, for example within the EM.
- *Flexibility and agility*: The architecture allows that the VNFs associated with a SFC can be executed without being linked to a specialized hardware and in cloud, edge or FC infrastructures. Additionally, the independence of software and hardware promotes a suitable space for the integration of solutions (hardware or software) from different manufacturers and providers.
- *Programmability*: The energy management functionalities of the architecture can change on the fly. The EM can decide the execution of a specific SFC (algorithms) to address the needs or requirements of a particular scenario, and all changes are transparently adopted by ECs.
- *Simplified and improved system management*: The management entities of NFV (NFVO, VNFM and VIM) and of SDN (control plane) by working together ensure a consistent control and management to deploy the scheduling strategies and allocation of resources for different IoT devices services, and applications.
- *Open ecosystem*: The proposed solution, based on open architectures (NFV and SDN), can adopt open-source interfaces and solutions developed from different players. For example, the Openflow protocol [30] to provide an SDN southbound interface and the OpenDaylight SDN controller [30] with the OpenStack [50] project as NFVI. The architecture could be even implemented using sophisticated projects, such as Open Source Mano (OSM) [52], or Open Platform for NFV (OPNFV) [53].
- *Critical services guarantees*: The dynamic generation-consumption operation of the architecture enables the distribution/reduction of consumption in order to guarantee the provision of energy for the execution of CS in accordance with the terms agreed between ES and ECs.

4.2 Open Research Challenges of the Proposed Architecture

This section summarizes open research issues intending to motivate future work in this field.

4.2.1 Complexity and Scalability

Our proposed architecture aims to be flexible and scalable enough to be applied from small scenarios with few services (devices), as is the case of domestic environments, to very large scenarios that can cover entire cities. Regarding the algorithmic solution for small-scale scenarios, possible alternatives are the optimal or exact strategies. However, to meet the very low latency requirements of modern networks that are in the order of milliseconds [97] and to deal with the billions of devices that are connected to the Internet [25], it is necessary to develop strategies that require a low running time, demand low computational resources, and produce high-quality solutions. In this regard, there are a number of heuristic and metaheuristic techniques and methods. However, considering that the P_{RES} minimization problem falls into the category of a 1/0 Knapsack Problem and based on the literature reviewed, we indicate the following methods as promising solutions: (i) a prepartitioning strategy based on a divide-and-conquer approach, (ii) a genetic-algorithm-based solution, and (iii) a dynamic-programming-based approach. For the development and evaluation of these strategies, the parameters, mechanisms, and results obtained with the exact method can be taken as a baseline.

4.2.2 Energy Storage Management

The integration into the architecture of an element that stores the energy produced by renewable and non-renewable sources (e.g., battery units) is expected to contribute to better use of energy because this component can act as a buffer to store or deliver energy according to the conditions of generation and consumption (energy states). The battery unit can potentially improve the overall performance of the architecture, but its prime benefit would be to contribute to the execution of CS. Among the different topics that can be addressed in future work are mathematical modeling and sizing, optimal location within the architecture, and coordination with other domains.

Another important aspect to analyze is the management of energy produced and stored by users. Traditionally, energy is used by the owner of the generation system, but in a more ambitious approach, under the coordination of the proposed architecture, this generated/stored energy could be distributed to other users or locations. This process would increase complexity in management, since continuous monitoring and coordination of potential energy consumers/producers (a.k.a., prosumers) would be necessary, but in turn, it would improve the use of all produced energy.

4.2.3 Security and Privacy

The security concerns have been addressed in different studies for SDN [98], NFV [99] and smart grids [100] independently. However, for the proposed architecture, a security framework that cover services such as integrity, authentication, privacy, and availability must be developed, which could be a very challenging task since each domain has different characteristics and requirements. Among the different aspects of security to consider, privacy and data integrity should be highlighted. The architecture must guarantee the anonymization of ECs in order to avoid targeted attacks, especially on CS. With respect to data integrity, the DR system must provide reliable transmission and processing to avoid alteration in control and management information and bill modifications.

4.2.4 Data Analytics

The information about the ECs and the network resources used is constantly sent to the EM. This information can be stored and used not only for billing, but also for monitoring of service quality, network utilization, and for obtaining performance indicators. Also, the stored data can be exploited to discover patterns or predict trends through the use of machine learning techniques, with the objective of performing accurate resource allocations and offering personalized services.

4.3 Potential Application Fields of the Proposal

In this section, we briefly describe potential application scenarios.

4.3.1 Management of the Public Infrastructures Energy Consumption and Provisioning

The proposed architecture can be applied to perform tasks of monitoring, management and control of energy resources in cities, municipalities, neighborhoods, and other locations. In these environments, the architecture can collect energy demands through a centralized or distributed implementation and perform an efficient energy allocation for a specific place, group of services or users according to resource availability. For example, our architecture can interact with ICT infrastructures of municipalities or local governments to manage and control smart power grids, intelligent transport systems, street lighting, controllers and other public infrastructures.

4.3.2 Energy Management of Electric Vehicles

Preliminary studies have demonstrated the application of NFV concepts in the field of electric vehicles [101]. In this regard, our proposal can replace, complement or improve the energy management tasks traditionally performed by traffic controllers or intelligent transportation systems. Through communications systems (mainly based on SDN), the architecture could monitor available energy, charging points, and consumption levels (battery levels). Thus, based on the information gathered, the DR architecture (NFV domain) could indicate to users the periods during the day for the use of vehicles or battery recharging. In addition, information on energy resources and consumption can be used to carry out weekly or monthly planning of energy distribution to avoid shortages or possible collapses of the electrical grid.

4.3.3 Ecosystems Powered by Renewable Energy Sources

The DR architecture has the capacity to efficiently manage the gradual contribution of renewable sources, being able to work entirely with green energies ($w_R = 1$ in Eq. 4.4 and Eq. 4.5) if necessary. It is expected that a greater contribution of renewable energy, mainly from sources such as solar and wind, will demand greater dynamics in the generation-consumption ecosystem in order to take special advantage of periods of surplus. Our approach responds to these requirements and can be envisioned as a promising alternative toward the deployment of high performance zero-emissions ICTs or power grid infrastructures.

4.4 Conclusions

This chapter presents architecture proposal for the adaptive consumption of available energy, whether 100% renewable or not, for IoT infrastructures, and it is envisioned as a candidate for the deployment of the IoE. The proposal covers the description and interaction of the stakeholders (ES, EM, and ECs), several management strategies, including service scheduling using time-shifting capabilities and prioritization of the energy supply, and a consumption model where consumers are an active part of the energy management process.

The complexity analysis has demonstrated that the proposed energy management solution has a hardness NP-hard and requires sophisticated ICT infrastructures for its operation. These requirements in the architecture are met by NFV and SDN technologies. Specifically, all the management strategies and calculation, such as workload scheduling, needed to adapt P_D to P_{ES} are carried out by NFV, whereas all connectivity for data exchange corresponding to instructions and notifications between ES and ECs is provided by SDN. Thus, NFV and SDN, both key participants in 5G, enable automated, agile, flexible, scalable, dynamically reconfigurable, and programmable energy management for IoT implementations with varied requirements.

Chapter 5

Algorithmic Strategies for Adaptive Energy Management

This chapter presents the mathematical model of adaptive energy consumption and the algorithmic strategies to solve this model. The topics that are covered in this chapter are as follows:

- Energy management model for adaptive energy consumption based on an ILP formulation.
- Optimal algorithmic strategy to solve the adaptive energy management model.
- Heuristic algorithmic strategies to solve the adaptive energy management model and to tackle the complexity of the optimal solution.
- Evaluation of the algorithmic strategies, optimal and heuristics, for different energy provisioning and consumption profiles and in various scenarios.

This chapter is based on:

J2 **Christian Tipantuña** and Xavier Hesselbach. NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy. *IEEE Open Journal of the Communications Society*, 1:357–380, 2020.

J5 **Christian Tipantuña** and Xavier Hesselbach. NFV-enabled efficient renewable and non-renewable energy management: Requirements and algorithms. *Future Internet*, 12(10):171, 2020.

J7 **Christian Tipantuña**, Xavier Hesselbach, and Walter Unger. Heuristic Strategies for NFV-Enabled Renewable and Non-renewable Energy Management in the Future IoT World. *IEEE Access*. doi: 10.1109/ACCESS.2021.3110246.

5.1 Energy Management Model

The energy model for adaptive energy consumption is illustrated in Fig. 5.1. This section presents the related mathematical formulation. Section 5.1.1 describes the assumptions considered in the proposed model. Section 5.1.2 presents the ILP formulation for adaptive energy management, while Section 5.1.2 analyzes its complexity.

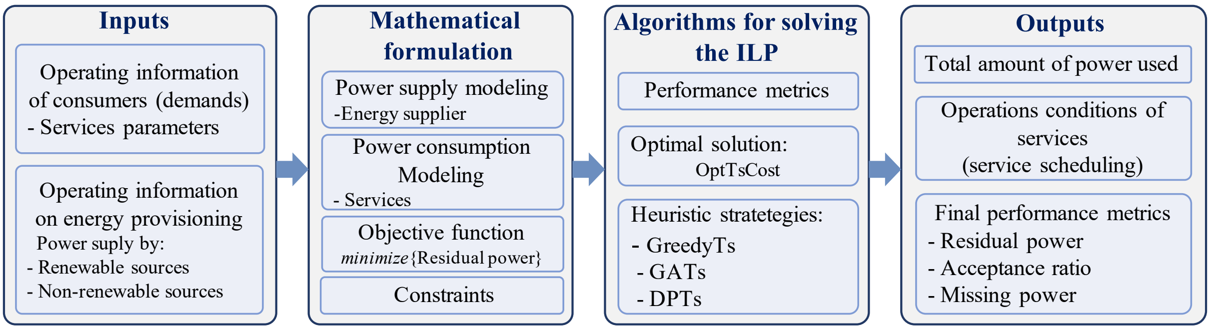


FIGURE 5.1: Schematic of the management model for adaptive energy consumption.

5.1.1 Assumptions Related to the Energy Management Model

To provide a reasonable implementation of the proposed adaptive energy management model, the following assumptions (simplifications) have been considered:

- The use of a discretized time model in which each time slot k has an equal duration within a maximum time horizon denoted as W (i.e., $k \in \{0, \ldots, W\}$). In this time model, the size of time slots can be customized to different time units (e.g., unit of seconds, minutes, or hours) depending on the application scenario.
- The use of integer discrete values for the parameters d_i, p_i, and u_i for each service S_i, with $i \in \{1, \ldots, N\}$. This operation condition allows maintaining the linearity of the energy management model and makes implementing the algorithmic strategies feasible.
- The energy is allocated for the execution of complete services. Fractional service scheduling is not allowed in the proposed algorithmic strategies. In this regard, the partial consumption of services can be addressed in future work.

5.1.2 ILP Problem Formulation for Adaptive Energy Management

5.1.2.1 Prioritization in the Use of Green Energy

Before discussing the objective function to optimize, this section presents the mathematical expression to prioritize green energy use for scenarios in which this process is applicable. Different approaches can be used to promote renewable sources; however, a simple solution is to establish a cost function related to power consumption, as shown in Eq. 5.1. The general cost function in Eq. 5.1, defined as $Cost_{PD}$, comprises individual costs associated with the consumption of renewable and non-renewable power, denoted as $Cost_{P_R}$ and $Cost_{P_{NR}}$, respectively. The ES can tune the $Cost_{P_D}$ by modifying the weights $w1$ and $w2$ in the range [0,1] according to some operating parameters (e.g., P_R available). In this context, to promote the use of green energy, the $Cost_{P_R}$ can be set to a minimum value (i.e., if $w_1 \ll w_2$) or zero (i.e., if $w_1 = 0$ or $Cost_{P_R} = 0$), in such a way that the total cost only depends of the use of P_{NR}.

The $Cost_{P_{NR}}$ can be defined as a value proportional to the amount of the P_{NR} consumed, as shown in Eq. 5.2. In this regard, the task of the ES (technically the EM) is to find the best supply-consumption conditions (e.g., encouraging consumption during a surplus of wind or solar energy) to obtain a minimum value of the $Cost_{P_{NR}}$ (Eq. 5.3). Thus, the minimization of $Cost_{P_{NR}}$ is equivalent to the minimization of P_{NR} (as shown in Eq. 5.4) or the prioritization of the consumption of P_R.

$$Cost_{P_D} = w_1 \times Cost_{P_R} + w_2 \times Cost_{P_{NR}} \tag{5.1}$$

$$Cost_{P_{NR}} = \begin{cases} 0 & \text{if } P_{ES} = P_R, \\ \sum_{i=1}^{n} p_i, n \subset N & \text{if } P_{ES} = P_{NR}. \end{cases} \tag{5.2}$$

$$minimize\ \{Cost_{P_{NR}}\} \tag{5.3}$$

$$minimize\ \{P_{NR}\} \tag{5.4}$$

5.1.2.2 Objective Function

Adaptive energy management is achieved by adapting consumption to generation and aims at the optimal use or consumption of the available supply (whether renewable or not). In this proposal, the optimal adaptive consumption is obtained by minimizing the wasted or unused available power, which is mathematically expressed as the difference between the P_{ES} and the P_D, as shown in Eq 5.5. Considering the analysis for each time slot k, Eq 5.5 becomes Eq 5.6. In addition, for simplicity the difference between P_{ES} and P_D is referred to as residual power and is denoted as P_{RES}. Thus, the objective function (linear function) in the proposed adaptive energy management model is summarized in the minimization of the P_{RES}, respecting the constraints shown in Section 5.1.2.3.

$$minimize\ \{P_{ES} - P_D\} \tag{5.5}$$

$$\forall k \in W : minimize \left\{ \sum_{k=1}^{W} (P_{ES}[k] - P_D[k]) \right\} \tag{5.6}$$

5.1.2.3 Constraints

The constraints are divided into capacity and domain constraints and time constraints, as detailed below.

- *Capacity and domain constraints:* The non-negative value of the P_{ES} and the P_{RES} is ensured by $C1$ and $C2$, respectively. $C3.1$ guarantees the assignment of a single priority for each service. The variable y_{ij} takes on a value of 1 if and only if the priority $q_i = j$ for service S_i exists, as shown in $C3.2$.

  $C4$ limits the maximum consumption capacity. In $C4$, the decision variable shown in constraints $C5.1$ and $C5.2$ ensures the processing of the service S_i with a single time-shifting value u_i. This is because among all possible time-shifting values (i.e., $\{-u_i, \ldots, 0, \ldots, +u_i\}$) only one value must be chosen to avoid multiple copies of the same service. In this regard, the application of time-shifting to N services produces N mutually disjoint classes $V_1, \ldots, V_N$ of services. Each class V_i is composed of the shifted versions of the service S_i considering the interval $\{t_i - u_i, \ldots, t_i, \ldots, t_i + u_i\}$. Hereinafter, for convenience, the shifted versions of the service S_i are also referred to as *variations of the service* S_i, and each variation of S_i is denoted as $Var_r^{S_i}$ (e.g., $Var_1^{S_1}$).

  Considering that each variation r, with $r \in V_i$, demands a certain amount of power during a finite interval and at a given starting time, the problem of the adaptive energy management using the time-shifting mechanism consists of choosing the best variations per class V_i (i.e., if x_{ir} takes on a value of 1), such that utilization of the available supply P_{ES} is maximized (minimization of the P_{RES}). In this context, the set of N or n (with $n \subset N$) variations analyzed is defined as a *combination of variations* or simply a *combination* and is denoted as $Comb_f$ (e.g., $Comb_1$). The algorithmic strategies (optimal or heuristics) have the task of finding the best combination among all possible combinations denoted as $AllComb$ ($Comb_f \in AllComb$) produced due to different variations per class V_i. The selection of the best combination is explained in Section 5.1.2.5.

- *Time constraints:* The time horizon of analysis is fixed in the range from 0 up to W and is ensured by $C6$, $C7$, and $C8$. In addition, the temporal constraints for the ES are guranteed by $C9$ and $C10$.

$$C1 : P_{ES}[k] \geq 0 \tag{5.7}$$

$$C2 : (P_{ES}[k] - P_D[k]) \geq 0 \tag{5.8}$$

$$C3.1 : \sum_{j=1}^{L} y_{ij} = 1, i \in \{1, \ldots, N\} \tag{5.9}$$

$$C3.2 : y_{ij} \in \{0, 1\}, i \in \{1, \ldots, N\}, j \in L \tag{5.10}$$

$$C4 : \sum_{i=1}^{N} \sum_{r \in V_i} p_{ir}[k] \times x_{ir} \leq P_{ES}[k] \tag{5.11}$$

$$C5.1 : \sum_{r \in V_i} x_{ir} = 1, i \in \{1, \ldots, N\} \tag{5.12}$$

$$C5.2 : x_{ir} \in \{0, 1\}, i \in \{1, \ldots, N\}, r \in V_i \tag{5.13}$$

$$C6 : t_i \geq 0 \tag{5.14}$$

$$C7 : \{t_i - u_i\} \geq 0 \tag{5.15}$$

$$C8 : W \geq \max\{t_i + d_i + u_i\} \tag{5.16}$$

$$C9 : T_{init}^{P_{ES}} \geq 0 \tag{5.17}$$

$$C10 : W \geq \{T_{init}^{P_{ES}} + m\} \tag{5.18}$$

5.1.2.4 Metrics to Evaluate the Combination of Variations

To quantitatively evaluate which combination or individual variation (if the algorithmic strategy works only with variations) produces the minimum value of the P_{RES} while maintaining the best comfort level of ECs (as far as possible processing of all N services with $u_i = 0, \forall i \in N$), three performance metrics are required and are defined as follows.

- *Standard deviation of residual power ($\sigma_{P_{RES}}$):* This metric measures the amount of the P_{RES} of a combination. A lower $\sigma_{P_{RES}}$ means better use of the P_{ES}, and the best value is $\sigma_{P_{RES}=0}$ if $P_{ES} = P_D$. The expression of $\sigma_{P_{RES}}$ within m is given by:

$$\sigma_{P_{RES_f}} = \sqrt{\frac{\sum (P_{REScomb_f})^2}{m}} \tag{5.19}$$

- *Acceptance ratio (AR):* This metric indicates the percentage of variations (services) that have been processed (i.e., services for which power has been allocated). The selection of unprocessed or rejected variations (services) denoted as $RejServ$, so that $P_{RES} \geq 0$ (constraint $C2$), is carried out by algorithmic strategies (optimal and heuristics). The criterion for the rejection of variation(s) is given first as a function of priority level (analysis in descending order of priorities from $j = 1$ down to $j = L$) and secondly by selecting those variations whose rejection allows minimizing the P_{RES} (i.e., selection of variations whose execution maximizes the use of available supply). Considering a total of $RejServ$ rejected variations (services), the AR can be expressed as:

$$AR_f = \frac{N - RejServ}{N} \times 100\,\% \tag{5.20}$$

To evaluate the missing amount of power (if $P_{ES} < P_D$ or the services cannot be moved to use the energy profile) to reach an $AR = 100\%$, the metric P_{LACK} or $P_{LACK(AR=100\%)}$ is additionally defined. This metric is not used to select the best combination. However, it is used in the algorithmic strategies to verify how efficiently the available energy is used by the combination analyzed (combination produced by algorithmic strategies). An adaptive energy management solution aims to deliver the lowest value of the P_{LACK} (with the best value if $P_{LACK} = 0$) produced if the available supply is optimally consumed. The P_{LACK} mathematically is expressed as:

$$P_{LACK_f} = \begin{cases} |P_{ES} - P_{DComb_f}| & \text{if } P_{ES} < P_{DComb_f}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.21}$$

In addition, with the mean value of the P_{ES}, the estimation of the interval m to promote an $AR = 100\%$ (if consumption can be adapted to the energy profile) is given by Eq. 5.22.

$$m \simeq \frac{\sum_{i=1}^{N} p_i \times d_i}{\overline{P_{ES}}} \tag{5.22}$$

- *Standard deviation of time shifting (σ_{Ts}):* This metric measures the cumulative time-shifting in a combination of variations. A lower σ_{Ts} stands for lower application of the time-shifting on variations, and the best value is $\sigma_{Ts} = 0$. The expression of σ_{Ts} is given by:

$$\sigma_{Ts_f} = \sqrt{\frac{(\sum_{\forall i \in N} u_i)^2}{N}} \tag{5.23}$$

5.1.2.5 Adaptive Energy Management Based on a Cost Function

The selection of the best combination of variations (i.e., the combination that produces the minimization of the P_{RES}) can be established based on the best result of one (mainly based on the values of $\sigma_{P_{RES_f}}$) or all metrics in Section 5.1.2.4. Nevertheless, to obtain the optimal allocation of energy resources and the corresponding optimal consumption, all the metrics and parameters related to adaptive energy management must be considered together. To this end, a possible mechanism shown in [14] is to perform a nested sorting to the set of all combinations *AllComb* based on an increasing value of $\sigma_{P_{RES_f}}$ and σ_{Ts_f} and a decreasing value of σ_{Ts_f}. As a result of this process, the first combination in the sorted list represents the optimal service scheduling that enables the minimization of the P_{RES}. Although the nested sorting method delivers the best combination of variations, it is limited to a small number of metrics because each new metric increases the complexity of the process by a factor equal to the size of the set *AllComb*; also, this method does not incorporate priority information. To overcome these limitations, in the proposed energy management model, the selection of the best combination is carried out in a single sorting step using the cost function shown in Eq. 5.24, which allows incorporating different parameters (more than three metrics) to cope with the specific requirements of the ES or the ECs.

The cost function in Eq. 5.24 is composed of individual costs related to $\sigma_{P_{RES}}$, AR, σ_{Ts}, and the cumulative value of priorities of the variations in the combination, which are denoted as $Cost_{P_{RES_f}}$, $Cost_{AR_f}$, $Cost_{U_f}$, and $Cost_{Q_f}$ and are given by Eq. 5.25, Eq. 5.26, Eq. 5.27, and Eq. 5.28, respectively. In addition, the values of the costs in Eq. 5.24 can be tuned by the EM according to the preference for some parameter(s) using the weights α, β, γ, and δ in the range [0,1]. For analytical simplicity, these weights are set to one.

$$Cost_{comb_f} = \alpha \times Cost_{P_{RES_f}} + \beta \times Cost_{AR_f} + \gamma \times Cost_{Q_f} + \delta \times Cost_{U_f} \tag{5.24}$$

$$Cost_{P_{RES_f}} = \begin{cases} \sigma_{P_{RES_f}} \times \varepsilon & \text{if } AR_f = 100\%, \\ \sigma_{P_{RES_f}} & \text{otherwise.} \end{cases} \tag{5.25}$$

$$Cost_{AR_f} = \begin{cases} 0 & \text{if } AR_f = 100\%, \\ RejServ \times \mathcal{M} & \text{otherwise.} \end{cases} \tag{5.26}$$

$$Cost_{U_f} = \begin{cases} 0 & \text{if } \forall Var_r^{S_i} : u_{ir} = 0, \\ \sum_{i=1}^{n} \sum_{r \in V_i} u_{ir} & \text{otherwise.} \end{cases} \tag{5.27}$$

$$Cost_{Q_f} = \begin{cases} 0 & \text{if } \forall Var_r^{S_i} : q_{ir} = j, \\ \sum_{i=1}^{n} \sum_{r \in V_i} q_{ir} & \text{otherwise.} \end{cases} \tag{5.28}$$

In Eq. 5.25, the parameter ε can be set in the range [0,1] ($\varepsilon = 0.5$) and can be used to differentiate the $Cost_{P_{RES_f}}$ of the different combinations of variations if $P_{RES} > 0$. In Eq. 5.26, instead, the parameter $\mathcal{M}$ represents a big value (e.g., $\mathcal{M} = 1000$) and is used to penalize those combinations that deliver more rejected variations. In addition, Eq. 5.26 can include the cumulative value of priorities of variations (i.e., $\sum_{i \in RejServ} q_i$) to penalize those combinations that produce higher priority rejected services.

The corresponding cost function is calculated for each combination of variations f, producing a list of costs denoted as *AllCost*. From this list, the best cost (*OptCost*) is the one with the lowest value, as shown in Eq. 5.29, and corresponds to the best combinations of variations (*OptComb*, with *OptComb* $\in$ *AllComb*) that enable adaptive management through the minimization of the P_{RES}.

$$OptCost = argmin\{Cost_{comb_f}\}, Cost_{comb_f} \in AllCost \tag{5.29}$$

5.1.3 Hardness of the problem

The objective of adaptive energy management in our proposal consists of maximizing the use of the available energy supply (minimization of the P_{RES}) through the execution (selection) of the best variations of services (one variation per class V_i, with $i \in N$). This process is analogous to the objective of the multi-dimensional multi-choice knapsack problem (MMKP) of choosing the most valuable items of a set of classes (one item per class) without overloading the knapsack [102], which has been demonstrated to be NP-hard [102]. Based on this analogy, we can then conclude that the proposed energy model falls in the MMKP classification and presents a complexity that is NP-hard.

5.2 Optimal solution: OptTsCost

To optimally solve the ILP model in Section 5.1.2, different approaches can be used. This section presents an exact or optimal algorithmic strategy defined as OptTsCost based on a brute-force exhaustive search paradigm in which the entire search space is explored to find the optimal solution. The algorithm is explained in Fig. 5.2 and starts with the computation of the variations of the services considering all values within the time-shifting interval (i.e., $\{t_i - u_i, \ldots, t_i, \ldots, t_i + u_i\}$). With the information on variations, the algorithm builds all possible combinations of variations. It then computes the metrics and cost function for each combination. Later, it selects the best combination based on the minimum value of the $Cost_{comb_f}$. This process is carried out iteratively for each priority level, and at the end the strategy OptTsCost delivers the optimal service scheduling that produces the minimization of the P_{RES}. Figure 5.3 shows a summarized example of the application of strategy OptTsCost for $N = 4$ services.

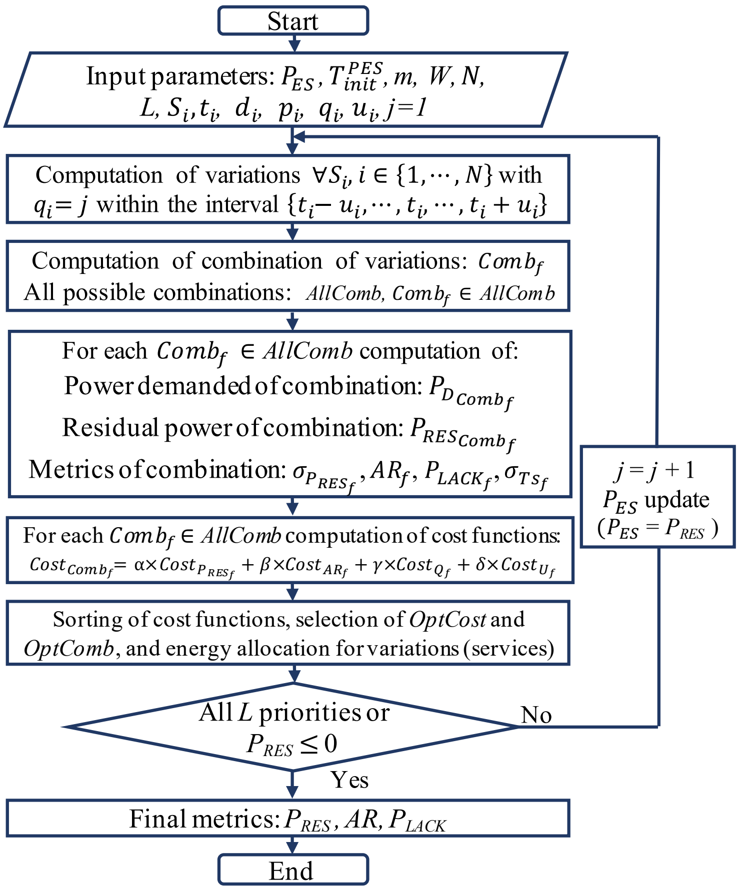


FIGURE 5.2: Flow chart of OptTsCost.

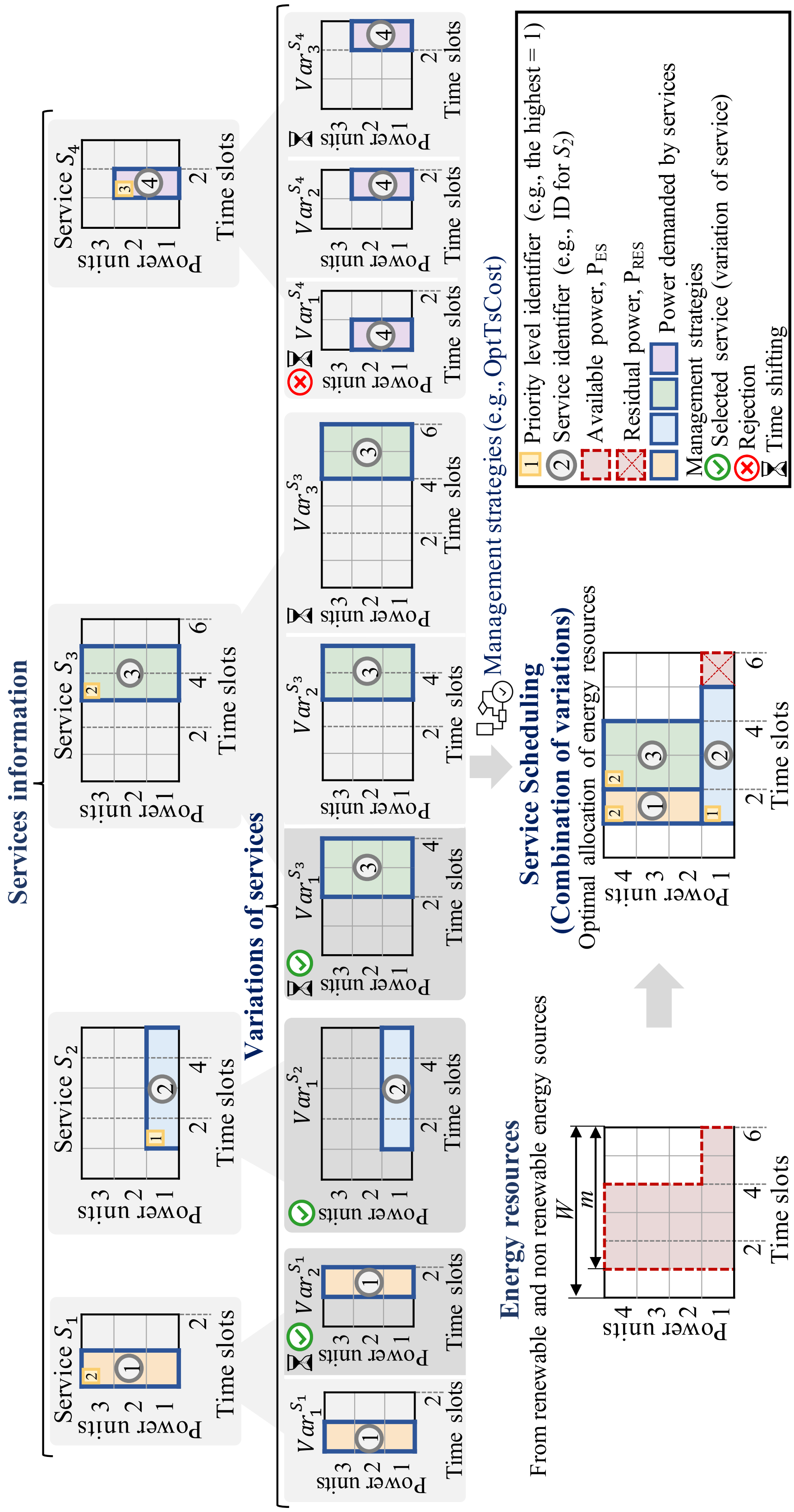


FIGURE 5.3: Example of application of the strategy OPTTSCOST.
In this example, the optimal service scheduling that produces the minimization of the P_{RES} is achieved if S_1 is executed one slot in delay, S_2 is processed in its original execution time, S_3 is executed one slot in advance, and S_4 is rejected.

The complexity in OptTsCost depends on the steps carried out. Considering for analytical simplicity the same value for the time-shifting backward and forward (u_i), the growth rate of OptTsCost as a function of N, u_i, and L can be summarized in Eq. 5.30. The terms that comprise this expression correspond to the number of services, variations, and combinations processed for each priority level. Of all the terms in Eq. 5.30, the most dominant is the third, which reveals that OptTsCost presents an exponential growth rate with an order of growth $\mathcal{O}(2^N)$ (according to Big-O notation) that depends on the maximum values of N and u_i. This condition imposes the applicability of the OptTsCost to small-scale scenarios. For instance, using only $N = 10$ services and $u_i = 4$ time slots the algorithm has to produce and later analyze over three billion combinations to find the best service scheduling.

$$f(L, N, u_i) = N + \big((2 \times N \times u_i + N) + (2 \times u_i + 1)^N\big) \times L \tag{5.30}$$

As an example of the deployment of management strategies as SFCs in the NFV domain, Fig. 5.4 illustrates the strategy OptTsCost decomposed into VNFs. These VNFs correspond to the different steps carried out by the algorithm to minimize the P_{RES}, can be deployed on virtual machines or containers, and can use on-demand computation resources according to the needs of the ES and the ECS.

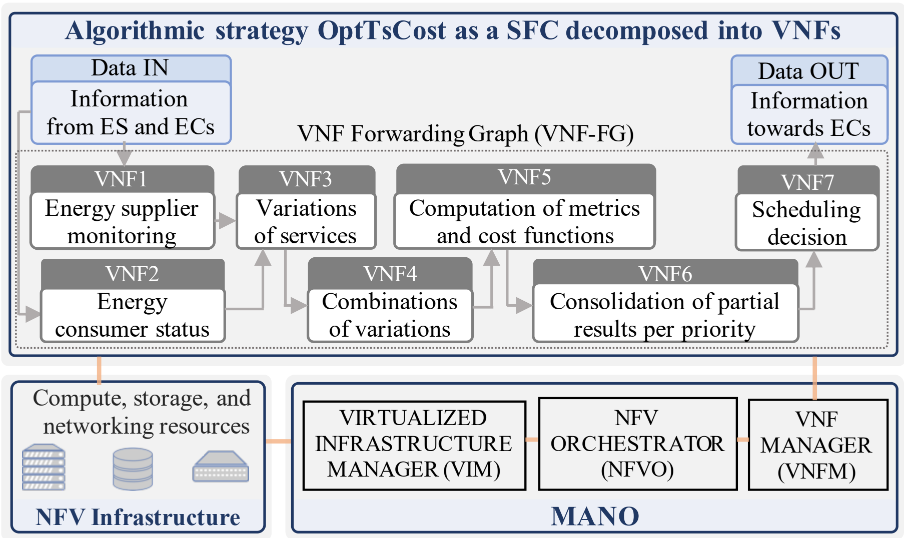


Figure 5.4: Example of the deployment of OptTsCost as a SFC in the NFV-enabled EM.

5.3 Heuristic strategies

Although the strategy OptTsCost is a powerful approach for finding optimal consumption conditions, evaluating all possible combinations of variations is computationally demanding and intractable for values of $N > 10$ services or $u_1 > 4$ time slots [14]. These operating conditions motivate the development of less complex approaches in which the optimal results can be relaxed to obtain reduced running time and lower utilization of computational capacity. In addition, a reduction of complexity can enable adaptive energy management in large-scale scenarios (e.g., for $N = 100$ services or $u_i = 10$ time slots).

Considering that the proposed energy management model is categorized as an MMKP, different existing techniques and strategies can be adapted to the context of adaptive energy management. In this regard, to tackle the exponential complexity of OptTsCost and to cover as far as possible the main categories of methods or techniques for efficiently solving the MMKP (in our proposal, the P_{RES} minimization problem) [103], three heuristic algorithmic strategies have been proposed: (i) a greedy strategy, (ii) a genetic algorithm-based solution, and (iii) a strategy based on a dynamic programming method. These strategies are described below.

5.3.0.1 Greedy Strategy: GreedyTs

Greedy algorithms are simple schemes intended to produce feasible solutions quickly. These algorithms are iterative and constructive in the sense that starting with an empty solution, in every iteration part of the solution is obtained (never changed later) so in the last iteration the complete solution is created. The decision in each iteration is made in an attempt to optimize a performance metric or maximize an immediate benefit (e.g., taking first the most valuable item and then the next most valuable in the 1/0 knapsack problem) [104]. In the context of adaptive energy management, the proposed greedy strategy defined as GreedyTs iteratively builds the optimal service scheduling that produces the minimization of the P_{RES}. The algorithmic strategy GreedyTs is explained in Fig. 5.5, the main steps carried out are summarized below.

1. *Analysis of services:* Unlike strategy OptTsCost, the algorithm GreedyTs works with the variations of services instead of combinations. This feature relaxes the complexity in adaptive energy management and reduces the analysis to an iterative search for the best variation per service, such that the best variations together produce the optimal service scheduling. Thus, for each priority level, as a first step the algorithm sorts the services according to the decreasing value of the $\sigma_{P_{RES}}$ of service (denoted as σ_{Pres^i}) and t_i. This criterion aims to maximize the number of services that can be processed. If the available supply is first allocated to small energy demands with earlier starting times, there is a greater amount of P_{RES} that can be used effectively by a greater number of services.
2. *Analysis of variations:* The services in the sorted list are analyzed iteratively until all N services are covered. For each service S_i, the corresponding variations are computed within the respective time-shifting interval. Next, two parameters are computed for each variation $Var_r^{S_i}$ obtained: (i) the cost function ($Cost_{Var_r^i}$) composed of the cost related to the residual power of variation ($Cost_{Pres_r^i}$) and the cost related to the time-shifting ($Cost_{U_r^i}$), and (ii) the gradient related to the residual power of variation ($\nabla_{Pres_r^i}$). Subsequently, the variation that produces the lowest

cost $Cost_{Var_r^i}$ and the highest gradient $\nabla_{Pres_r^i}$ is selected, and the corresponding energy is allocated. A lower cost means better use of the available energy, and a higher gradient indicates a greater energy resource for the next service. Once all L priority levels have been explored, the algorithm delivers the metrics P_{RES}, AR, and P_{LACK}, applied to the set of variations (services) processed.

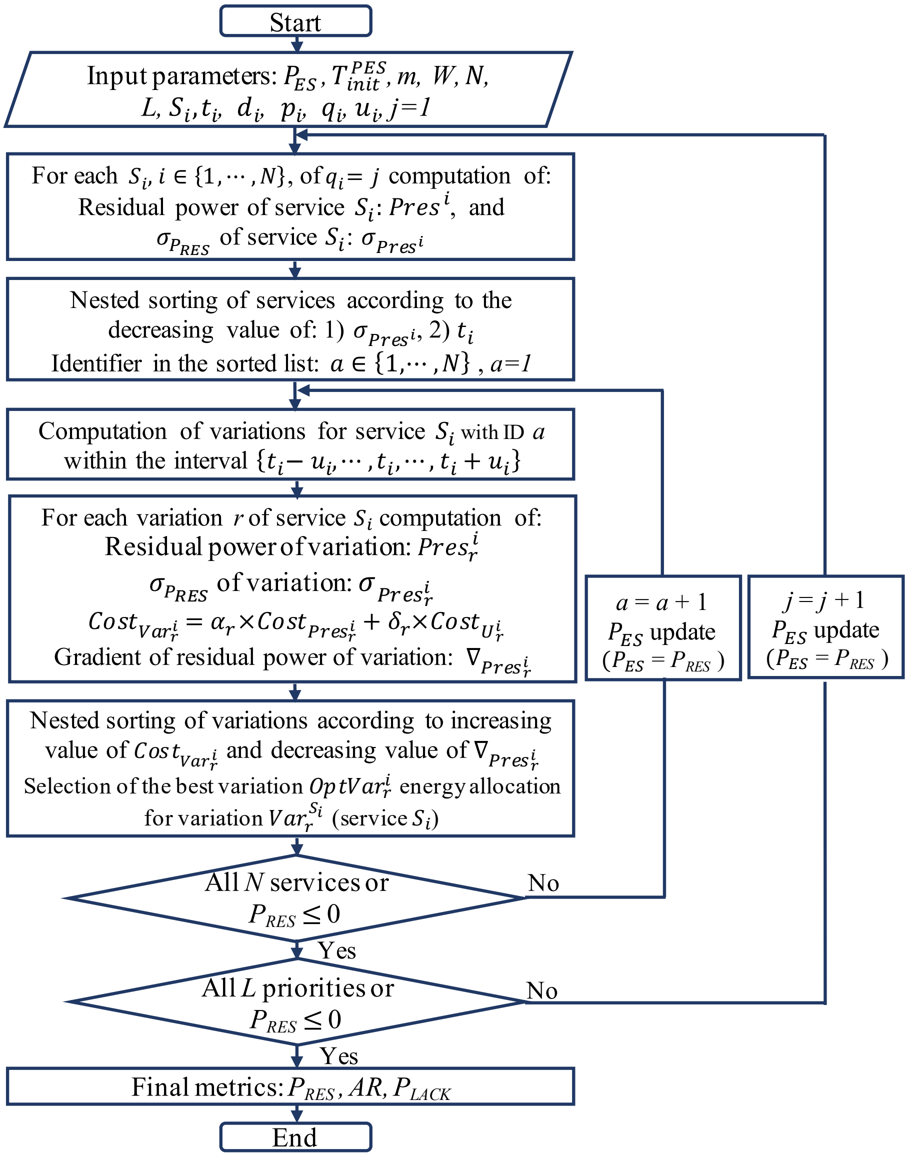


FIGURE 5.5: Flow chart of GREEDYTS.

The complexity of GREEDYTS is summarized in Eq. 5.31. In this expression, the second term is dominant and reveals that the growth rate of the algorithm is polynomial and depends on the maximum values of N and u_i. However, if one of the parameters remains constant (whether N or u_i), the growth rate of GREEDYTS can become linear.

$$f(L, N, u_i) = N + (2 \times N \times u_i + N) \times L \tag{5.31}$$

5.3.0.2 Genetic Algorithm Based Strategy: GATs

Genetic algorithms belong to the class of evolutionary algorithms and are search techniques used to generate near-optimal solutions in varied optimization problems [105]. These algorithms are inspired by the biological evolution of the species over generations and by the process of natural selection of the fittest. Regarding the implementation, genetic algorithms follow principles of natural genetics and rely on operators such as reproduction, mutation, crossover, and selection [104].

The evolution process starts with a population of possible solutions to an optimization problem created using a random or greedy method. In this population, each candidate solution is called a chromosome or individual (in our problem, combination of variations), is composed of genes (in our problem variations of services), and can be affected by mutation and crossover operators. For each chromosome in the population, a fitness function is computed, which is commonly related to the objective function (e.g., minimization of the P_{RES}) of the optimization problem to be solved. If within the initial population there is at least one chromosome that meets the desired value of the fitness function (e.g., a combination that produces the optimal P_{RES}), the algorithm could terminate its execution; otherwise, iteratively, the population continues to evolve towards better solutions. In genetic algorithms, each iteration emulates a generation.

In each generation, the algorithm selects a set of chromosomes (denoted as parents) from the current population for reproduction. The selection of parents can be based on a random process or can be related to the value of the fitness function (e.g., roulette wheel selection or elitist selection). From the set of parents, couples (pairs) are created that, using a crossover operator (e.g., one-point operator), produce an offspring (child solutions) that inherits characteristics from both progenitors. The algorithm can let a parent be chosen more than once to form a couple. Commonly, the fitter the chromosome (better solution), the more times it is likely to be selected to reproduce. The number of couples and children created in a generation determines the size of the offspring, which usually is a percentage (e.g., 50%) of the current population. In addition, to increase diversity, a fraction of the offspring solutions can be affected by mutation operators (e.g., swap mutation), that is, the chromosomes can be slightly and randomly changed to emulate mutations. Finally, the natural selection process of creating a new generation is carried out by replacing the worst chromosomes of the current population with the best offspring solutions, usually keeping the initial population size. The execution of the algorithm is carried out iteratively, selecting the best solutions in each generation (iteration) until any stop criterion is met, for instance, a maximum number of generations or a given level of the fitness function.

In the context of adaptive energy management, the proposed genetic-based strategy defined as GATs iteratively creates a set of combinations of variations and selects the ones with the best P_{RES} values for the next generation. In the end, the algorithm delivers the best possible combination that leads to the minimization of the P_{RES}. The algorithmic strategy is explained in Fig. 5.6, and the main steps carried out are summarized below.

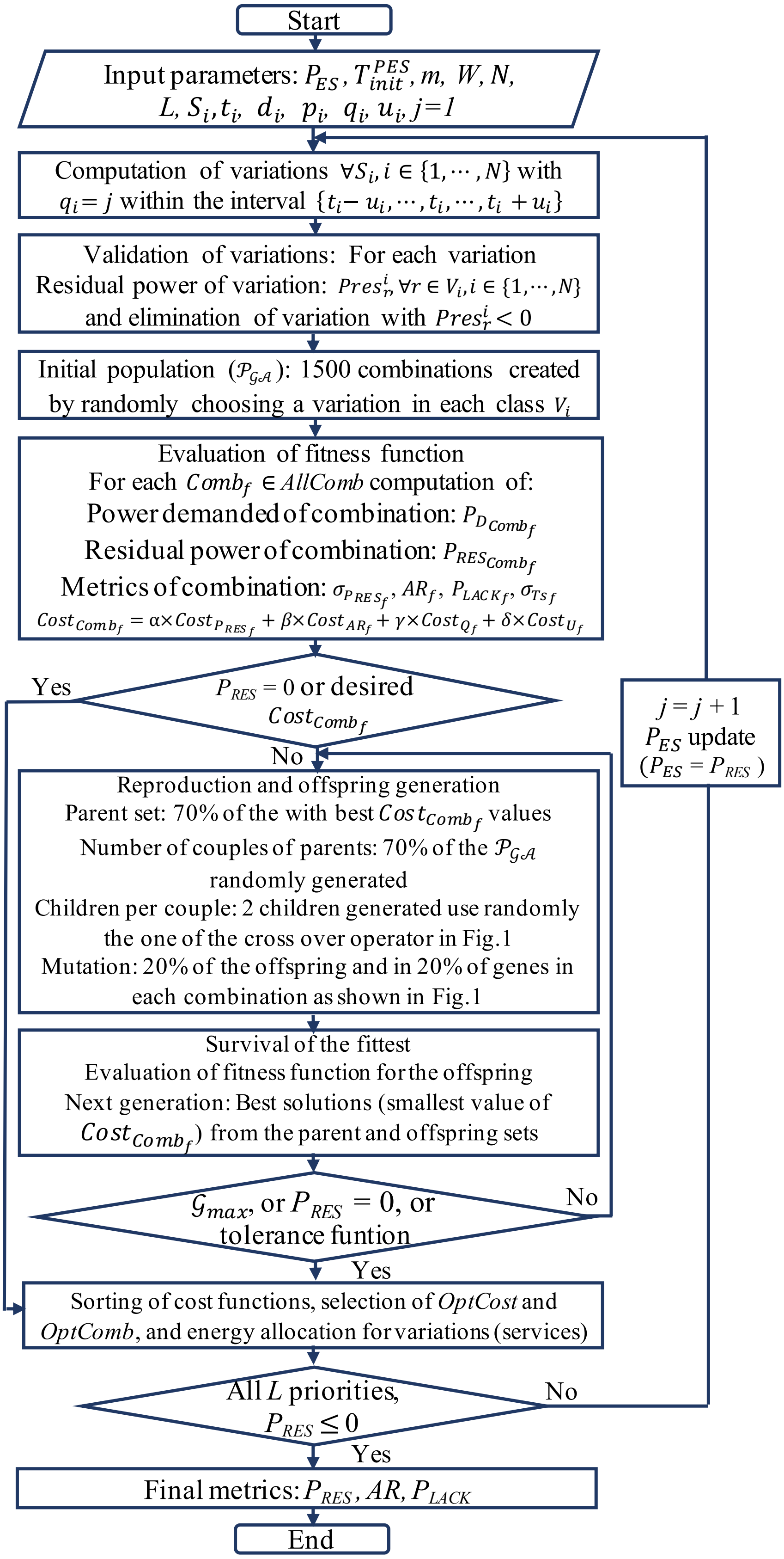


FIGURE 5.6: Flow chart of GATs.

1. *Coding the solution:* In the genetic algorithms and adaptive energy management scope, a chromosome represents a combination of variations, and each gene represents a variation of service.

2. *Validation of variations:* For each priority level, in this step those variations that individually produce a negative residual power ($Pres_r^i$) are eliminated. This procedure allows reducing the number of combinations with a $P_{REScomb} < 0$ and avoiding the analysis for the rejection of the corresponding service(s).

3. *Initial population:* It is recommended that the population (denoted as $\mathcal{P}_{\mathcal{GA}}$) has hundreds or thousands of chromosomes to guarantee diversity in solutions [104]. In this regard, a preliminary analysis in [14] has demonstrated that if the number of combinations is lower than or equal to 2401 (a situation that occurs when $u_i = 3$ and $N = 4$), the exploration of the entire search space can be performed in less running time (e.g., units of seconds or less). Based on this reference and after preliminary tests (tests applied to set other parameters of the algorithm) on the quality of the solutions produced, the initial population size has been set at a maximum of 1500 non-repeated combinations, which are generated by randomly choosing the variations (valid) of each class V_i. In the case that the theoretical number of combinations $\big((2 \times u_i + 1)^N\big)$ is lower than 1500, all possible combinations are used in the population. The population size remains constant for the rest of the algorithm steps.

4. *Evaluation of fitness function:* The fitness function in the adaptive energy management is represented by the cost function $Cost_{comb_f}$, which is computed for all combinations in the population. If within the initial population there are combinations that produce a desired performance metric or threshold (e.g., $P_{RES} = 0$ or a given $Cost_{comb_f}$), the combination with the best value (e.g., minimum $Cost_{comb_f}$) is chosen, and immediately the energy is allocated to the respective variations. Otherwise, the algorithm continues to iteratively create and analyze generations of combinations to obtain better values of $Cost_{comb_f}$.

5. *Reproduction and offspring generation:* For reproduction, 70% of the population (ensuring an even number of combinations) with the minimum cost function values ($Cost_{comb_f}$) is chosen to create the parent set (i.e., following an elitist approach). The combinations in this set are randomly selected to form pairs. At this point, the number of pairs is half the number of combinations of the paternal set because a parent can only participate in one pair at a time. Each pair produces two child solutions created by mixing the variation of progenitors and randomly using one of the three established crossover operators (one-point, multipoint, and uniform crossover operators). Figure 5.7 shows an example of obtaining offspring and the application of crossover operators.

   Twenty percent of the offspring obtained are affected by the mutation process. A mutation operator is applied to each solution (combination) of this group, which that affects approximately 20% (genes) of the variations. Usually, the mutation operator randomly exchanges/flips some genes in the analyzed chromosome. However, in the context of the proposal this procedure is performed by selecting a different variation from the one analyzed in the class V_i; an example is shown in Figure 5.7.

Throughout generating offspring and applying genetic operators, the created and mutated solutions (children) are conditioned to be different from the combinations of the current population (parents) to guarantee diversity in the search space.

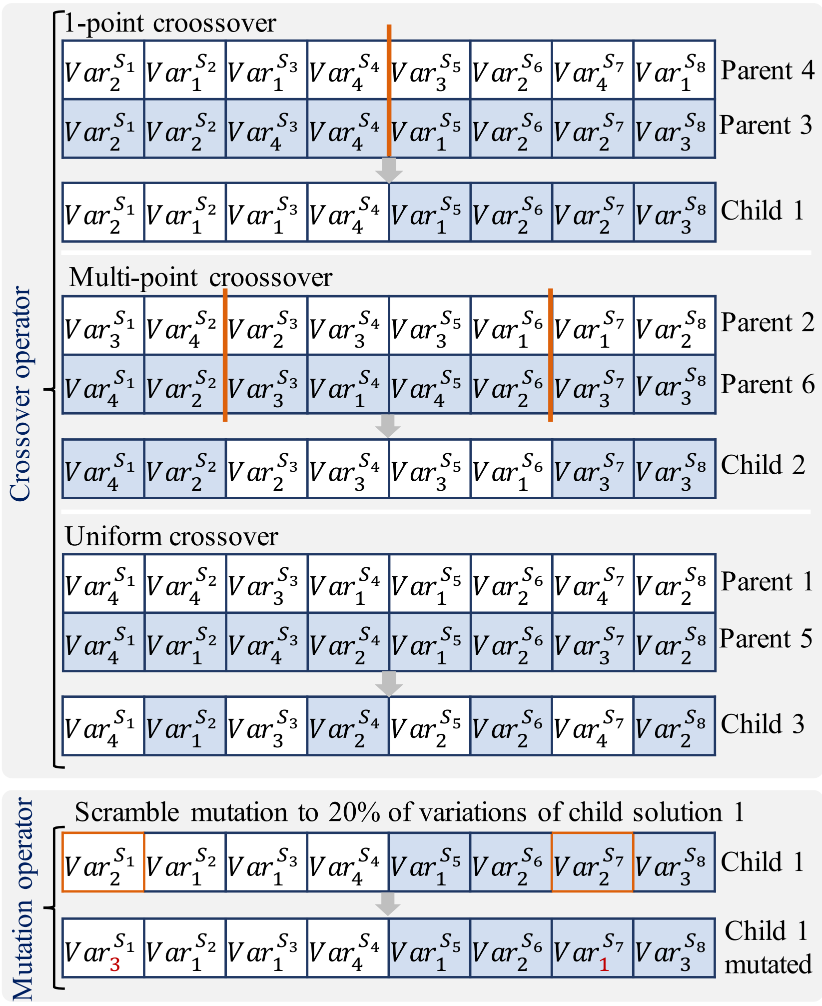


FIGURE 5.7: Example of crossover and mutation operators in the Genetic Algorithm for $N = 8$ services.

6. *Survival of the fittest:* Following an elitist approach, the combinations with the minimum $Cost_{comb_f}$s values from the parent and offspring sets are selected to build the next generation of individuals (of the same size as the current population $\mathcal{P}_{\mathcal{GA}}$). The algorithm then continues the iterative process until any of the following stop conditions is met: (i) reach the maximum number of generations $\mathcal{G}_{max}$, which in the strategy is fixed to 20 generations based on previous validation of solutions, (ii) obtain at least a combination with $P_{REScomb} = 0$ (i.e., ensure the utilization of all available supply), or (iii) obtain a relative change in the value of the $P_{REScomb}$ lower than the tolerance function, which has been set as 1% of the $P_{REScomb}$ of the previous generation.

The complexity of GATs depends on the number of variations analyzed and the combinations explored over generations, as summarized in Eq. 5.32. The process of exploration of combinations is dominant in Eq. 5.32 and reveals that the growth rate of the algorithmic strategy can be: (i) exponential for values of N and u_i that produce a number of combinations smaller than 1500 (e.g., for $N = 4$ and $u_i = 2$) or (ii) polynomial if the population remains constant (1500 combinations) and there are variations (increases) in the number of generations used to reach feasible solutions. In a particular case, the growth rate of GATs can be linear if the number of generations remains constant as analyzed services increase.

$$f(L, N, u_i) = N + \big((2 \times N \times u_i + N) + \mathcal{P}_{\mathcal{GA}} \times \mathcal{G}_{max}\big) \times L \tag{5.32}$$

5.3.0.3 Dynamic Programming Strategy: DPTs

Dynamic programming (DP) is a mathematical technique that solves a complex problem optimally or sub-optimally by breaking it into simpler subproblems. Then, each of those subproblems is solved (optimally) just once, and their solutions are stored (in a data structure, e.g., an array) so that they can be used (repeatedly if necessary) to solve the original problem. DP can be applied when a solution to a problem can be recursively described in terms of solutions to subproblems (i.e., when the subproblems overlap) [106]. For instance, in a naive recurrent Fibonacci computation, the same values are computed repeatedly for each new number. A simple computation for the second and third Fibonacci numbers requires the computation of the first Fibonacci number twice (one for the second and one for the third). DP solves this issue by storing the already computed values so that the second time they are needed, they can be obtained immediately. In addition, since the subproblems are interrelated, the final solution can be obtained easily using a traceback process through the partial solutions. A variety of computational and optimization problems can be addressed using DP approaches, including solving MMKP [107].

The storage of partial solutions in DP provides high-quality results (optimal or suboptimal) and reduced time (complexity) compared to other methods such as brute force strategies [106]. There are two ways to store the partial solutions so that these values can be reused: (i) a bottom-down approach, also known as tabulation, that iteratively solves all subproblems and uses their solutions to fill up a table (a data structure whereby starting from the first entry, all entries are filled one by one), then, the stored results in the table are used to compute the solution to the original problem (or bigger subproblems); and (ii) a top-down approach, also known as memoization, that also uses a tabular form to store partial solutions but differs from the tabulation approach because the table is filled on-demand. Before solving a subproblem in the top-down approach, the algorithm will search for its solution in the lookup table. If the solution has been stored, this result can be directly used; otherwise, the problem is solved, and its solution is stored in the table so that it can be used later.

The computational implementation of tabulation is based on an iterative method; instead, memoization exploits recursivity. If a certain problem requires all subproblems to be solved (as in the case of adaptive energy management, in which all services (variations) must be analyzed to determine the set of N services that minimize P_{RES}), tabulation usually outperforms memoization. This is because the former has no overhead for recursion, and the required solutions can be directly retrieved from values in the table. Moreover, tabulation has been proven to be an effective DP method for solving knapsack problems [108]. Based on this information, a tabulation-based approach has been chosen for implementing the proposed DP algorithmic strategy.

In the context of adaptive energy management, the DP-based strategy, defined as DPTs progressively analyses variations, and by means of a tabulation approach, it selects the ones that enable P_{RES} minimization while maximizing the AR. The algorithmic strategy is explained in Fig. 5.8, and the main steps carried out are summarized below.

1. *Analysis of services and variations:* Like the GreedyTs approach, the strategy DPTs focused on the analysis of variations (instead of combinations) for each priority level. As the first step, the algorithm computes the variations of services within the corresponding time-shifting intervals. Then, the variations that individually produce a negative residual power ($Pres_r^i$) are eliminated. This procedure, by reducing the search space for the best variations, contributes to speeding up the execution of the algorithm.

   Once the variations are validated, the algorithm sorts them based on the increasing value of the starting time (t_i). This criterion aims to maximize energy use and considers the time evolution of the available energy resource (i.e., $T_{init}^{P_{ES}}$ and m). Thus, an efficient allocation in the first (earlier) services will promote a greater remaining available power (more freedom to select variations that minimize the P_{RES}) for the subsequent services. At this point, the strategy DPTs selects the variation with the lowest starting time (i.e., the first one) from the sorted list $sortedVarList$ and processes all concurrent/simultaneous variations ($DPVar$). A variation is considered concurrent if executed (coexits) within the lifetime (d_i) of the variation under analysis. The idea is to analyze a set of concurrent variations instead of individual variations and solve them using a DP method in the next step of the algorithm. Given that the DPTs algorithmic strategy progressively analyzes groups of simultaneous variations instead of all possible variations for practicality and to reduce the complexity of the associated search, the solution produced by the strategy is not optimal. However, studies prove that DP applied to MMKP can deliver high-quality approximate or suboptimal solutions reasonably quickly (compared to other methods such as brute-force) [109].

2. *Dynamic programming tabulation and the selection of processed variations:* In this step, the algorithm applies DP to the concurrent variations to select the ones whose execution optimizes P_{RES} and maximizes AR. For the implementation of DP, the algorithm uses a tabulation method and considers the energy resource that coexists with the time slots of variations analyzed. Figure 5.9 shows an example of the application of DP for a set of seven variations that belong to four services.

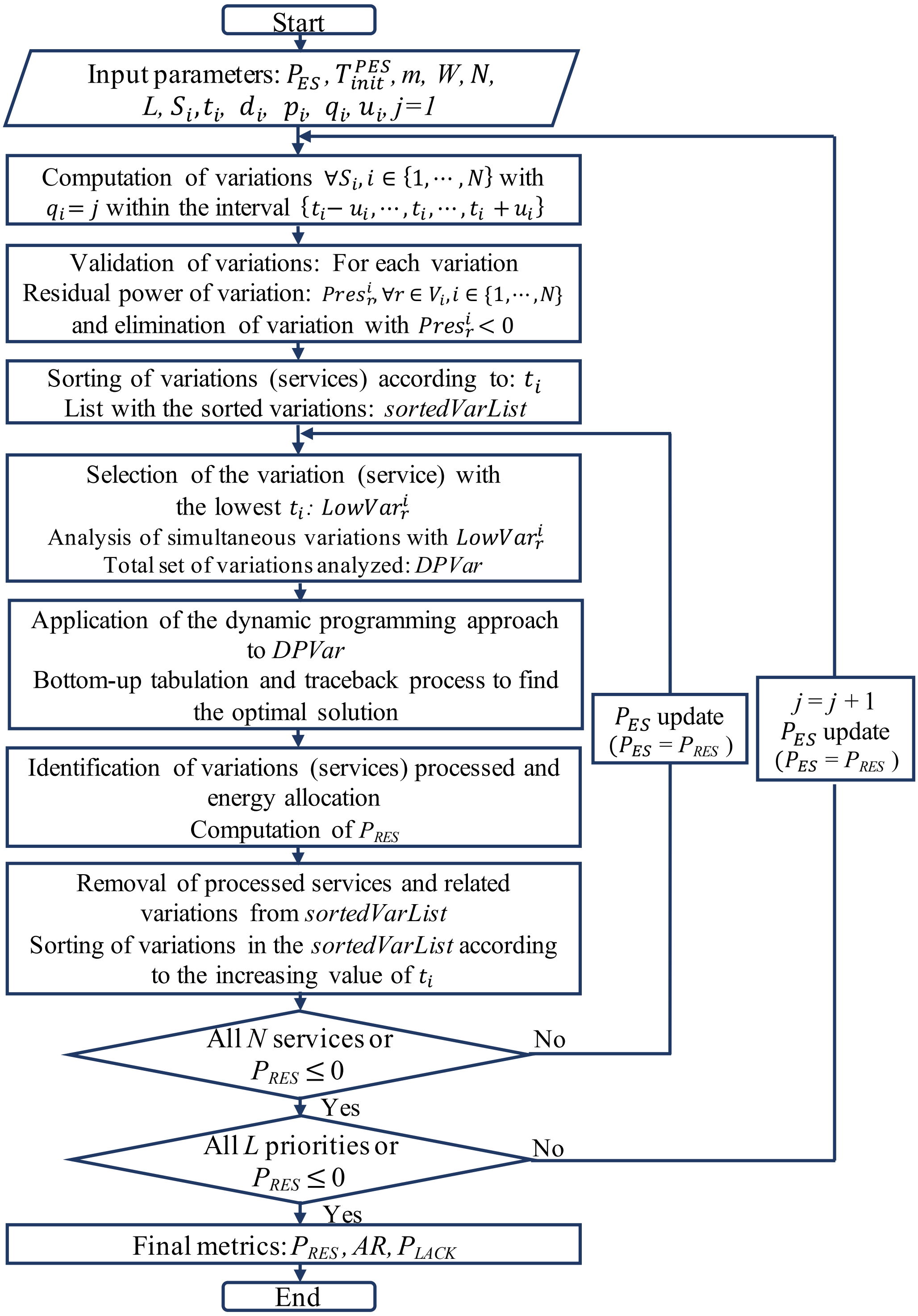


FIGURE 5.8: Flow chart of DPTs.

To carry out DP on variations, the algorithm starts by creating a table with a number of rows equal to the energy capacity per time slot K plus one unit (e.g., $P_{ES}[k] + 1 = 3 + 1$ for time slot 10 in Fig. 5.9) and with a number of columns equal to the number of concurrent variations plus one unit (e.g., n +1 = 4 +1 in

Fig. 5.9). A rigorous implementation of the energy model described in Section 5.1.2 requires that the algorithm performs the DP method for each slot k. In this case, the algorithm determines the processed variations based on the results obtained during all time slots analyzed. A variation is considered as processed if the algorithm (using a DP method) can allocate energy for all time slots in which it exists (i.e., for d_i). Moreover, for a more exhaustive assessment of processed variations, the algorithm DPTs can include a simple greedy-based method that verifies the power demanded by each processed variation and the remaining available power. In the event that the level of P_{ES} remains constant during all time slots analyzed (as shown in Fig. 5.9), the algorithm could simplify the DP analysis to the first time slot (time slot 10 in the example), which can speed up the exploration of optimal variations of services. In any of the cases, before carrying out the DP, the algorithm evaluates the generation and consumption conditions for the set of variations analyzed.

In the created DP table, the rows with identifier a represent all possible values of available power (i.e., $0 \leq a \leq P_{ES}$). In contrast, the columns with identifier b correspond to the individual information of variations (i.e., $0 \leq b \leq n$), including the information when no variation is selected (the first column). Based on the adaptation of the DP method to solve MMKP [107], the entries in the table of strategy DPTs correspond to the cumulative optimal value or profit due to the selection of variation(s) respecting the maximum energy capacity. The individual value linked to each variation (service) is identified as v_b, and the entry stored in the table in row a and column b is denoted as $\mathcal{V}[a, b]$. The value v_b in the proposed strategy DPTs is assigned according to the priority level of the variation in analysis so that a higher value is assigned to a higher priority (e.g., $v_b = L$ if $q_b = 1$). If all the concurrent variations have the same priority level, the value v_b is unique and is set to one (i.e., $v_b = 1 \, \forall b \in n$, as shown in Fig. 5.9). An alternative criterion to establish the v_b value could be based on a cost function or metric (e.g., P_{RES}) that indicates the impact on energy use due to the selection (processing) of certain service(s).

The strategy DPTs systematically fills up the table colum by colum (i.e., one service at a time, from 0 up to n). For each column b, the algorithm progresively analyses each row a (from 0 up to P_{ES}) and assesses the selection (or not) of the variation with identifier b so that the value of the entry $\mathcal{V}[a, b]$ corresponding to the selected variation(s) $\hat{n}$, with $\hat{n} \subset n$ (i.e., $\mathcal{V}[a, b] = \sum_{b \in \hat{n}} v_b$) is maximized (optimal). In each row a, the algorithm verifies that the power demanded by the selected variation(s) fits the available energy capacity. Moreover, in each column and corresponding row, the algorithm verifies if the value for the entry $\mathcal{V}[a, b]$ has been already computed in the previous column to avoid recomputation of the same value (i.e., using stored values to solve greater subproblems). This condition gives the bottom-up approach to the tabulation method and makes the space memory and running time rather efficient. In summary, the computation of any entry $\mathcal{V}[a, b]$ depends on the power demanded and values of the variations in column b, and values in the previous column of the table, as established by Bellman's equation [106], that adapted to the context of the strategy DPTs is shown in Eq. 5.33.

At the end, of the exploration of all entries, the best possible cumulative value is stored at the bottom right corner of the table (i.e., at $\mathcal{V}[a_{th}, b_{th}]$). In the example in

Fig. 5.9, the best value is equal to three, which indicates that three variations (services) have been processed, and in this particular case, they use the P_{ES} optimally. To identify the variations that have been selected (processed), the algorithm uses a traceback method, in which, one by one, the columns (variations) are analyzed. The process starts with the value at entry $\mathcal{V}[a_{th}, b_{th}]$; this value is compared to the value at entry $\mathcal{V}[a_{th}, b_{th-1}]$. If these values are different, the algorithm has selected the variation in column b_{th}. Then, the next entry analyzed is $\mathcal{V}[(a_{th-p_{b_{th}}}, b_{th-1}]$, and the value in this entry is compared with the one of the previous column (same row). The process continues progressively until the algorithm reaches the beginning of the table (i.e., the upper left corner). In the example, in Fig. 5.9, the services S_1, S_3, and S_4 are processed.

$$\mathcal{V}[a,b] = \begin{cases} 0 & \text{if } n = 0 \text{ or } P_{ES} = 0, \\ max\{v_{b-1} + \mathcal{V}[a-p_{b-1}, b-1], \mathcal{V}[a, b-1]\} & \text{if } p_{b-1} < a, \\ \mathcal{V}[a, b-1] & \text{otherwise.} \end{cases} \tag{5.33}$$

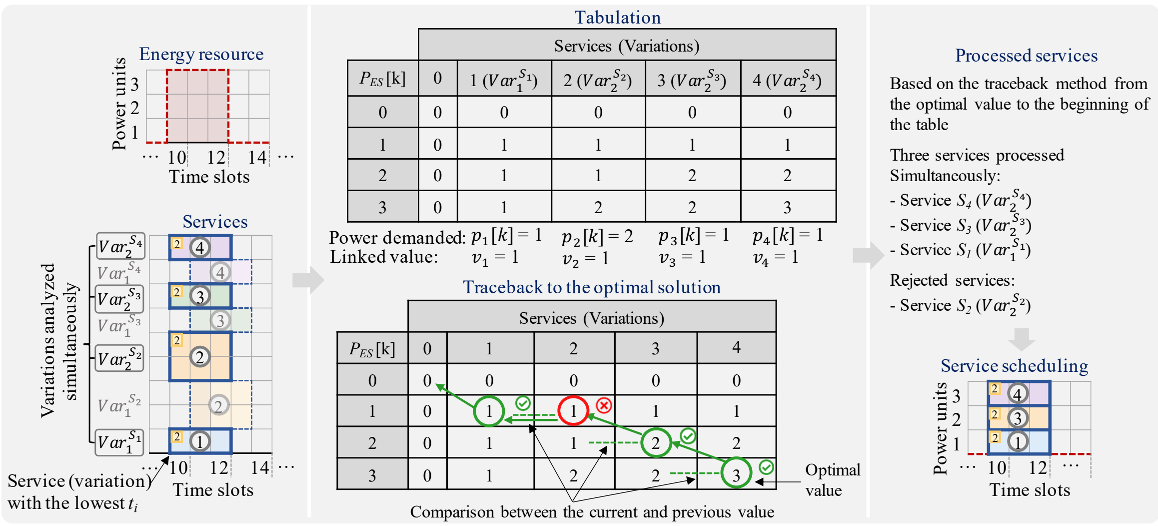


FIGURE 5.9: Example of application of the Dynamic Programming Algorithm for $N = 4$ services.

3. *Service scheduling:* The selected variations and all others beloging to the processed services (because a service can produce several variations) are removed from the $sortedVarList$. Then, the algorithm proceeds to reorder the remaining variations based on the increasing value of t_i. The selection of variations through PD is executed progressively until the energy resource is not available or until all variations and levels of priority have been explored. At the end the algorithm presents the metrics P_{RES}, AR, and P_{LACK}.

The literature has demonstrated that a tabulation-based DP can solve an MMKP in pseudopolynomial time because a solution boils down to filling in values in the DP table (using two nested for loops) and each row is typically computed in constant time. Even DP for MMKP can deliver a polynomial time if the number of items to be packed into the knapsack is small [110]. Consequently, the complexity in the proposed strategy DPTs is pseudopolynomial. Specifically, the growth rate of DPTs depends on the maximum values of N, u_i, and mainly on the progressive DP analysis of the set of concurrent variations $DPVar$, as shown in Eq. 5.34.

$$f(L, N, u_i) = N + ((2 \times N \times u_i + N) + N \times DPVar) \times L \tag{5.34}$$

5.3.0.4 Prepartitioning Strategy

Preliminary tests on the heuristic strategies (which are discussed in detail in Section 5.4) have demonstrated high-quality results in energy use and reduced running time compared to OptTsCost (optimal solution) with applicability to scenarios in the range of thousands of energy demands. However, to further scale up the applicability of energy management to IoT scenarios with dozens or hundreds of thousands of services in this section, we propose a complementary method that could be applied to OptTsCost, as well as to GreedyTs, GATs, and DPTs to improve their performance in terms of scalability. The proposed idea is the application of a prepartitioning method on services. Inspired by a divide-and-conquer approach, a well-known design technique proven to produce efficient solutions with little or no loss of accuracy [111], the proposed prepartitioning method aims to iteratively analyze smaller subsets of simultaneous services instead of the original set of energy demands. These subsets are then solved using algorithmic strategies (optimal or heuristics), and their partial solutions are combined to obtain adaptive energy management for the original problem.

The prepartitioning method applied to the proposed algorithmic strategies is explained in Fig. 5.10. In this method, the total number of partitions is denoted as $NumPart$ ($1 < NumPart \leq N$), $partition_z$ (subset of services, $partition_z \subset N$), which has a length $lenPart_z$. This length can be the same (or approximately the same) for all partitions, or it can be different for each partition depending on factors such as the priority of services, application scenario, or other specific objectives required in energy management. In any case, the reduction of search space in the partition domain contributes to reducing the complexity of the original strategy, either optimal or heuristic. Iteratively, each $partition_z$ is solved by the selected algorithmic strategy. All partitions are resolved by the same algorithm, although a hybrid strategy (e.g., the joint application of DP and GATs) may be considered in future work. Once all the partitions have been solved, or the P_{ES} has been allocated, the prepartitioning method delivers the scheduling (suboptimal) for all N services and the metrics P_{RES}, AR, and P_{LACK}. In summary, the application of the prepartitioning method on OptTsCost, GreedyTs, GATs, and DPTs originates four additional heuristic strategies that are identified in this paper as OptTsCostPart, GreedyTsPart, GATsPart, and DPTsPart, respectively. Regarding the complexity, Eq. 5.35 summarizes the growth rate related to the application of the prepartitioning method. In this expression, the third term is dominant and represents the cumulative complexity of solving all $NumPart$ partitions by the selected strategy.

$$f(L, N, u_i) = N + (2 \times N \times u_i + N) \times L + \sum_{z=1}^{NumPart} StratPart_z \tag{5.35}$$

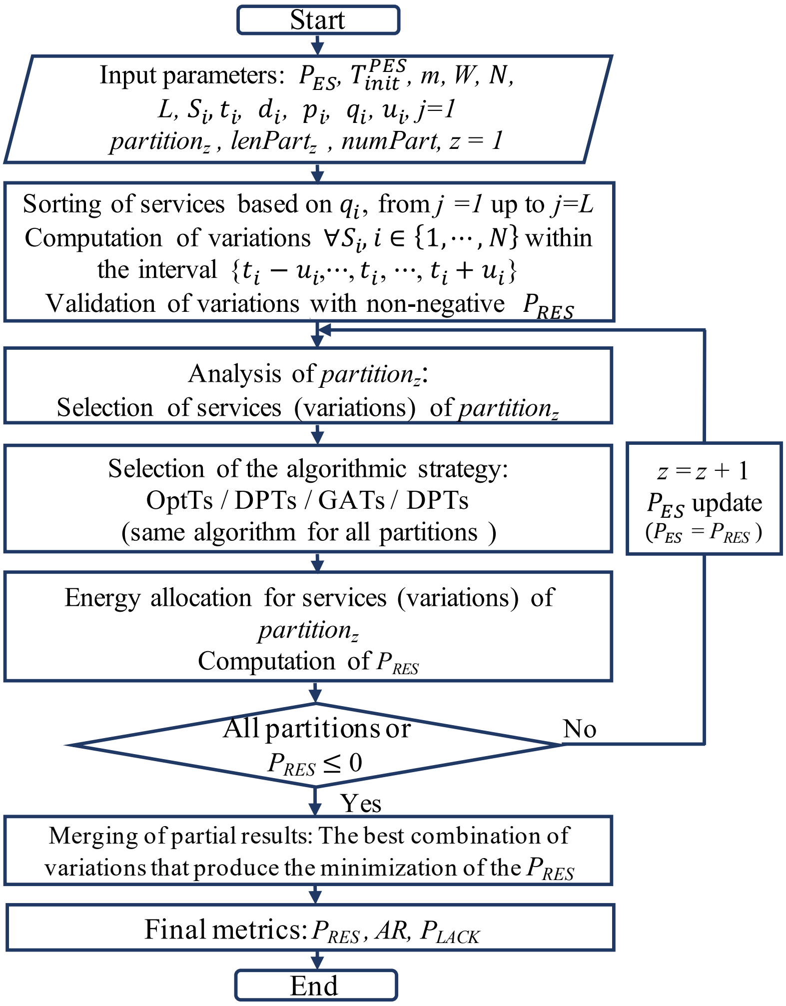


FIGURE 5.10: Flow chart of the prepartitioning method applied to the algorithmic strategies.

5.3.0.5 Adaptation of Algorithmic Strategies for Online Scenarios

Adaptive energy management can be implemented for offline or online approaches. In the offline approach, the service scheduling strategies, as shown in Section 5.2, Section 5.3.0.1, Section 5.3.0.2, and Section 5.3.0.3, know in advance all generation and consumption parameters, and they are capable of performing both backward and forward time-shifting on services. The offline approach can be used to plan the distribution of energy resources, reshape the load profile (e.g., reduce peak loads), and prioritize the use of renewable energy sources, which can produce a reduction of overall operational cost and carbon emission levels and promote sustainability in the generation and consumption ecosystem [9]. Instead, in the online approach, the service scheduling algorithmic strategies have no future information about generation and consumption; the services are processed as time evolves, and only the forward time-shifting can be applied to services. The online approach represents the real-time dynamic of provisioning and consumption in which adaptive energy management must be performed. In this regard, this section presents the online version of the proposed service scheduling strategies, including the prepartitioning method. Fig. 5.11 explains the generic algorithm for adapting the developed service scheduling strategies (optimal and heuristics) for online applications. The main steps and additional features performed are described below.

1. *Initial analysis of services:* The online implementation of service scheduling strategies starts with the differentiation of the energy resources for the processing of CS (P_{CS}) and NCS (P_{NCS}). The proposed strategy assumes that once the service is accepted (in its first slot), there is P_{ES} for its completion (the model does not accept fractional processing, as discussed in Section 5.1.1). An additional feature for the online approach is the inclusion of a list named *waitingList*, which stores information on the variations of services that were not processed in their original starting time (t_i). This is due to energy allocation to higher-priority service(s).

2. *Analysis for CS:* If service S_i at time slot k is identified as a CS, the strategy allocates the demanded energy resource. Later, the strategy updates the P_{CS} for the rest of CS.

3. *Analysis for NCS:* If service S_i at time slot k is identified as an NCS, the strategy performs a similar analysis as for the offline approach, considering the variation at time slot k, the simultaneous variations with the variation in analysis, and the variations in *waitingList*. In this regard, a variation is considered simultaneously if it exists within the lifetime (d_i) of the analyzed variation (service) at time slot k. Once the strategy selects the best combination/variation, the energy allocation is made, and the energy resources for subsequent NCS are updated.

4. *Final metrics:* After analyzing all services at the time horizon W, the strategy delivers the performance metrics.

5.4 Evaluation

This section evaluates the performance of both the energy model and the proposed algorithmic strategies through extensive simulations. Different generation and consumption profiles and several scenarios have been used to show the benefits of energy use in terms of proposed metrics and the applicability of developed algorithmic strategies.

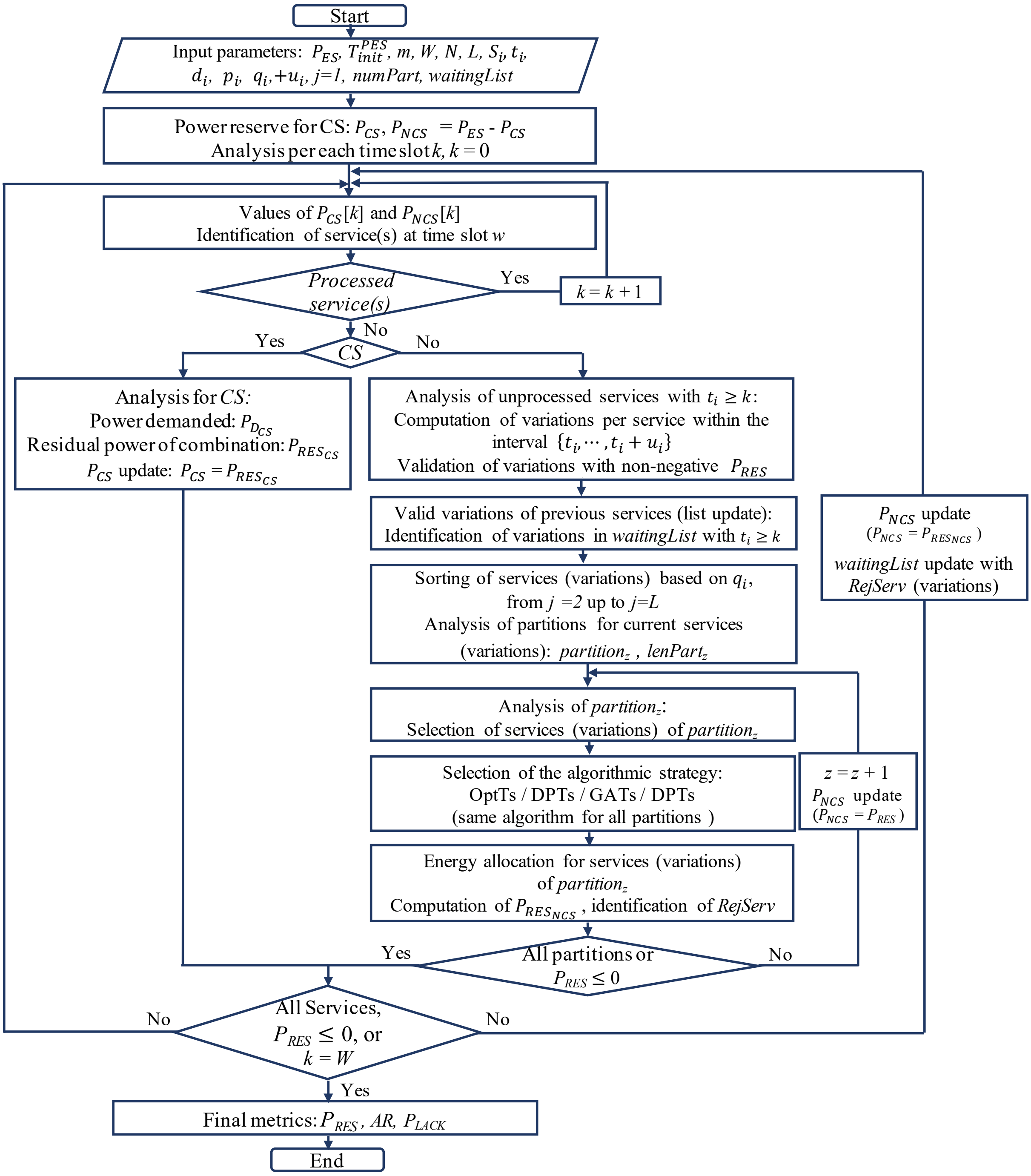


FIGURE 5.11: Flow chart of the adaptation of the service scheduling strategies for an online approach.

5.4.0.1 Simulation setup

The algorithmic strategies have been implemented on Matlab R2018b and running on a machine with a 3.33 GHz ×12 cores Intel Core i7 Extreme processor and 24 GB RAM. The simulations leverage parallel processing with the concurrent use of up to 6 cores. The results obtained of metrics P_{RES}, AR, and P_{LACK} are compared to the benchmark scenario in which no management mechanism is applied (i.e., no time-shifting, prioritization, or rejection) [112].

5.4.0.2 Description of Generation and Consumption Profiles for Simulated Scenarios

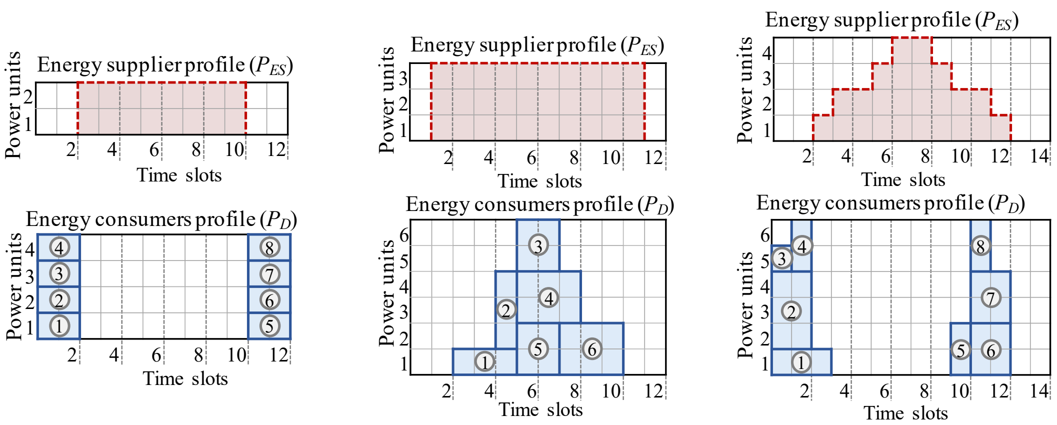
- *Generation and consumption profiles:* To analyze the performance of the proposed algorithmic strategies in different generation and consumption conditions, four profiles have been considered. In these profiles, the total available energy is equal to the total energy demand. The profiles are summarized in Fig. 5.12 and Fig. 5.13a and are described below.

  - *Profile I:* This profile allows the analysis of the performance of service scheduling strategies in total desynchronization periods of energy supply and consumption (i.e., during periods of scarcity and abundance of power), as shown in Fig. 5.12a. In this case, the P_{ES} is consumed only if time-shifting is applied to services. Moreover, in Profile I and Profile II, a flat-supply profile (representing a realistic generation scenario, as studied in [113]) has been chosen for simplicity in the analysis. However, the algorithmic strategies have no restrictions working with any demand and supply profile if needed.
  - *Profile II:* This profile simulates a peak demand due to high load, as shown in Fig. 5.12b. Moreover, in this profile, services with random values of p_i and d_i have been considered to simulate a more realistic consumption scenario.
  - *Profile III:* This profile allows for the analysis of the performance of scheduling strategies in a futuristic environment 100% powered by green energy sources. This is a very promising approach to tackle sustainability issues, increasing carbon emissions due to generation and cosumption of non-renable energy sources, and is an important requirement for deploying the IoE [12]. In this profile, as shown in Fig. 5.12c, the services have random values of p_i and d_i and are partially desynchronized with the P_{ES}. Moreover, the supply follows a Gaussian distribution to simulate the renewable generation patterns (e.g., through photovoltaic panels), as shown in [114].
  - *Profile IV:* This profile allows for the analysis of the application of the heuristic scheduling strategies in a HEMS. Figure 5.13a shows the supply and consumption profiles for the simulated HEMS. The consumption of 20 services (appliances) is adapted from [115], constrasted with the data in [116], and summarized in Table 5.1.

- *Scenario description:* Using the profiles described above, seven application scenarios have been analyzed, summarized in Table 5.2. According to N, because this parameter is directly related to consumption and has a direct impact on complexity, as analyzed in Section 5.2 (Eq. 5.30), the scenarios are grouped in three categories: (i) small-scale scenarios, for $N \leq 20$ services in which both optimal and heuristic strategies have been analyzed, except for the HEMS in which only heuristics are

TABLE 5.1: Description of consumption for a HEMS scenario.

Load description	Quantity	p_i[W]	t_i[Hour]	d_i[Hour]	q_i
Freezer	1	210	0	24	1
Refrigerator	1	650	0	24	1
Oven	1	1800	16	3	3
Lighting	9	25	17	7	1
TV	1	140	18	5	1
Laptop	1	90	17	5	1
PC	1	140	18	6	1
Vacuum Cleaner	1	600	19	3	1
Water Heater	1	2000	17	6	2
Air-Conditioner	1	1280	14	7	2
Washing Machine	1	1350	19	3	3
Dishwasher	1	1250	18	3	3



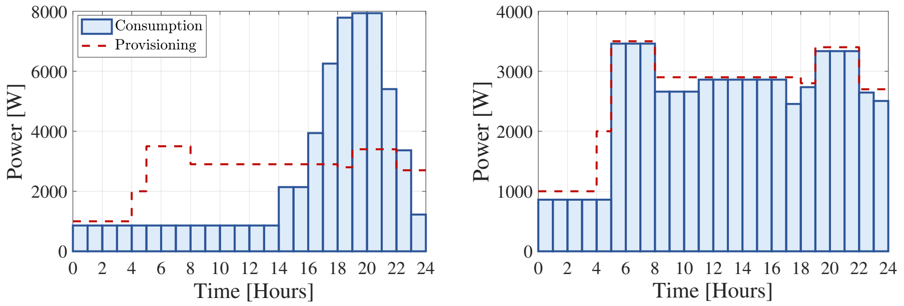
(A) Profile I. (B) Profile II. (C) Profile III.

FIGURE 5.12: Power supply and consumption profiles for simulated scenarios of service scheduling strategies.

applied; (ii) large-scale scenarios for $20 \leq N \leq 10^4$ services in which only heuristics are evaluated, including a online scenario for $N = 10^2$ services (i.e., other scenarios in Table 5.2 are offline); and (iii) very large-scale scenarios for $10^4 \leq N \leq 10^6$ services in which only the prepartitioned versions of *GreedyTs* and *DPTs* are analyzed. In all scenarios, the generation conditions are adapted to the consumption of N services ($P_{ES} \geq P_D$). Moreover, in scenarios in which random values (e.g., p_i and d_i) or conditions (e.g., in GATs) are used, the simulations have been repeated 20 or 50 times considering a confidence interval of 95%, to ensure stability of results.

5.4.0.3 Analysis of Results in Small-scale Scenarios

This section presents the evaluation of the algorithmic scheduling strategies, both optimal and heuristics, in small-scale scenarios (first four scenarios in Table 5.2) based on the results of performance metrics (mainly AR), running time, and RAM and CPU usage in the simulation domain. To obtain the prepartitioned version of OPTTSCOST, GREEDYTS, GATS, and DPTS (e.g., OPTTSCOSTPART or GREEDYTSPART) two partitions have used. Moreover, the results from heuristics are compared with the optimal bounds deliver by OPTTSCOST.



(A) Profile IV: Profiles before the application of service scheduling strategies.

(B) Power supply and consumption profiles after the application of GATs.

FIGURE 5.13: Profile IV: Power supply and consumption profiles in the HEMS. Parameters: According to Table 5.1

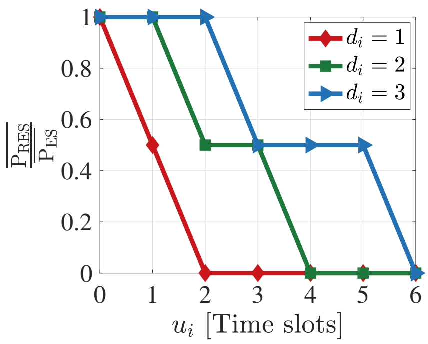
TABLE 5.2: Summary of simulation parameters.

Scenario	W	N	Size	Profile	d_i	p_i	q_i	$max\{u_i\}$
I	12	8	Small	I	1-3 $\forall S_i$	1 $\forall S_i$	1 $\forall S_i$	$\pm 6 \forall S_i$
II	12	6	Small	II	Uniform distribut. random value[1-3], $\forall S_i$	Uniform distribut. random value[1-3], $\forall S_i$	1 $\forall S_i$	$\pm 4 \forall S_i$
III	14	8	Small	III	Uniform distribut. random value[1-3], $\forall S_i$	Uniform distribut. random value[1-3], $\forall S_i$	1 $\forall S_i$	$\pm 4 \forall S_i$
IV (HEMS)	24	20	Small	IV	According to Table 5.1			
V	24	10^2 10^3 10^4	Large	III adapted to N and W	Uniform distribut. random value[1-3], $\forall S_i$	Uniform distribut. random value[1-3], $\forall S_i$	1 $\forall S_i$	$\pm 10 \forall S_i$
VI (Online)	24	10^2	Large	III adapted to N and W	Uniform distribut. random value[1-3], $\forall S_i$	Uniform distribut. random value[1-3], $\forall S_i$	1 $\forall S_i$	$+10 \forall S_i$
VII	24	10^5 10^6	Very Large	III adapted to N and W	Uniform distribut. random value[1-3], $\forall S_i$	Uniform distribut. random value[1-3], $\forall S_i$	1 $\forall S_i$	$\pm 10 \forall S_i$

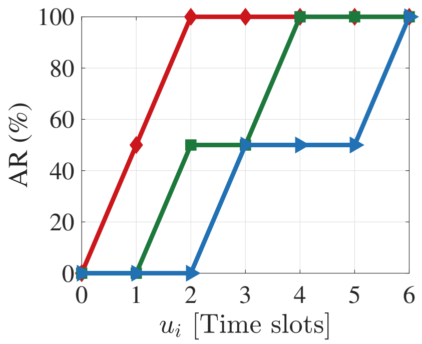
Figure 5.14 and Table 5.3 show the simulation results deliver by optimal and heuristics strategies in Scenario I. While, Fig. 5.15, Fig. 5.16, and Fig. 5.18 present the evaluation of OptTsCost, GreedyTs, GATs, and DPTs in Scenarios II, III and IV, respectively. The simulation results report that as the value of u_i increases, decreases the values of P_{RES} (e.g., in Fig. 5.14a) and P_{LACK} (e.g., in Fig. 5.14c) and increases the value of AR (e.g., in Fig. 5.14b). These values indicate that through the use of management mechanisms such as time-shifting, the algorithmic strategies can adapt energy demands to availability, being able to make an optimal use/consumption (e.g., 100% in Scenario I) of the energy produced (i.e., P_{RES} minimization) while allowing the processing of services (ECs) that under normal conditions (i.e., without management mechanisms) would be rejected. Consequently, the simulation results indicate that applying the proposed energy model through the algorithmic strategies such as OptTsCost promotes better use and distribution of P_{ES}, and potential reduction in peak consumption and energy costs.

Because the metric AR is a direct indicator of energy utilization and service processing, and for practicality in the presentation of results, only the maximum value of this metric (i.e., using the $max\{u_i\}$, e.g., $max\{u_i\} = 4$ in Scenario II) is presented for strategies with prepartitioning. Table 5.4 summarizes the values of AR for all optimal and heuristics strategies in small-scale scenarios. The maximum AR values (Final AR) obtained by the algorithmic strategies in all scenarios show improvements (AR Gain) over 15% (i.e., 15% more services processed) compared to the baseline case (if $u_i = 0$). Depending on the scenario and the strategy used, the improvements can reach 100%, such as in DPTs in Scenario I. Table 5.4 reports that the heuristic strategy that delivers the best AR gains in all scenarios is GATs (using in all cases a number less than 10 generations), with values that are the same or very similar to those obtained with OptTsCost (whose results are optimal). Moreover, Table 5.4 shows a minimum difference between the values of AR produced by the original strategies and their version with prepartitioning; in the worst case for GATsPart in Scenario II, this difference with GATs is less than 11%.

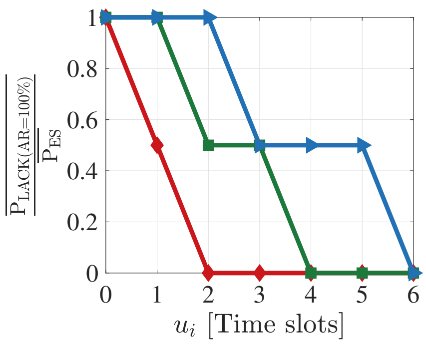
For the evaluation in a HEMS, only heuristic approaches have been considered due to the complexity of OptTsCost for values of $N > 8$ services or of $u_i > 6$ time slots. This scenario also allows us to analyze the performance of the heuristic strategies for energy demands with different priority levels, as shown in Table 5.1. Regarding the AR value reached in this scenario, although the percentage of improvement is lower than the rest of the scenarios in Table 5.4 (between 15% to 20%), the adaptation of consumption patterns to P_{ES}, by the heuristic strategies, allows a reduction of peak power by more than 55% (from 8000W down to 3500 W, as shown in Fig. 5.13b) and obtain an $AR = 100\%$ such as reported the results for GATs.



(A) Scenario I: P_{RES} OptTsCost.



(B) Scenario I: AR OptTsCost.

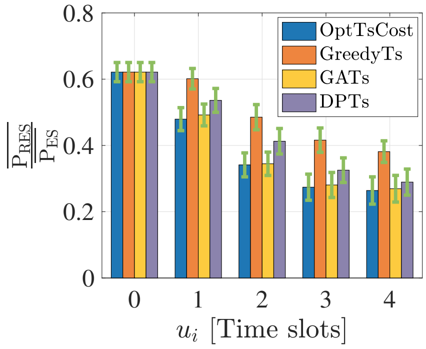


(C) Scenario I: P_{LACK} OptTsCost.

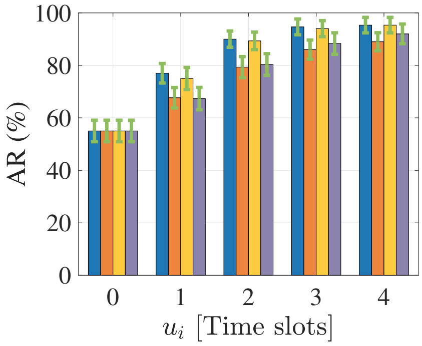
Figure 5.14: Performance evaluation of OptTsCost in Scenario I.
Parameters: According to Table 5.2.

TABLE 5.3: Performance evaluation of optimal and heuristic strategies in Scenario I.

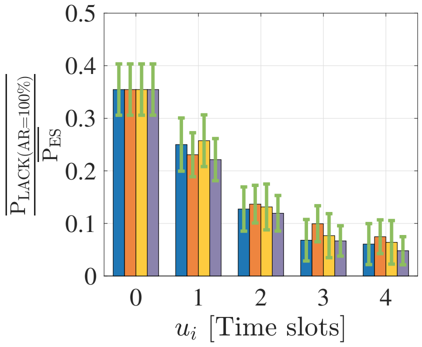
Metric	u_i	$d_i = 1$				$d_i = 2$				$d_i = 3$			
		OptTsCost	GreedyTs	GATs	DPTs	OptTsCost	GreedyTs	GATs	DPTs	OptTsCost	GreedyTs	GATs	DPTs
P_{RES}	0	1	1	1	1	1	1	1	1	1	1	1	1
	1	0.50	0.50	0.50	0.50	1	1	1	1	1	1	1	1
	2	0	0	0	0	0.50	0.50	0.50	0.50	1	1	1	1
	3	0	0	0	0	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
	4	0	0	0	0	0	0	0	0	0.50	0.50	0.50	0.50
	5	0	0	0	0	0	0	0	0	0.50	0.50	0.50	0.50
	6	0	0	0	0	0	0	0	0	0	0	0	0
AR	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	50	50	50	50	0	0	0	0	0	0	0	0
	2	100	100	100	100	50	50	50	50	0	0	0	0
	3	100	100	100	100	50	50	50	50	50	50	50	50
	4	100	100	100	100	100	100	100	100	50	50	50	50
	5	100	100	100	100	100	100	100	100	50	50	50	50
	6	100	100	100	100	100	100	100	100	100	100	100	100
P_{LACK}	0	1	1	1	1	1	1	1	1	1	1	1	1
	1	0.50	0.50	0.50	0.50	1	1	1	1	1	1	1	1
	2	0	0	0	0	0.50	0.50	0.50	0.50	1	1	1	1
	3	0	0	0	0	0.50	0.25	0.50	0.25	0.50	0.50	0.50	0.50
	4	0	0	0	0	0	0	0	0	0.50	0.33	0.50	0.33
	5	0	0	0	0	0	0	0	0	0.50	0.17	0.50	0.17
	6	0	0	0	0	0	0	0	0	0	0	0	0



(A) Scenario II: P_{RES} optimal and heuristics.

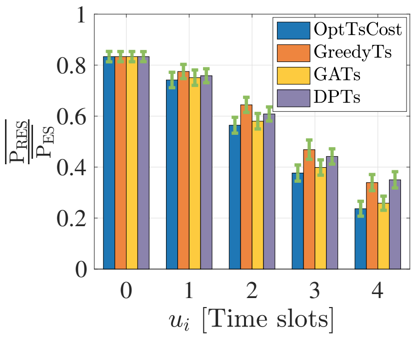


(B) Scenario II: AR optimal and heuristics.

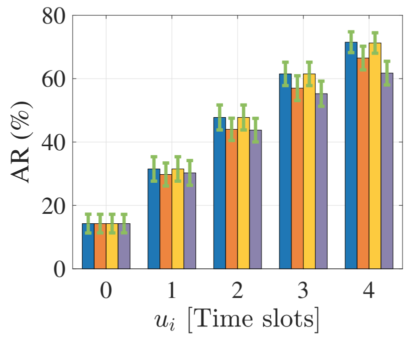


(C) Scenario II: P_{LACK} optimal and heuristics.

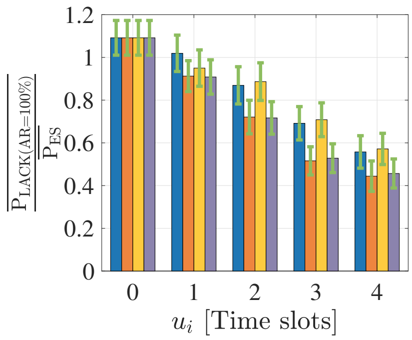
FIGURE 5.15: Performance evaluation of optimal and heuristic strategies in Scenario II. Parameters: According to Table 5.2.



(A) Scenario III: P_{RES} optimal and heuristics.



(B) Scenario III: AR optimal and heuristics.



(C) Scenario III: P_{LACK} optimal and heuristics.

FIGURE 5.16: Performance evaluation of optimal and heuristic strategies in Scenario III. Parameters: According to Table 5.2.

To quantitatively evaluate the difference, in terms of AR, between the optimal strategy and the heuristics, the criterion of *approximation ratio* (ρ) has been adopted [117]. This parameter estimates how many times bigger the approximate result is compared to the optimal solution. Adapted to the conditions of the proposed energy model, the ρ parameter is defined by:

$$\rho = 1 - \frac{1}{max\{u_i\}} \sum_{b=1}^{max\{u_i\}} \frac{|Opt_b - SubOpt_b|}{disOpt_b} \tag{5.36}$$

$$disOpt_b = \begin{cases} 1 & \text{if } Opt_b = Opt_0, \\ |Opt_b - Opt_0| & \text{if } Opt_b \neq Opt_0. \end{cases} \tag{5.37}$$

where, the first term in Eq. 5.36 represents the optimal solution, while the second term corresponds to the mean absolute error of all time-shifting values, except for $u_i = 0$. In Eq. 5.36, each absolute error b is weighted to the $disOpt_b$ parameter (Eq. 5.37), which represents the maximum distance between the optimal value (Opt_b) and the baseline value (Opt_0, when $u_i = 0$), to obtain the proportional error of each time-shifting b. The ρ parameter ranges from 0 to 1, and this latter is produced if the optimal and suboptimal values are equal. An intermediate ρ value represents the similarity or closeness factor to the optimal solution (e.g., DPTs $=$ OptTsCost $\times$ ρ). For a better understanding of ρ factor, Fig. 5.17 presents an example for DPTs in Scenario II. Eq. 5.38, shows the analytical computation.

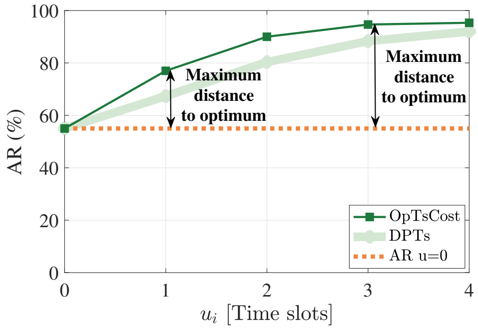
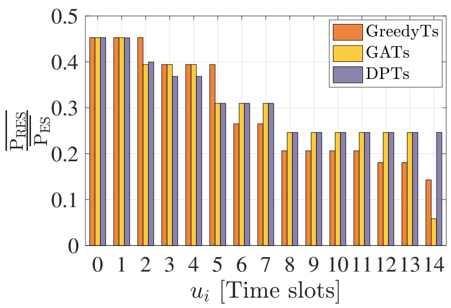


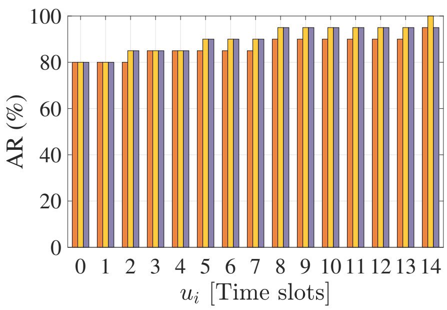
FIGURE 5.17: Difference of AR values of OPTTSCOST and DPTs in Scenario II.

$$\rho_{AR} = 1 - \frac{1}{4} \times \left(\frac{77-67.3}{77-55} + \frac{90-80.3}{90-55} + \frac{94.6-88.3}{94.6-55} + \frac{95.3-92}{95.3-55} \right) = 0.76 \tag{5.38}$$

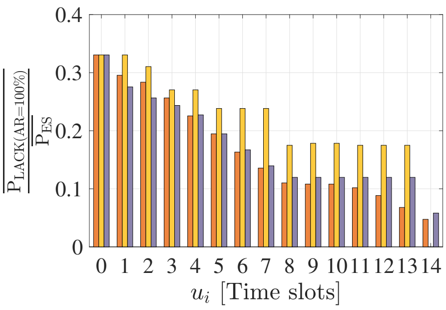
The result in Eq. 5.38 shows that DPTs is similar to OPTTSCOST in a factor equal to 0.76 (76% similarity), or that DPTs is able to produce a solution that is within $\sim 1.3\times$ the optimal result. Table 5.5 summarizes the ρ factors for small - scale scenarios and reveal that the heuristics strategies produces near-optimal or optimal solutions and a stable performance. In the worst case ($\rho_{AR} = 0.70$ for DPTs in Scenario II), the heurisctic strategy is within only $\sim 1.4\times$ the optimal solution.



(A) Scenario IV (HEMS): P_{RES} heuristics.



(B) Scenario IV (HEMS): AR heuristics.



(C) Scenario IV (HEMS): P_{LACK} heuristics.

FIGURE 5.18: Performance evaluation of optimal and heuristic strategies in Scenario IV. Parameters: According to Table 5.2.

TABLE 5.4: Maximum value of AR achieved by algorithmic strategies in small-case scenarios.

Scenario	No Strategy	OptTsCost		OptTsCostPart		GreedyTs		GreedyTsPart		GATs		GATsPart		DPTs		DPTs Part	
	Initial AR	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain
I	0	**100**	**100**	100	100	100	100	100	100	100	**100**	100	100	100	100	100	100
II	55	**95.33**	**40.33**	94	39	89	34	88.66	33.66	95.33	**40.33**	91	36	92	37	89.66	34.66
III	14.25	**71.5**	**57.25**	71.5	57.25	66.50	52.25	66.50	52.25	71.25	**57**	70	55.75	61.75	47.5	61.75	47.5
IV	80	-	-	-	-	95	15	95	15	100	**20**	95	15	95	15	95	15

TABLE 5.5: Approximation ratio of heuristic strategies in small-scale scenarios.

Strategy	Scenario I			Scenario II	Scenario III
	d_1	d_2	d_3	$\overline{d_i, p_i}$	$\overline{d_i, p_i}$
GREEDYTS	1	1	1	0.72	0.90
GATS	1	1	1	0.97	1
DPTS	1	1	1	0.76	0.88
OPTTSCOSTPART	1	1	1	0.97	1
GREEDYTSPART	1	1	1	0.71	0.90
GATSPART	1	1	1	0.76	0.99
DPTSPART	1	1	1	0.70	0.87

Regarding the running time, the evaluation results for the original strategies and those adapted to the prepartitioning method are summarized in Table 5.6 and Table 5.7, respectively. The results in Table 5.6 and Table 5.7 report that the heuristic strategies are executed (with a relative gain in time G_R) between two and seven orders of magnitude faster than OPTTSCOST. Moreover, the running time of the prepartitioned versions are slightly higher than the original verisons (i.e., of GREEDYTS, GATS, and DPTS), this due to the iterative process of the partitions and the subsequent union of partial solutions. This condition indicates that for small-scale scenarios, in terms of running time, it is preferable to apply the original heuristic strategies instead of their versions with prepartitioning.

To better describe the difference in running time and computational resources used by optimal and heuristics strategies, we have performed the analysis for Scenario I considering only the maximum value of time-shifting (i.e., $u_i = 6\,\forall S_i$) and varying the services (i.e., $1 \leq N \leq 8$). The evaluation results in Fig. 5.19 for a sigle iteration report that: (i) the running time of the heuristics is at least two orders of magnitude less than OPTTSCOST (see Fig. 5.19a, as indicated in Table 5.6 and Table 5.7, (ii) that the use of the RAM of the heuristic strategies is between 2% and 3% of the amount used by OPTTSCOST (see Fig. 5.19b), and (ii) that the CPU usage of the heuristics is between 4% and 40% of the resource used by OPTTSCOST (see Fig. 5.19c).

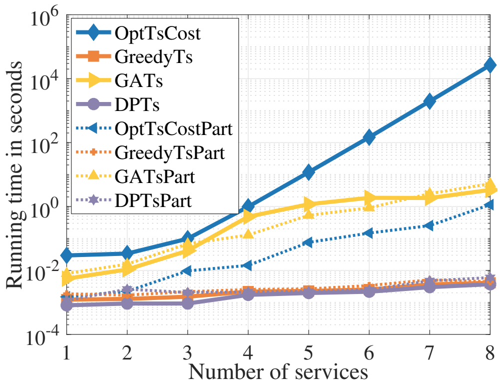
In summary, the simulation results in this section report that the original and prepartitioned heuristic strategies produce high-quality solutions that outperform the optimal solution in terms of running time, and RAM and CPU usage. This features enable that heuristic strategies when deployed in NFV domain (or a similar computing facility e.g., a HEMS or a fog computing domain) can be applied for adaptive energy management in small-scale scenarios.

TABLE 5.6: Running time in seconds of algorithmic strategies and G_R of heuristics concerning OPTTSCOST in small-case scenarios.

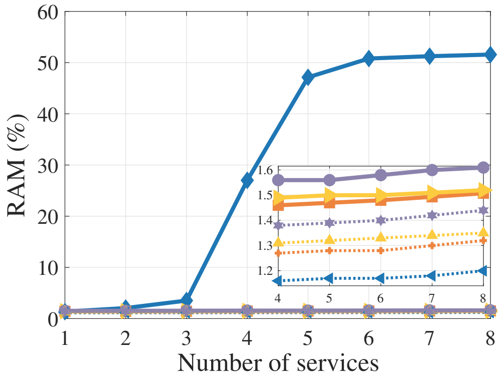
Scenario	**OptTsCost**	**GreedyTs**		**GATs**		**DPTs**	
	Running time	Running time	G_R	Running time	G_R	Running time	G_R
I	1.88×10^6	2.37×10^{-1}	$\sim 7.93 \times 10^6$x	8.10×10^0	$\sim 1.33 \times 10^5$x	1.69×10^{-1}	$\sim 1.11 \times 10^7$x
II, 50 iterations	2.89×10^4	8.72×10^0	$\sim 3.31 \times 10^3$x	1.93×10^2	$\sim 1.50 \times 10^2$x	8.82×10^0	$\sim 3.27 \times 10^3$x
III, 50 iterations	1.41×10^6	7.39×10^0	$\sim 1.90 \times 10^5$x	2.07×10^2	$\sim 6.81 \times 10^3$x	7.17×10^0	$\sim 1.97 \times 10^5$x
IV (HEMS)	—	2.96×10^{-1}	—	2.10×10^1	—	9.69×10^{-1}	—

TABLE 5.7: Running time in seconds of algorithmic strategies considering prepartitioning and G_R concerning OPTTSCOST in small-case scenarios.

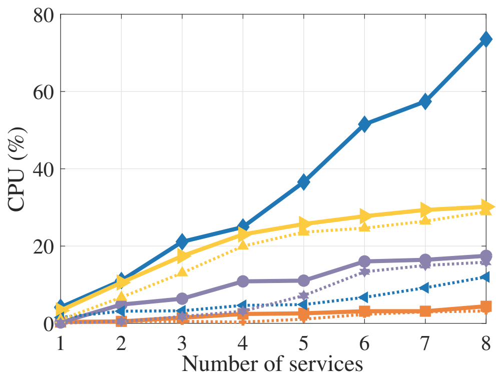
Scenario	**OptTsCost**	**OptTsCostPart**		**GreedyTsPart**		**GATsPart**		**DPTsPart**	
	Running time	Running time	G_R	Running time	G_R	Running time	G_R	Running time	G_R
I	$1.88 \cdot 10^6$	$3.76 \cdot 10^1$	$\sim 5.00 \cdot 10^4$x	$8.62 \cdot 10^{-1}$	$\sim 2.18 \cdot 10^6$x	$4.28 \cdot 10^1$	$\sim 4.39 \cdot 10^4$x	$8.74 \cdot 10^{-1}$	$\sim 2.15 \cdot 10^6$x
II, 50 iterations	$2.89 \cdot 10^4$	$2.15 \cdot 10^1$	$\sim 1.34 \cdot 10^3$x	$1.14 \cdot 10^1$	$\sim 2.54 \cdot 10^3$x	$8.82 \cdot 10^0$	$\sim 3.27 \cdot 10^3$x	$1.42 \cdot 10^1$	$\sim 2.04 \cdot 10^3$x
III, 50 iterations	$1.41 \cdot 10^6$	$1.74 \cdot 10^2$	$\sim 8.10 \cdot 10^3$x	$1.04 \cdot 10^1$	$\sim 1.36 \cdot 10^5$x	$7.17 \cdot 10^0$	$\sim 1.97 \cdot 10^5$x	$1.09 \cdot 10^1$	$\sim 1.29 \cdot 10^5$x
IV (HEMS)	—	—	—	$3.38 \cdot 10^{-1}$	—	$9.69 \cdot 10^{-1}$	—	$5.94 \cdot 10^{-1}$	—



(A) Running time of algorithmic strategies in Scenario I, for $u_i = 6\,\forall S_i$, varying N, and using a single core.



(B) RAM usage of algorithmic strategies in Scenario I, for $u_i = 6\,\forall S_i$, varying N, and using a single core.

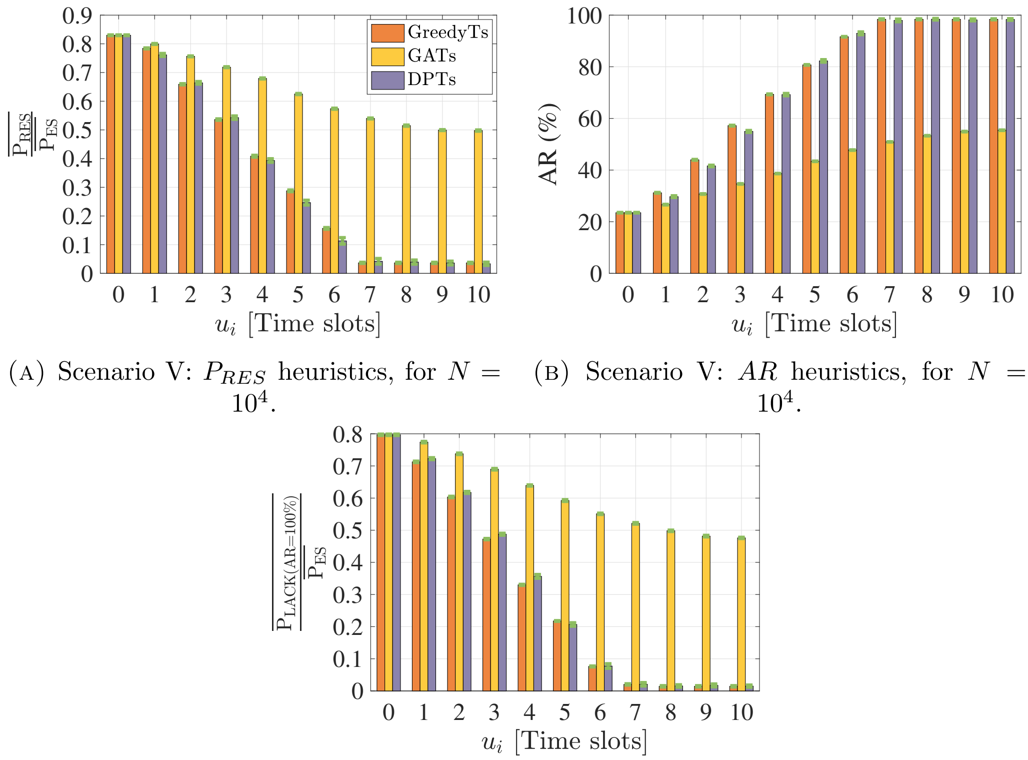


(C) CPU usage of algorithmic strategies in Scenario I, for $u_i = 6\,\forall S_i$, varying N, and using a single core.

FIGURE 5.19: Example of performance evaluation of algorithmic strategies according to the running time and RAM and CPU usage (single core) in small-scale scenarios.

5.4.0.4 Analysis of Results in Large and Very-large Scenarios

This section presents the evaluation of the heuristic strategies in large-scale and very-large-scale scenarios (scenarios V, VI, and VII in Table 5.2) based on the results of performance metrics (mainly AR), running time, and RAM and CPU usage in the simulation domain. To obtain the prepartitioned versions of heuristic strategies, the number of partitions in each scenario and for each value of N is such that the length of each partition is equal to 10 services (e.g., for DPTsPart with $N = 10^4$ services, the $NumPart = 10^3$ partitions). This length has been chosen to produce equal-sized partitions (i.e., all partitions of 10 services). Moreover, based on results in small-scale scenarios, partitions of this length (e.g., in Scenario IV) have demonstrated to produce high-quality solutions. For practicality in the presentation of results, this section only shows the evaluation of the metrics P_{RES}, AR, and P_{LACK} for a single value of N in each scenario (e.g., $N = 10^4$ for Scenario V, $N = 10^3$ for Scenario VI, and $N = 10^6$ for Scenario VII). However, a summary of the evaluation of heuristics for all cases (i.e., all values of N), in terms of AR, running time, and computational capacity usage, is presented in the corresponding figures and tables. Fig. 5.20, Fig. 5.21, Fig. 5.22, Table 5.8, and Table 5.9 show the simulation results produced by heuristics strategies in large-scale scenarios. Particularly, the results in Fig. 5.20, Fig. 5.21, and Fig. 5.22 report that as the value of u_i increases, decreases the values of the values of P_{RES} (see Fig. 5.20a) and P_{LACK} (see Fig. 5.20c) and increases the value of AR (see Fig. 5.20b). These values indicate that the proposed energy model enables efficient adaptive energy management in large-scale scenarios.

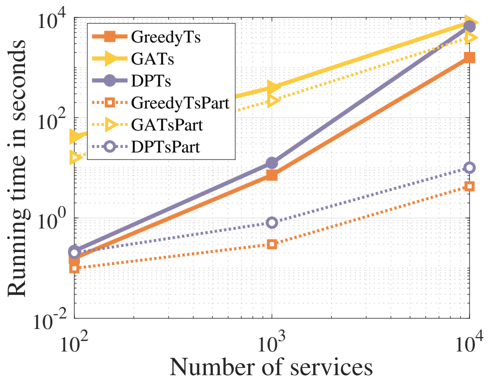


(A) Scenario V: P_{RES} heuristics, for $N = 10^4$.

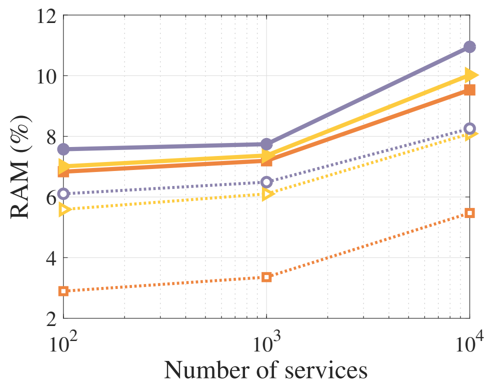
(B) Scenario V: AR heuristics, for $N = 10^4$.

(C) Scenario V: P_{LACK} heuristics, for $N = 10^4$.

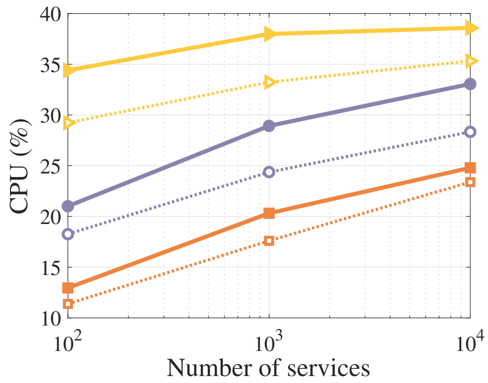
FIGURE 5.20: Example of performance evaluation of heuristic strategies for Scenario V (large-scale) according to P_{RES}, AR, and P_{LACK}.



(A) Running time of heuristic strategies in Scenario V, for $u_i = 10\,\forall S_i$, varying N, and using a single core.



(B) RAM usage of heuristic strategies in Scenario V, for $u_i = 10\,\forall S_i$, varying N, and using a single core.

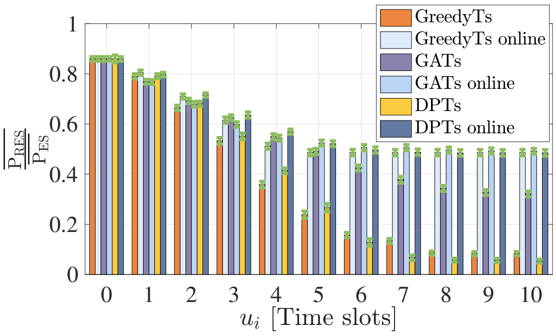


(C) CPU usage of heuristic strategies in Scenario V, for $u_i = 10\,\forall S_i$, varying N, and using a single core.

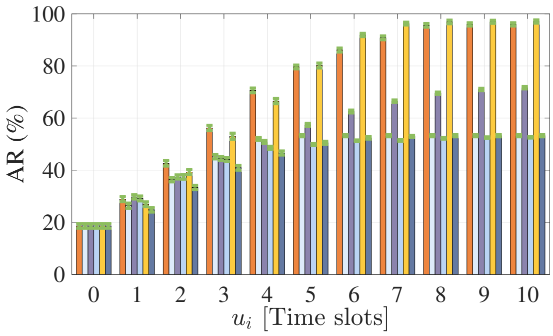
FIGURE 5.21: Example of performance evaluation of heuristic strategies for Scenario V (large-scale) according to Running time, RAM Usage, and CPU Usage.

Table 5.8 summarizes the values of AR and AR gain achieved by heuristics strategies, both the original and prepartitioned versions. These results demonstrate that heuristics deliver improvements in services processing (energy use) ranging from 31% (e.g., GATs in Scenario V for $N = 10^4$) up to 79% (e.g, GreedyTs in Scenario V for $N = 10^4$). The best values of AR in Table 5.8 are obtained by strategies GreedyTs and DPTs, while GATs produces the smallest improvements in all cases. Unlike the near-optimal solutions generated by GATs in small-scale scenarios, in large-scale scenarios, this strategy in its original version has a degraded performance due to the small size of the population ($\mathcal{P}_{\mathcal{GA}} = 1500$ chromosomes for all scenarios), compared to problem size (especially if $N \geq 10^3$). This shortcoming can be solved by proportionally increasing the $\mathcal{P}_{\mathcal{GA}}$, although this modification would cause an increase in complexity, as well as of running time and computational capacity demanded. According to the simulation results in Table 5.8, we observe that the low performance of GATs, in terms of AR, is overcomed if the prepartitioning method is applied to the strategy (i.e., GATsPart produces better values of AR than GATs). This is because at the particion domain a greater search space is available for obtaining better combinations and consequenlty better quality solutions. The values of AR in Table 5.8 (for offline approaches) show that GATsPart outperforms

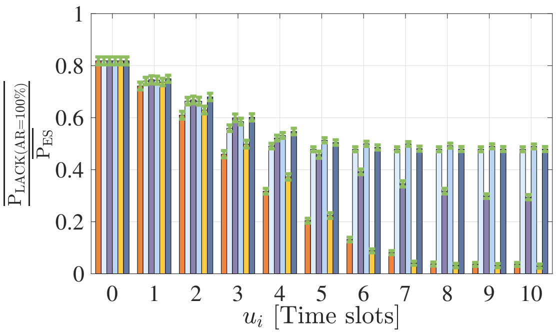
GATs by an average of 35%. Whereas the partitioned versions of GreedyTs and DPTs produce values of AR very similar and even the same as their original versions (e.g., $AR = 98.01$ for DPTs and DPTsPart in Scenario V for $N = 10^4$).



(A) Scenario VI (online): P_{RES} heuristics, offline and online approaches for $N = 10^2$.



(B) Scenario VI (online): AR heuristics, offline and online approaches for $N = 10^2$.



(C) Scenario VI (online): P_{LACK} heuristics, offline and online approaches for $N = 10^2$.

Figure 5.22: Example of performance evaluation of heuristic strategies for Scenario VI (large-scale).

The results of metrics P_{RES} (Fig. 5.22a), AR (Fig. 5.22b), and P_{LACK} (Fig. 5.22c) in Scenario VI reveal that heuristic strategies can be applied for adaptive energy management in online approaches. In this scenario, the improvements obtained, in terms of AR as shown in Table 5.8, are on average approximately 30% and these values, as expected, are lower than those obtained in the offline approach (approximately half), because the algorithms are limited to the use of forward time-shifting. In this scenario, the performance of all the strategies including the prepartitioned ones is similar, the best values of AR are obtained by GreedyTs and DPTs while the worst AR metric is generated by GATsPart, although the difference in the results is less than 1% (eg., $AR = 34.60\%$ for DPTs and $AR = 33.96\%$ for GATsPart). A feature that can be analyzed in future work is the incorporation of forecasting methods of energy supply in the algorithmic strategies to improve the service scheduling and, consequently, the AR metric.

Table 5.9 shows that the running time of the heuristic strategies and the G_R computed from the ratio between the heuristics and their prepartitioned versions. These results report that the partitioned strategies are executed in less time than the original versions with a difference of up to three orders of magnitude (e.g., DPTsPart in Scenario V for $N = 10^4$). The lowest running time values are obtained by GreedyTsPart and DPTsPart, which reveals their potential applicability to larger scenarios (i.e., for $N > 10^4$). To better describe the difference in running time and computational resources used by heuristics strategies, we present the analysis for Scenario V considering only the maximum value of time-shifting (i.e., $u_i = 10\,\forall S_i$) and varying the services (i.e., $10^2 \leq N \leq 10^4$). The evaluation results in Fig. 5.21a, Fig. 5.21b, and Fig. 5.21c for a sigle iteration report that: (i) the strategies with prepartitioning are executed in lower running time and use less RAM and CPU capacity than the original version of heuristics; (ii) the application of GATs, GATsPart, GreedyTs, and DPTs is limited to a maximum of $N = 10^3$ services, because for larger scenarios (e.g., for $N = 10^4$) the running time is around units of hours; (iii) the computational capacity used by the heuristcis is between 3% and 11% for RAM (see Fig. 5.21b), and between 4% and 40% for CPU (see Fig. 5.21c); and (iv) the evaluation in terms of running time, RAM and CPU usage, demonstrate that the best strategies for large-scale scenarios are GreedyTsPart and DPTsPart.

Simulations in large-scale scenarios verify the validity of the proposed energy management model and heuristic strategies developed. Particularly, evaluation results demonstrate that the prepartitioning method improve the operation of heuristics in terms of running time, and RAM and CPU usage, which makes these strategies (e.g., GreedyTsPart) have applicability to larger scenarios ($N > 10^4$). Moreover, the performance metrics values indicate that the prepartitioning method can extend the scalability of heuristics and improve the inner operation of the proposed algorithms as in the case of GATs for which its prepartitioned version (i.e., GATsPart) produce better values of AR.

Table 5.8: Maximum value of AR achieved by heuristic strategies in large-case scenarios.

Scenario	No Strategy	GreedyTs		GreedyTsPart		GATs		GATsPart		DPTs		DPTsPart	
	Initial AR	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain	Final AR	AR Gain
V, $N = 10^2$	18.60	95.96	77.36	95.96	77.36	71.64	53.04	97.44	**78.84**	97.06	78.46	97.06	78.46
VI, $N = 10^2$, online approach	18.60	53.20	34.60	53.16	34.56	52.58	33.98	52.56	**33.96**	53.20	34.60	53.12	34.52
V, $N = 10^3$	19.40	98.40	79.00	98.37	78.97	60.51	41.11	98.24	**78.84**	98.01	78.61	98.01	78.61
V, $N = 10^4$	23.49	98.39	74.90	98.39	74.90	55.43	31.94	99.07	**75.58**	98.33	74.84	98.33	74.84

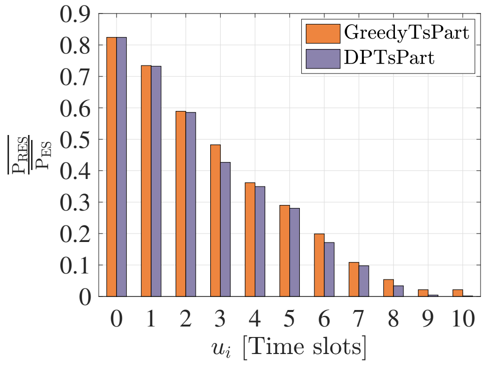
Table 5.9: Running time in seconds of heuristic strategies and G_R of pre-partitioned versions concerning the original version of heuristics in large-scale scenarios.

Scenario	GreedyTs	GreedyTsPart		GATs	GATsPart		DPTs	DPTsPart	
	Running Time	Running Time	GR	Running Time	Running Time	GR	Running Time	Running Time	GR
V, $N = 10^2$, 50 iterations	1.99×10^1	1.76×10^1	$\sim 1.13 \times 10^0$x	4.51×10^3	4.46×10^3	$\sim 1.01 \times 10^0$x	5.57×10^1	2.13×10^1	$\sim 2.62 \times 10^0$x
VI, $N = 10^2$ online, 50 iterations	1.30×10^1	1.58×10^1	$\sim 1.22 \times 10^0$x	3.77×10^3	3.71×10^3	$\sim 1.02 \times 10^0$x	1.65×10^1	1.61×10^1	$\sim 1.02 \times 10^0$x
V, $N = 10^3$, 50 iterations	1.95×10^3	6.89×10^1	$\sim 2.83 \times 10^1$x	5.44×10^4	4.11×10^4	$\sim 1.32 \times 10^0$x	2.14×10^3	1.09×10^2	$\sim 1.96 \times 10^1$x
V, $N = 10^4$, 20 iterations	8.55×10^4	2.26×10^2	$\sim 3.78 \times 10^2$x	7.83×10^5	1.59×10^5	$\sim 4.92 \times 10^0$x	5.64×10^5	3.64×10^2	$\sim 1.55 \times 10^3$x

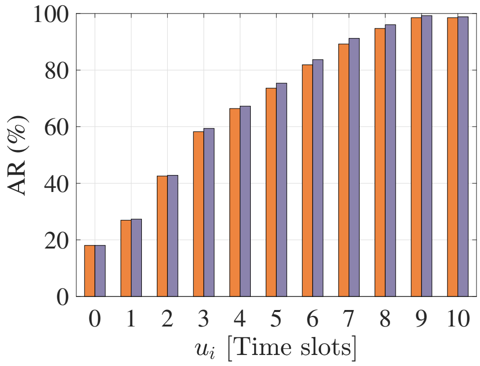
For the evaluation in very-large-scale scenarios, GreedyTsPart and DPTsPart have been chosen, due to their computational capacity usage (less than 9% of RAM and 29% of CPU) and running time (less than 11 seconds) delivered in large-scale scenarios. Fig. 5.23,

Fig. 5.24, Table 5.10, and Table 5.11 summarize the simulation results produced by these heuristics. Like the results obtained in smaller scenarios, for $10^5 \leq N \leq 10^6$, the values of metrics P_{RES} (Fig. 5.23a), AR (Fig. 5.23b), and P_{LACK} (Fig. 5.23c) verify the effectiveness of GreedyTsPart and DPTsPart to adapt consumption to the availability, which results in minimization of P_{RES}. As indicated in Table 5.10, the improvements, in terms of AR, achieved by the two heuristics are very similar to each other and are around 74%; although their running times differ as indicated in Table 5.11 and reveal that the application of DPTsPart is limited to scenarios with $N < 10^6$.

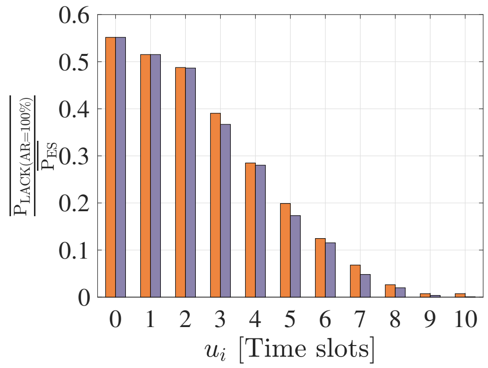
To better differentiate the performance of GreedyTsPart and DPTsPart in very-large-scale scenarios, we have analyzed Scenario VII considering the maximum value of time-shifting (i.e., $u_i = 10\,\forall S_i$) and varying N, as shown in Fig. 5.24. Simulations results in Fig. 5.24a, Fig. 5.24b, and Fig. 5.24c show that in terms of running time and use of RAM and CPU, GreedyTsPart presents a better performance than DPTsPart. Specifically, the running time of DPTsPart for $N = 10^6$ in order of tens of hours, confirm its applicability for scenarios with $N \leq 10^5$. Regarding the computational capacity usage, Fig. 5.24b and Fig. 5.24c report that the heuristic strategies consume between 3% and 12% of RAM, and between 11% and 32% of CPU, respectively. These values and the results of the running-time demonstrate feasibility of adaptive energy management in scenarios with hundreds of thousands or even millions of services.



(a) Scenario VII: P_{RES} heuristics, for $N = 10^6$ and with $NumPart = 10^5$.

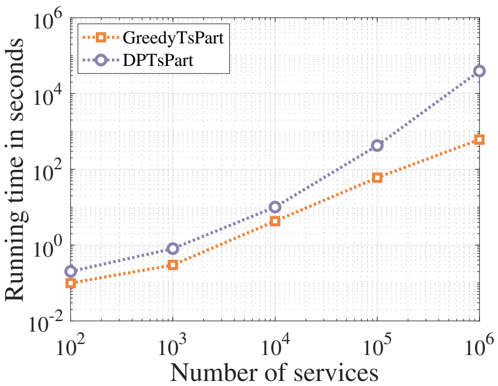


(b) Scenario VII: AR heuristics, for $N = 10^6$ and $NumPart = 10^5$.

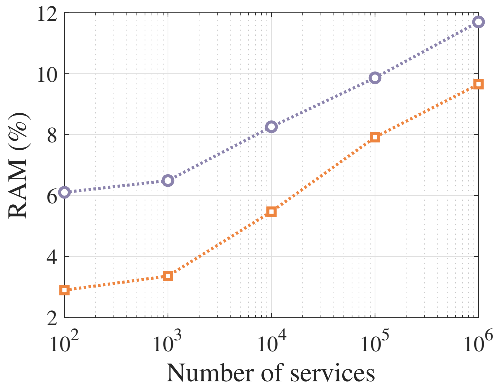


(c) Scenario VII: P_{LACK}, for $N = 10^6$ and with $NumPart = 10^5$.

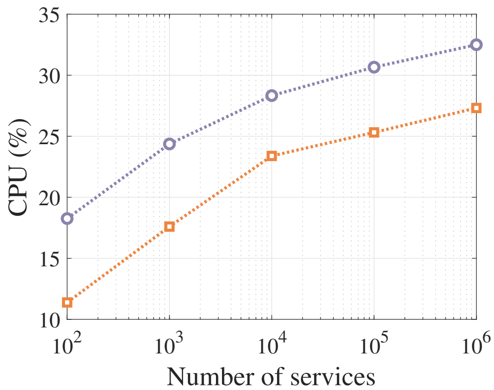
Figure 5.23: Example of performance evaluation of GreedyTsPart and DPTsPart for very-large-scale scenarios according to P_{RES}, AR, and P_{LACK}.



(A) Running time of heuristic strategies in Scenario VII, for $u_i = 10 \forall S_i$, varying N, and using a single core.



(B) RAM usage of heuristic strategies in Scenario VII, for $u_i = 10 \forall S_i$, varying N, and using a single core.



(C) CPU usage of heuristic strategies in Scenario VII, for $u_i = 10 \forall S_i$, varying N, and using a single core..

FIGURE 5.24: Example of performance evaluation of GREEDYTSPART and DPTSPART for very-large-scale scenarios according to Running time, RAM Usage, and CPU usage.

TABLE 5.10: Maximum value of AR achieved by the heuristic strategies in very-large-scale scenarios.

Scenario	No Strategy	GreedyTsPart		DPTsPart	
	Initial AR	Final AR	AR Gain	Final AR	AR Gain
$N = 10^5$	19.40	98.40	79.00	98.37	78.97
$N = 10^6$	23.49	98.39	74.90	98.39	74.90

TABLE 5.11: Running time in seconds of prepartitioned heuristic strategies in very-large-scale scenarios.

Scenario	GreedyTsPart	DPTsPart
	Running Time	Running Time
$N = 10^5$, 1 iteration	6.89×10^1	1.09×10^2
$N = 10^6$, 1 iteration	2.26×10^2	5.64×10^5

The simulations in large-scale and very large-scale demonstrate that the heuristics strategies, when deployed in the NFV domain (or a similar ICT infrastructure), can enable efficient adaptive energy management within reasonable running time and use of computational capacity (mainly in terms of RAM and CPU). Table 5.12 summarizes the main operating features, the mean values of AR and computation capacity used (considering the evaluation in small-scale scenarios for a fair comparison), and the applicability of the proposed service scheduling strategies (both optimal and heuristic).

TABLE 5.12: Summary of features, mean values of metrics, and applicability of service scheduling algorithmic strategies.

Strategy	**Version**	**Performance metrics small-scale scenarios**			**Application scope**		
		AR gain (%)	RAM usage (%) single core	CPU usage (%) single core	Small-scale scenarios $N \leq 20$	Large-scale scenarios $20 < N \leq 10^4$	Very large-scale scenarios $10^4 < N \leq 10^6$
Optimal approach	OPTTSCOST	65.86	29.34	35.04	Yes constrained to $N \leq 10$	No	No
	OPTTSCOSTPART	65.42	1.16	5.68	Yes	Yes constrained to $N \leq 10^3$	No
Greedy approach	GREEDYTS	62.08	1.46	2.26	Yes	Yes but suggested up to $N = 10^3$	No
	GREEDYTSPART	61.97	1.28	1.38	Yes	Yes	Yes
Genetic algorithm	GATS	65.77	1.49	20.91	Yes	Yes but suggested up to $N = 10^3$	No
	GATSPART	63.92	1.32	18.03	Yes	Yes but suggested up to $N = 10^3$	No
Dynamic programming	DPTS	61.50	1.56	10.42	Yes	Yes but suggested up to $N = 10^3$	No
	DPTSPART	60.72	1.40	7.11	Yes	Yes	Yes but suggested up to $N = 10^5$

5.5 Conclusions

This chapter provides insight into optimal and heuristic strategies that can be deployed in the NFV domain (or a similar ICT infrastructure) for achieving adaptive energy management, either renewable or not, in small, large, or very large-scale IoT-enabled scenarios. In this context, this chapter starts by describing the mathematical model related to adaptive consumption and an algorithmic strategy based on brute-force search, denoted as

OptTsCost, to solve the energy management model optimally. The optimal strategy allows us to identify all concerns related to the algorithmic implementation of adaptive energy management.

Given the NP-Hard nature of adaptive energy management and the exponential growth of OptTsCost (which depends on the values of N and u_i as shown in Eq. 5.30), we propose three heuristic strategies identified as GreedyTs, GATs, and DPTs, which are based on a greedy approach, an evolutionary algorithm, and dynamic programming method, respectively. To scale up the applicability of adaptive energy management to scenarios with thousands and hundreds of thousands of energy demands, we have incorporated a prepartitioning method for both the optimal strategy and the heuristics. As a result of the prepartitioning method, four additional heuristics were created and are denoted as OptTsCostPat, GreedyTsPart, GATsPart, and DPTsPart.

The optimal and heuristic strategies are evaluated through intense simulations in various scenarios with different values of N and u_i, using different generation and consumption profiles and offline and online approaches. The evaluation also includes a HEMS scenario in which real-world consumption data is used. The simulation results in all scenarios demonstrating the effectiveness of the proposed adaptive energy management model, which, implemented through the algorithmic strategies, offers improvements in energy use (i.e., more appliances consuming in the same period or the execution of services which would otherwise be rejected without a management mechanism) and reduction of peak demands. The values of performance metrics P_{RES}, AR, and P_{LACK} show that as the time-shifting value increases, the energy model better adapt the P_D to the P_{ES}, which is reflected in a progressive increasing of AR and a decrease of P_{REs} and P_{LACK}. The simulation results also reveal that the proposed strategies in this paper can be a useful tool for the planning of energy consumption and distribution or for real-time load control and optimization of energy use in IoT-enabled environments. These tools can, in turn, offer operational (e.g., the reduction of energy outage preventions) or economic benefits (e.g., reduced energy tariffs) for the ES and the ECs, and the overall improvement of the stability and reliability of the energy ecosystem.

In terms of quality and complexity of solutions, simulation results indicate that the heuristic strategies, both the originals and the versions with prepartitioning, produce high-quality solutions while performing between two and six orders of magnitude faster than the optimal approach OptTsCost (as shown in Table 5.6). Regarding the computational power demanded by the heuristics, the evaluation of heuristics indicates that these strategies only use a fraction of RAM and CPU capacity used by OptTsCost (as shown in Fig 5.19). In the worst case scenario for RAM usage, DPTs uses approximatly 5% of capacity used by OptTsCost, and in the worst case scenario for CPU usage, GATs uses approximatly 60% of capacity used by OptTsCost. Therefore, reduced running-time and computational capacity usage make possible the implementation of heuristics on advanced NFV-enabled infrastructures or on embedded systems in homes (e.g., on a Raspberry Pi platform) for adaptive energy management in IoT-enabled scenarios with hundreds or thousands of services. Likewise, the application of the prepartitioning method allows the energy model to extend the potentialities of the heuristics to IoT-enabled environments with hundreds of thousand of energy demands (as in the current and future communications infrastructures) or as future initiatives of the IoE.

Chapter 6

Application Scenarios

This chapter presents several application scenarios in which the proposed architecture (Chapter 4) or the developed algorithmic strategies (Chapter 5) can be applied for efficient resource management, including energy and spectrum.

This chapter is based on:

J1 **Christian Tipantuña**, Xavier Hesselbach, Victor Sánchez-Aguero, Francisco Valera, Iván Vidal, and Borja Nogales. An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles. *Wireless Communications and Mobile Computing*, 2019.

J3 Victor Sánchez-Aguero, Francisco Valera, Ivan Vidal, **Christian Tipantuña**, and Xavier Hesselbach. Energy-aware management in multi-uav deployments: Modelling and strategies. *Sensors*, 20(10):2791, 2020.

J4 **Christian Tipantuña** and Xavier Hesselbach. Adaptive energy management in 5G network slicing: Requirements, architecture, and strategies. *Energies*, 13(15):3984, 2020.

J6 **Christian Tipantuña** and Xavier Hesselbach. IoT-enabled proposal for adaptive self-powered renewable energy management in home systems. *IEEE Access*, 9:64808–64827, 2021. doi: 10.1109/ACCESS.2021.3073638.

C4 **Christian Tipantuña** and Xavier Hesselbach. NFV-enabled optimal spectrum allocation in flex-grid optical networks. In 2020 22nd International Conference on Transparent Optical Networks (ICTON), pages 1–7, 2020.

In a wide variety of scenarios, performing adaptive energy management can be essential due to the limited energy capacity o because of the dynamic generation and consumption nature of the environment under analysis. In this context, this chapter presents several examples in which the proposed architecture, algorithmic strategies, and the adaptive energy model can be applied entirely or partly. Section 6.1 describes the application of the architecture, management model, and the prepartitioning method for adaptive energy management in a HEMS. Section 6.2 instead presents applying the proposed adaptive energy management concepts into the 5G network slicing landscape. Section 6.3 presents adaptive energy management for a UAV-enabled communications system using an optimal strategy in a single service region where total battery consumption is considered for replacement. Section 6.4 instead describes the energy management in a multi-UAVs enabled communications environment where (through heuristic strategies) multiple service regions are covered, and the battery replacement is not constrained to the total consumption. Although in the examples in Section 6.3 and Section 6.4, the architecture, strategies, and consumption model are not applied directly (as in the case of the examples in Section 6.1 and Section 6.2), the concepts and ideas developed for adaptive energy management are adapted to achieve efficient management of available energy obtained form batteries of drones. This condition indicates that the knowledge acquired in this thesis has a great potential for direct or indirect application in various fields, such as in communications enabled by UAVs. As examples of other application fields for adaptive energy management, we can mention energy management in spacecraft in which the only supply comes from the sun or in electric vehicles in which adequate energy management would improve the performance of the vehicle, the distribution and operation of charging points, as well as the duration and life of the battery.

Likewise, to demonstrate that the algorithmic strategies proposed in this thesis (Chapter 4) can be applied beyond the energy field, Section 6.5 presents an optimal spectrum allocation scheme in DWDM flex-grid optical networks. In this case, adaptive management concepts are applied to efficient management of available spectrum by proposing an ILP model, which is solved by an exact method and a heuristic method based on a prepartitioning strategy.

We also state that the application scenarios of Section 6.3 and Section 6.4 have been joint work with Universidad Carlos III de Madrid. In this regard, we indicate that from the UPC we have collaborated with the proposal and development of the research, which has resulted in publications J1 and J3 (see Section 1.4). We have decided to include all the corresponding text of the proposals to explain the problems addressed comprehensively, the algorithmic solutions developed and the results obtained within the scope of the scenarios considered.

6.1 Adaptive Energy Management in Home Energy Systems

The new generation of communication networks can provide massive connectivity of devices, extremely low latency, higher capacity, and increased bandwidth. These features enable the deployment of management systems in different sectors such as energy and in a wide variety of environments such as agriculture, surveillance or home systems. In this regard, this section proposes a self-powered adaptive and automated home energy control system enabled by the IoT technologies. The system aims to adapt the consumption patterns to the availability of self-generated renewable energy (produced from solar panels, wind-mills, etc.). As part of the proposal, this section presents the consumption negotiation scheme for IoT devices, the management mechanisms to optimize the use of the available energy, and the related model. Given the complexity of the adaptive management process, the proposal also presents a heuristic strategy based on a prepartitioning method to obtain feasible solutions in a reasonable running time. The simulation results for offline and online scenarios validate the advantages of the proposed strategy, and the numerical improvements are presented.

6.1.1 Introduction

As ICT become more reliable, cheap, accessible, and are able to provide lower latency and higher bandwidth, a wide range of IoT devices are being extensively deployed worldwide. According to some studies [118], it is estimated that by 2022 the number of IoT devices will exceed 28.5 billion. This large number of devices, together with modern communications networks such as 5G and beyond, represents an essential component in our current digital society, which can be used in various vertical applications for different sectors, such as agriculture, transportation and energy. Specifically, in energy systems, the incorporation of ICT and IoT technologies has enabled the improvement of control, monitoring, and management operations and the deployment of DR schemes [62].

A prominent scope of application of DR systems is in domestic environments (including homes and buildings), which are active players in the energy sector and responsible for around 40% of the consumption of the total energy generated [119]. Considering the penetration of connectivity through access networks (such as Wi-Fi) in households and the automation and communication infrastructure available on modern appliances (e.g., a smart tv), implementing an IoT-enabled energy consumption solution in domestic environments is a feasible alternative with current technologies. This kind of approach in the literature is referred to as HEMS and mainly aims to reduce consumption and improve energy efficiency [119].

Given that a reduction or efficient use of the available supply (mainly provided and distributed by the energy utility company) is not sufficient to address the ever-increasing demand, solutions such as integrating renewable energy sources (also referred to as green energy) into energy systems are envisioned as a sustainable alternative, in the road to replace completely the non-renewable sources. The energy obtained from sources such as sun or wind is practically an inexhaustible resource that generates a lower environmental impact than fuel or carbon-based energy and can be used to meet partial or total consumer requirements. In this regard, some research has analyzed the primary and exclusive use of green energy. For instance, major IT providers such as Google [4] and Microsoft [5] are already promoting networking infrastructures fully powered by renewable energy. For domestic environments, the use of self-generated renewable energy from solar panels or wind stations has also been recently promoted [120]. This trend has changed the traditional role of the end-user from consumer to energy producer; the term ”prosumer” has even been coined to indicate the dual behavior of generation and consumption.

Energy production on the consumer side avoids extra generation by the ES (which can produce pollution) while contributing to a better distribution of the generated energy resource. However, considering that green sources have stochastic behavior and their production is affected by environmental conditions or geographical location, the available supply is frequently underused or wasted because it cannot be consumed or stored in mass amounts. Therefore, an adaptive HEMS focused on optimized utilization of self-generated green energy is essential because it has implications for the consumer side (lower tariffs and supplier independence), on the supplier side (lower level of production and reduction of power peaks), and on the environment (lower utilization of fossil fuel-based energy).

6.1.2 Related Work

This section discusses research work related to ICT- and IoT-enabled customer-side energy management, including the HEMS.

6.1.2.1 ICT- and IoT-enabled Customer-side Energy Management

ICT and IoT technologies enable robust automation, control, monitoring, management processes, the enhanced inclusion of new power plants (mainly based on green sources), and bidirectional communication between ES and EC [62]. This latter feature is of paramount importance in current energy systems because it allows EC to participate in DR programs actively and make decisions to adapt consumption to availability. The adaptive customer-side management helps energy providers flatten the demand curve by allowing EC to schedule power usage from peak to off-peak periods [68, 121], for instance adapting the charge of the electric car at home [122]. These actions result in increased stability and sustainability of the energy system, reduced carbon emissions levels due to lower production and energy usage, and reduced overall operational cost through the optimization of available resources. In addition, EC can be motivated to participate in DR adaptive schemes by the variation of energy prices over time, incentives related to energy use, or when the energy system requires reliable and efficient operation.

To implement DR schemas, various mathematical models and strategies (algorithms) have been proposed in the literature focused on different requirements and objectives to be achieved. In this regard, adaptive consumption has been modeled through the operational features of ES and EC, including in some cases price information, energy storage units participation [121], and green energy provisioning [71]. To mathematically formulate appropriated models, optimization methods such as integer linear programming (ILP) [72] and mixed-integer linear programming [114] have been used. In addition, different objective functions have been established for these models, such as minimization of power consumption (reshaping the peak demand) [68], maximization of user utility [69], and minimization of costs for the ES, EC, or both [121]. To solve the adaptive energy management problem (energy model), offline [73] and online [74] approaches have been analyzed in research works. For offline approaches the algorithmic strategy has complete knowledge of supply and demand during the time horizon of analysis to manage energy. For online approaches the algorithmic strategy has no future information about ES and EC, and decisions on loads are restricted as energy resources and demands evolve.

The offline and online approaches can be addressed by optimal or suboptimal methods. The optimal or exact strategies allow the best allocation of energy recourse through the optimal scheduling of loads [74]. However, these methods present an exponential complexity that grows as customers and management techniques (e.g., time-shifting and prioritization) increase; also, they are computationally expensive and time-consuming [14]. To overcome complexity limitation and for practical applications, some research work has focused on deploying sub-optimal and faster methods based on heuristics strategies [13]. For instance, for customer-side DR energy management Logenthiran *et al.* in [13] propose a heuristic strategy based on evolutionary algorithms for minimizing peak load demands; this strategy is tested for different types of loads in residential, commercial, and industrial service areas.

6.1.2.2 IoT-enabled HEMS

Communications systems such as Wi-Fi access networks and IoT infrastructures (including smart sensors, devices, appliances, and virtual assistants) allow home automation, as well as the control of loads. Regarding energy management, the ICT and IoT technologies facilitate the deployment of different schemes [64]. The proposed approaches can be limited to manual or partially automated control of consumption (through the activation or deactivation of appliances) or can comprise entire HEMS, in which a smart entity (i.e., software or hardware platforms) efficiently schedules the consumption of loads. The energy demands can be scheduled according to multiple criteria, including the consumption optimization, costs minimization, or ensuring certain comfort levels for customers. The HEMS can also operate in different application scopes (e.g., residential or professional) considering weather conditions, load and supply profiles, energy storage capability, green energy mixing, and interaction with other energy systems such as smart grids [123].

Many architectures, software and hardware tools, platforms, and tesbeds have been proposed for implementing HEMS [124]. These approaches are based on existing communications networks, either wired (e.g., SDN [9] or power line communications [125]) or wireless (e.g., Wi-Fi [126]), and the vast range of IoT technologies and protocols (e.g., machine-to-machine communications protocols [127]) that are included or that can be easily incorporated into domestic appliances. Some approaches also exploit cloud computing solutions such as Thingspeak [128] and data analytics [129]. In addition, using renewable energy in HEMS is an important feature addressed in the literature to deliver an environmental friendly and cost-effective solution. For more than 15 years, studies have analyzed the possibility of using exclusively renewable energy in households [120]. In this regard, different renewable sources have been evaluated; however, due to the ease of installation (e.g., rooftop photovoltaic units [130]), accessibility in the market (easy installation kits [131]), and production capacity (e.g., 7000 [W] [131]), the use of solar energy (photovoltaic energy) has been analyzed in many research works.

Some studies describe entire HEMS while others focus mainly on the development of strategies (including DR schemes). For instance, AlFaris *et al.* in [64] present a HEMS that, using IoT-enabled appliances, sensors, actuators, smart meters, and a central hub (in which DR schemas are executed) wirelessly interconnected, performs consumption management of supply generated by photovoltaic panels. Regarding strategies, a variety of mathematical models and algorithms are proposed for HEMS by researchers (in a similar way to the customer-side management approaches in Section 6.1.2.1). The proposed models are based on optimization techniques such as linear programming [71] and stochastic programming [130] and solved by optimal [69] or heuristic algorithms (e.g., based on evolutionary algorithms [132]). These latter are used for dealing with the complexity issues.

In summary, existing works show that ICT and IoT are key enablers for deploying customer-side management and HEMS. Most of the proposals analyze the HEMS as an extension of smart grids and as a component that can be managed to reduce the load of ES. In many proposals, HEMS are targeting the minimization of total energy consumption. In contrast, the proposal of this section is focused mainly on the domestic environment. The proposed HEMS aims to optimize the utilization of available supply, primarily produced from green sources, through efficient scheduling of IoT-enabled consumers. In addition, the proposed solution, which maximizes the utilization of available supply at all times, allows the implementation of a battery-free HEMS system, which is in line with the recent concept of battery-free IoT networks [61]. The battery-free paradigm seeks to address the problems related to energy storage, such as limited storage capacity and lifetime of battery units, mandatory replacement, and pollution produced [60].

6.1.3 Proposal for Adaptive Energy Management in HEMS

6.1.3.1 Problem Description

The concept of adaptive energy management (through DR schemes) applied to the domestic environment and deployed in a HEMS, allows optimizing the use of self-generated energy (mainly from green sources), avoiding wasted supply and reverse power flow if generation is sent to the power grid (substation) in an uncontrolled manner [133]. In this regard, adaptive energy management can impact in the ES and distributors to change the exclusive centralized control and generation ecosystem to a paradigm in which the customer can autonomously manage the energy demands according to the self-generation. Furthermore, enabled by IoT, a HEMS can not only perform real-time monitoring and smart energy management of domestic appliances but also optimize energy utilization, by controlling the individual consumption of devices, specially on peak times [124].

6.1.3.2 IoT-enabled Architecture Proposal

Figure 6.1 shows the proposed HEMS architecture; the components are described below.

- *Communications technologies:* Communication systems are a fundamental component in modern energy systems because they allow bidirectional communications between generation and consumption. In HEMS, two communications systems can be identified: (i) the backbone network or Wide Area Network (WAN), which exchanges energy-related data (instructions and notifications) with the ES needed for DR programs (consumer-side participation) and (ii) the ICT in the premises of the end-user that allows the creation of a Home Area Network (HAN) in a domestic environment, a Building Area Network (BAN) in a building, and an Industrial Area Network (IAN) in industrial facilities to control, monitor, and manage loads. A variety of wired and wireless standards and communications technologies can be used according to the requirements and needs of control applications and customers. Different factors such as data rate, latency, power consumption, topology, network size (number of devices being controlled), reliability level, operating frequency, range of coverage, security level, and implementation costs can determine the choice communication systems [123]. For instance, for the WAN between HEMS and ES, a possible alternative is Power Line Communications (PLC) or synchronous digital hierarchy, which can use an optical power ground wire. For HAN (BAN or IAN), Wi-Fi is widely used, although other technologies can provide connectivity in HEMS. Table 6.1 shows a summary of possible ICT for deploying a HAN.

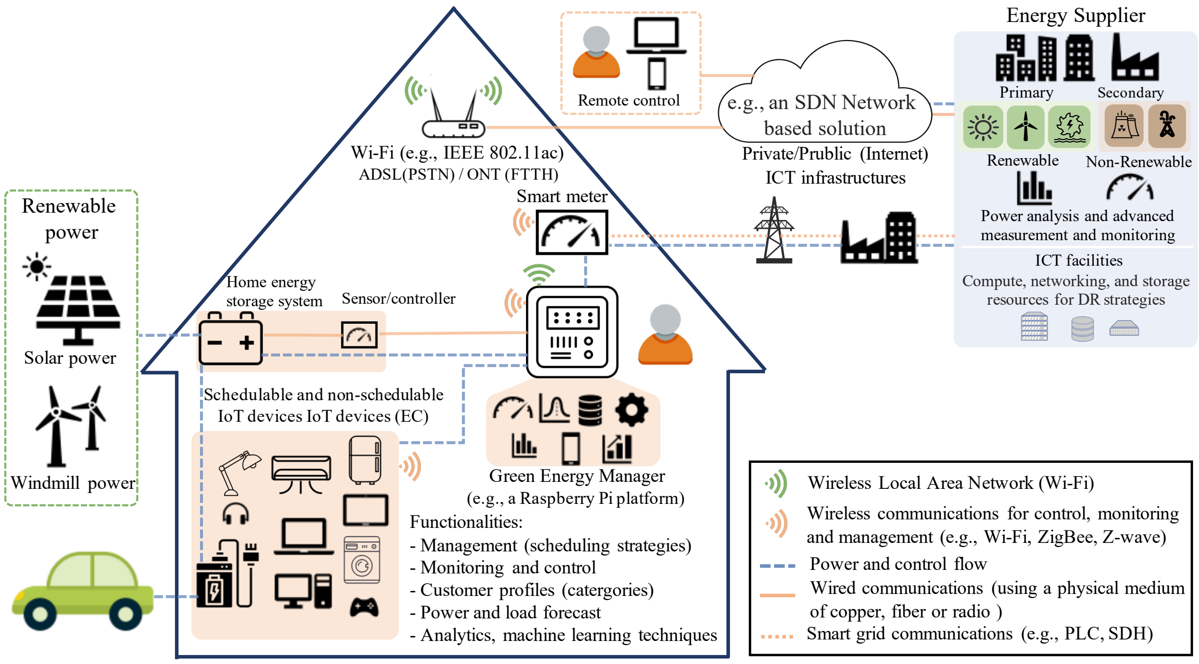


FIGURE 6.1: IoT-enabled HEMS architecture proposal.

Depending on the application scope of the HEMS and customer requirements, more specific analysis such as channel modeling can be carried out to estimate operation limits of a particular technology for deploying a HAN, as in the case of PLC studied in [134]. Other factors that can influence the selection of the technology for implementing a HAN are pace of technology innovation, upgradability, device ownership (e.g., smart meters from the utility company), interoperability (i.e., availability of compatible appliances or interfaces) [135], self-managiability, maintainability, and resiliency [136]. Because secure management in a HEMS is a relevant aspect that avoids compromising EC and ES operation, requirements such as client (device) authentication, integrity, and confidentiality of energy-related data must also be met by the selected HAN technology [137].

In addition, communication protocols compatible with wired and wireless technologies can be used to implement energy management applications. The proposed HEMS has no special constraints regarding the protocols architecture, it can use the Internet TCP/IP suite, a generic protocol stack in which physical, network, and application layers are defined, as discussed in [138], or even other radically different approaches, such as RINA [139].

The physical and data-link layer in HEMS are mostly defined by Ethernet, Wi-Fi or PLC protocols. For the network layer, IPv6 is preferred to IPv4, regarding the capacity to handle overcrowded scenarios with millions of IoT devices. About the transport layer, TCP is in favor of reliability and UDP for real-time operation. In upper layers, customized solutions can be implemented (e.g., a web server running the management application), or existing protocols such as COAP and MQTT [140] can be adapted to the application scope or need of EC.

Table 6.1: Comparison of different communications and networking technologies for HEMS, based on [124] and [12].

Connectivity	Technology	Standard	Data rate	Range
Wired	Ethernet	IEEE 802.3	10-1000 Mbps	100 m
	X10	X10 standard	50-60 Kbps	300 m
	HomePlug	IEEE P1901	14-200 Mbps	300 m
	Insteon	X10 standard	1.2 Kbps	3 Km
	ITU G.hn	ITU G.hn	Up to 2 Gbps	500 m
Wireless	Wi-Fi	IEEE 802.11	54 Mbps at 2.4 GHz, 5 Gbps at 5 GHz	100 m
	Zigbee	IEEE 802.15.4	40 Kbps at 915 MHz, 250 Kbps at 2.4 GHz	100 m
	Thread	IEEE 802.15.4	250 Kbps at 2.4 GHz	30 m
	Bluetooth	IEEE 802.15.1	1 Mbps at 2.4 GHz	10 m
	Z-wave	Zensys, IEEE 802.15.4	40 Kbps	30 m
	6LowPAN	IEEE 802.15.4	20 Kbps at 868 MHz, 40 Kbps at 915 MHz, 250 Kbps at 868 MHz	75 m
	ONE-NET	Open source	38.4 - 230 Kbps	70 m
	EnOcean	EnOcean standard	120 Kbps	30 m
	LoRa	LoRa	10 Kbps at 863 MHz, 100 Kbps at 915 MHz	10 Km
	Sigfox	Sigfox	10 Kbps at 863 MHz, 100 Kbps at 915 MHz	10 Km

- *Smart meters:* The smart meters in HEMS are used to establish two-way communication with the utility company. They are responsible for sending consumption data, receiving signals (e.g., about energy pricing information) for participation in DR programs, and requesting energy resources if the self-generated renewable energy fails or becomes insufficient for the demand. Smart meters can display energy usage patterns to end-users and are coordinated by Green Energy Manager (GEM), which is the main controller in HEMS.

- *IoT devices:* IoT devices are home appliances (e.g., a Smart TV) and other devices (e.g., access points) that participate in energy management process. They are connected, coordinated, and controlled by GEM. Each IoT device requires a unique identification (e.g., an IPv6 address) to send consumption information (periodically, when activation is needed, by polling, or others) and receive operating instructions (e.g., to change to an activation or deactivation state). IoT devices can incorporate sensors and actuators to perform measurement, monitoring, and control tasks. In the proposed model, their operation is uninterruptible. In addition, these devices can be fully automated (e.g., washing machines, dishwashers, and air conditioners) or can require manual operation by the end-users (e.g., computer, television, and vacuum cleaner).

- *Renewable power:* Renewable power coming from green sources is the primary supply in the proposed HEMS. The distribution of this available supply is controlled and managed by GEM. DC/AC, AC/DC, or DC/DC converters can be used optionally into green energy generation to match different load requirements. Moreover, renewable energy production systems can incorporate battery units to ensure stable transitions between energy sources (either green or not). Battery units in a HEMS can also act as energy buffers, storing energy during surplus periods or releasing the stored capacity according the requirements of consumers (always under the coordination of GEM). In the context of a HEMS, the battery units already included in the structure of green sources (e.g., photovoltaic panels) can be used for these purposes. However, a more interesting approach (also part of smart cities) is the use of electric vehicles. Specifically, the battery units in modern electric vehicles can store the surplus energy generated, which can then be used when necessary by a single household, several households, or an entire building in a neighborhood, and can even be distributed to other locations in smart cities through coordinated actions between the HEMS and the ES. Since energy storage can improve HEMS performance, an analysis of battery sizing and characterization can be considered for future research.

  In the proposed HEMS model, the exceeding renewable energy self-generated can be incorporated to the general power grid. The management and control actions are performed by GEM and smart meter. On the contrary, if the available green energy is insufficient to meet all the demands, GEM and smart meter can execute actions to request extra energy from the ES.

- *Green energy manager:* GEM is a software and hardware platform with two main responsibilities: (i) creation and management of the HAN (or BAN or IAN) and (ii) control, monitoring, and energy consumption control. Once the network is created by the GEM, each IoT device must be associated. In this process, the device is authenticated, identified, registered, and assigned an IP address.

  To allow interaction with the end-user, GEM has an interface able to display information about the consumption (e.g., through a dashboard) and allows input of data related to the appliances and requirements. In general, the end-user provides parameters (such as the type of device, average consumption level, priority level, and possible operation intervals). If the user does not have information about the power demanded by the home appliance, GEM can use data stored in its database. Once integrated into the HAN, the IoT device can report real consumption data to GEM. To facilitate, GEM can provide the user with pre-defined profiles, (including parameters of the mechanisms such as time-shifting or quality degradation) or can be customized on basis of consumer preference. GEM can use different technologies to manage and control the consumers (e.g., SDN technology).

The adaptive management of GEM, is summarized in the following three steps: (i) reception of information about the consumption of appliances, level of available supply (self-generated and coming from the ES), and data provided by end-user; (ii) execution of appropriate scheduling strategies; and (iii) transmission of orders to the appliances in order to handle the degree of energy consumption. GEM can include features such as logging for users, alarm reporting, and fault detection. In addition, through the smart meter or home gateway (e.g., access point), GEM can interact with external networks and allow remote monitoring and management (e.g., through web applications). In the latter case, the application of security mechanisms (e.g., encryption and hashing) is essential to guarantee user privacy and non-corruption of the energy management data. GEM could also incorporate other types of management strategies based on data analytic (e.g., generation and consumption forecasting) or machine learning techniques.

6.1.3.3 Management Mechanisms to Adapt Consumption to the Available Supply in HEMS

The proposed HEMS considers a prioritization scheme defined by the end-user that differentiates the applicability of the management mechanisms upon the energy loads. Several priority levels can be created according to the preferences and needs of end-users; however, for practicality, these levels can be grouped into CS and NCS. CS includes appliances that are considered essential for the end-user. GEM must ensure the energy allocation (even if it comes from the ES) for their execution. In contrast, NCS covers appliances whose operation is secondary and subject to modification according to the strategies used.

The mechanisms applied to NCS are: (i) the time-shifting capability (backward or forward) to adapt consumption (flattened, smoothed) to the shape of the available supply so improving energy efficiency; (ii) the rejection of services (i.e., not allocation of energy by GEM); and (iii) a quality degradation level, in addition to the mechanisms described in Chapter 4 (Section 4.1.2), this mechanism allows to reduce the power consumption of an appliance, to the cost of degradation in quality. The degradation can be unappreciated in many devices with a minimum and maximum operating thresholds. Selecting the minimum threshold would cause less power to be assigned to a device, allowing adaptation of the demand to the current supply and potentially reserving power to execute another device. Fig. 6.2 illustrates both the normal operation and the application of management mechanisms on appliances in HEMS, considering mainly auto-generated green energy.

As mentioned above, in a HEMS, there may be several levels of priority according to user requirements and the complexity of the desired management system. For instance, for a HEMS with three priorities, the first priority level would correspond to CS, and no management mechanism would affect appliances in this category. An example of service (appliance) in this category could be the operation of a refrigerator. For the second priority level, only time-shifting would be used with values established according to the interests of the end-user. An example of a service in this category would be the operation of a washing machine. For the third priority level, both time-shifting and quality degradation could be used; charging a laptop battery can be included in this category. In any case, the users could configure as many levels as they consider appropriate and establish the management mechanisms and their respective ranges for each priority according to their preferences. It is the task of GEM and specifically of load scheduling strategies to determine the best actions for each appliance.

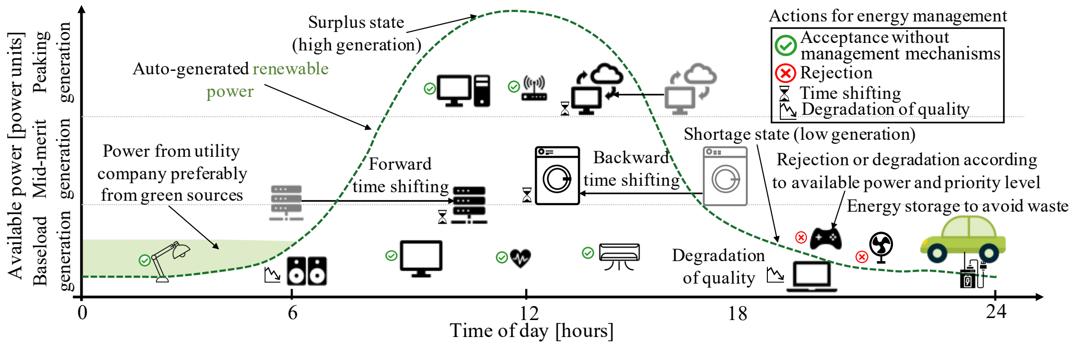


FIGURE 6.2: Example of the application of management mechanisms to achieve adaptive energy consumption in HEMS.

Figure 6.3 summarizes the communication dialog or signaling between the appliances and the GEM in the HEMS. This is a particular application of the scheme shown in Fig. 4.5 (Section 4.1.4). In this negotiation, the existing supply, the parameters of the appliances (i.e., consumption, starting time, and duration), and the information provided by the end-user (priority levels and values of the management mechanisms) are considered. The GEM computes and notifies the appliances through the HAN, about the consumption conditions. This is basically the start time and the consumption levels (including degradation, if applicable). Subsequently, the appliance activates its operation using actuators or control circuits. Different communication schemes can be used to establish communication between the GEM and the appliances. A simple implementation can be based on a client-server architecture using a Wi-Fi access network and the classical Internet protocols to exchange operation conditions and consumption instructions, as exemplified in Fig. 6.3.

6.1.3.4 Adaptive Energy Management Model

The management model in Section 5.1 has been adapted to the scope of the proposed HEMS. In this context, the total available power in the HEMS is identified as P_A and is composed, primarily, by the self-generated green energy that is denoted as P_R and, secondly, by the supply offered by the ES defined as P_{ES}, as shown in Eq. 6.1.

$$P_A = P_R + P_{ES} \tag{6.1}$$

The self-generated energy available in the HEMS can be composed of different green sources, such as sun or wind, as shown in Eq. 6.2; while, the energy supplied by ES can come from renewable (P_{ES}^{R}) and non-renewable (P_{ES}^{NR}) sources, as indicated in Eq. 6.3. If necessary, the HEMS (GEM) can request power from the ES, indicating the preference for green sources in the DR negotiation.

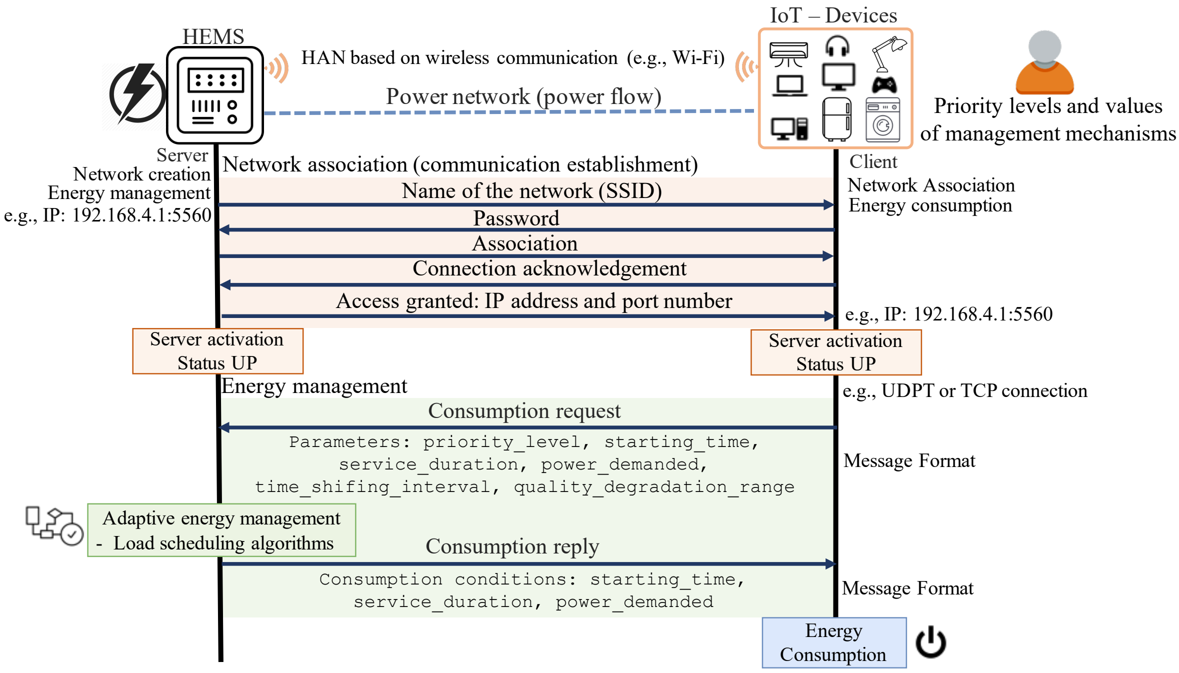


FIGURE 6.3: Summarized energy consumption negotiation process between the GEM and the IoT-enabled devices.

$$P_R = +P_{solar} + P_{wind} \tag{6.2}$$

$$P_{ES} = +P_{ES}^{NR} + P_{ES}^{R} \tag{6.3}$$

To guarantee the primary use of self-generated energy, the provisioning selection process by the GEM is based on a cost function. The generalized form of this function is presented in Eq. 6.4. This expression is composed of the costs associated with P_R and P_{ES} and is affected by the weights w_1 and w_2 that can be set in the interval [0,1], if applicable. Considering the preference for P_R, its associated cost would be zero; in this case, the expression in Eq. 6.4 would be simplified as indicated in Eq. 6.5. The value of $Cost_{P_{ES}}$ would be proportional to the amount of resource demanded (used). Therefore, the prioritization of the self-generated supply would be given by minimizing $Cost_{P_{ES}}$ (i.e., minimization in the use of P_{ES}), as shown in Eq. 6.6.

$$Cost_{P_A} = w_1 \times Cost_{P_R} + w2 \times Cost_{P_{ES}} \tag{6.4}$$

$$Cost_{P_A} = Cost_{P_{ES}} \tag{6.5}$$

$$minimize\{Cost_{P_{ES}}\} \tag{6.6}$$

In the proposed energy management model, the appliances are characterized by their operation, considering the respective management mechanisms. In the context of adaptive energy management in HEMS a service i is identified as S_i and is defined by the parameters in Table 6.2.

TABLE 6.2: Parameters of a service.

Parameter	Description	Unit/Comment
$\mathcal{N}$	Set of services	$\mathcal{N} = \{1, \ldots, N\}$, integer number
i	Service identifier	$i \in \mathcal{N}$
$\mathcal{L}$	Set of priority levels	$\mathcal{L} = \{1, \ldots, L\}$, integer number
j	Priority level identifier	$j \in \mathcal{L}$, $j = 1$ for CS, and $j = \{2, \ldots, L\}$ for NCS
$\mathcal{Q}$	Set of quality degradation levels	$\mathcal{Q} = \{1, \ldots, Q\}$ integers related to a multiplication factor
k	Quality level identifier	$k \in \mathcal{Q}$
t_i	Starting time of service S_i	Time units (e.g., units of minutes or hours)
d_i	Duration of service S_i	Time units (e.g., units of minutes or hours)
p_i	Power demanded of service S_i	Power units (e.g., units watts or kilowatts)
l_i	Priority level of service S_i	Integer number
q_i	Quality level of service S_i	Integer number related to a multiplication factor to decrease consumption (e.g., for $q_i = 2$, $p_i \times 0.75$)
Ts_i	Time-shifting value of service S_i	Time units, $+Ts_i$ forward, $-Ts_i$ backward

In the energy management model, each service S_i has independent operation parameters t_i, p_i, and d_i, and belongs to a certain priority level l_i that defines the actions that affect the service. For instance, a service S_i can be subject to a forward time-shifting (i.e., when $t_i + Ts_i$) for delayed execution or a backward time-shifting (i.e., when $t_i - Ts_i$) for anticipated execution. If the service S_i runs at its original time, then $Ts_i = 0$. In any case, the service S_i can be analyzed by the GEM (scheduling strategy) in the interval $\{t_i - Ts_i, \ldots, t_i, \ldots, t_i + Ts_i\}$. Analogously, the quality degradation q_i can affect the normal consumption p_i of the service S_i. It is the responsibility of the GEM to find the best action (i.e., values of the management mechanisms) for each service, so that the available energy supply is used optimally. In the worst case (the offline approach), the GEM must simultaneously perform the analysis for all N services. In this case, the total amount of power demanded denoted as P_D is equal to the sum of the contributions of each service, as indicated in Eq.6.7.

$$P_D = \sum_{i=1}^{N} p_i \tag{6.7}$$

Therefore, the objective of optimizing the available power consumption can be expressed as the difference between P_A and P_D (i.e., the P_{RES}) , and, specifically, as the minimization of this difference, as indicated in Eq. 6.8.

$$minimize\ \{P_A - P_D\} \tag{6.8}$$

The objective of minimizing the P_{RES} in the proposed HEMS considers the assumptions in Section 5.1.1 and can be expressed as indicated in Eq. 6.9.In this model, the initial time of appliances or of P_A (denoted as $T_init^{P_A}$) can vary from zero to the maximum time horizon W (equally discretized in time slots w), which usually is set to 24 hours. In addition, the size of the time slots can be configured based on the application scope and requirements of each end-user. For instance, a time slot w could represent 10 minutes or 1 hour.

$$\forall w \in \mathcal{W} : minimize \left\{ \sum_{w=1}^{W} (P_A[w] - P_D[w]) \right\} \tag{6.9}$$

The objective function in Eq. 6.9 considers the following constraints.

$$C1 : P_A[w] \geq 0 \tag{6.10}$$
$$C2 : (P_A[w] - P_D[w]) \geq 0 \tag{6.11}$$
$$C3.1 : \sum_{j=1}^{L} y_{ij} = 1, i \in \{1, \ldots, N\} \tag{6.12}$$
$$C3.2 : y_{ij} \in \{0, 1\}, i \in \{1, \ldots, N\}, j \in L \tag{6.13}$$
$$C4 : t_i \geq 0 \tag{6.14}$$
$$C5 : \{t_i - Ts_i\} \geq 0 \tag{6.15}$$
$$C6 : W \geq \max\{t_i + d_i + Ts_i\} \tag{6.16}$$
$$C7 : T_{init}^{P_A} \geq 0 \tag{6.17}$$
$$C8 : \sum_{k=1}^{Q} \sum_{i=1}^{N} \sum_{e \in G_i} (p_{kie} \times q_{kie})[w] \times x_{kie} \leq P_A[w] \tag{6.18}$$
$$C9.1 : \sum_{e \in G_i} x_{kie} = 1, k \in \{1, \ldots, Q\}, i \in \{1, \ldots, N\} \tag{6.19}$$
$$C9.2 : x_{kie} \in \{0, 1\}, k \in \{1, \ldots, Q\}, i \in \{1, \ldots, N\}, e \in G_i \tag{6.20}$$

$C1$ and $C2$, ensure a positive value for P_A and P_{RES}, respectively. $C3.1$ guarantees the assignment of a unique priority level for the service S_i. The variable y_{ij} is set to 1 if the priority $l_i = j$ of S_i exists, as shown in $C3.2$. The temporal constraints in the energy model are ensured by $C4$, $C5$, $C6$, and $C7$.

$C8$ constraints the maximum consumption capacity in the energy model. In $C8$, the decision variable shown in $C9.1$ and $C9.2$ guarantees the processing of the service S_i with unique values of time-shifting and quality degradation. The variable x_{kie} is set to 1 if the service S_i with priority $l_i = j$ and quality degradation level $q_i = k$ exists, as shown in $C9.2$. This limitation avoids the processing (energy allocation) of multiple copies of the same service S_i.

The application of time-shifting and/or quality degradation mechanisms on a service S_i produces different versions of the service S_i, which we define as *variations of the service* S_i. To know the dynamics of these variations in the proposed model, first, the impact of the time-shifting, and then the quality degradation on the N services, are analyzed. The use of the time-shifting mechanism on N services produces N mutually disjointed classes $G_1, \ldots, G_N$ of services. Each class G_i is composed of the shifted versions (variations) of the service S_i considering the complete time-shifting interval (i.e., including the original version of services, when $Ts_i = 0$). Once the shifted versions (variations) of the services have been obtained, including the original versions, we proceed to analyze the application of the quality degradation mechanism on them. As for the time-shifting mechanism, in the case of quality degradation, all possible values are considered. This procedure causes Q mutually disjointed classes $H_1, \ldots, H_Q$ of variations to be created. Each class H_k is composed of the degraded versions of the shifted variations of the service S_i (i.e., variations first affected by time-shifting and then affected by quality degradation).

Considering that the application of management mechanisms produces many variations, the adaptive energy management problem is summarized by choosing the best possible variations of services (i.e., for which x_{kie} takes on a value of 1), such that the utilization of P_A is maximized (and P_{RES} is minimized). In this context, the set of N or n (with $n \subset N$) variations simultaneously analyzed is defined as a *combination of variations* or *combination*, denoted as $Comb_f$ (e.g., $Comb_1$).Then, the objective of the load scheduling strategies (optimal or heuristics algorithms) is to find the best combination among all possible combinations (denoted as $AllComb$, with $Comb_f \in AllComb$) produced due to different variations of services.

Since adaptive management of consumption is summarized in finding the best combination of services (joint action of energy demands that minimize P_{RES}) through load scheduling strategies optimal (e.g., brute force methods) or approximate (e.g., heuristic methods), it is necessary to establish the criteria that allow: (i) assessment of the improvement in energy use, and (ii) selection of the most suitable combination among all possibilities. To meet these criteria, the metrics in Section 5.1.2.4 are considered, including an additional metric defined as Energy Utilization (E_{A_U}). This metric is expressed as indicated in Eq. 6.21 and measures the amount of energy allocated to processed services (appliances) concerning available energy ($E_{A_U} = 100\%$, if $P_D = P_A$).

$$E_{A_U} = \frac{\sum_{k=1}^{Q} \sum_{i=1}^{N} \sum_{e \in G_i} p_{kie} \times q_{kie} \times d_{kie}}{P_A \times W} \times 100\% \tag{6.21}$$

The proposed energy model establishes a cost function to evaluate the quality of the combination of variations ($Comb_f$) to be selected, i.e., the one that optimizes the use of P_{RES}) while maximizing the comfort level of the end-user. Specifically, the combination with the lowest cost represents the scheduling of services that optimizes use of P_A (i.e., adaptive energy management). The cost function denoted as $Cost_{comb_f}$ considers information from the metric AR ($Cost_{AR_f}$, with $\mathcal{M}$ a big value, e.g., $\mathcal{M} = 1000$) and the cumulative value of parameters l_i ($Cost_{L_f}$), q_i ($Cost_{Q_f}$), and Ts_i ($Cost_{Ts_f}$) of variations (services) in the combination f. Equation 6.22 defines $Cost_{comb_f}$, while the individual costs that are part of this expression are defined in Eq. 6.23, Eq. 6.24, Eq. 6.25, and Eq. 6.26. If necessary, the values of cost functions in Eq. 6.22 can be modified by GEM based on the preferences of the end-user, using the weights α, β, γ, and δ in the range [0,1]. For analytical simplicity, these weights are set to one in the proposed model. The cost function is computed for each $Comb_f \in AllComb$. From this list, the best cost identified as $OptCost$ has the lowest value, as shown in Eq. 6.27, and represents the optimal scheduling of services that achieves adaptive energy consumption. In case there are several optimum costs, the selection can be made randomly.

The objective of adaptive energy management in our proposal consists of maximizing the use of the available energy supply (mainly from green sources) through the execution (selection) of the best possible variations of services (i.e., for which x_{kie} takes on a value of 1 in Eq. 6.18). As described in Section 5.1.3, this process is analogous to the objective of the 1/0 Knapsack problem of selecting the most valuable items without overloading the knapsack; in consequence, the proposal for adaptive energy management has a complexity level of NP-hard [102].

$$Cost_{comb_f} = \alpha \times Cost_{AR_f} + \beta \times Cost_{L_f} + \gamma \times Cost_{Q_f} + \delta \times Cost_{Ts_f} \tag{6.22}$$

$$Cost_{AR_f} = \begin{cases} 0 & \text{if } AR_f = 100\%, \\ RejServ \times \mathcal{M} \times pri\, RejServ & \text{otherwise.} \end{cases} \tag{6.23}$$

$$Cost_{L_f} = \begin{cases} 0 & \text{if all services have } j = 1, \\ \sum_{i=1}^{N} l_i & \text{otherwise.} \end{cases} \tag{6.24}$$

$$Cost_{Q_f} = \begin{cases} 0 & \text{if all services have } j = 1, \\ \sum_{i=1}^{N} q_i & \text{otherwise.} \end{cases} \tag{6.25}$$

$$Cost_{Ts_f} = \begin{cases} 0 & \text{if all services have } j = 1, \\ \sum_{i=1}^{N} Ts_i & \text{otherwise.} \end{cases} \tag{6.26}$$

$$OptCost =_{Cost_{comb_f} \in AllCost} Cost_{comb_f} \tag{6.27}$$

6.1.4 Heuristic Algorithmic Strategies

Adaptive energy management constrained to the available supply through management mechanisms, such as time-shifting and quality degradation, can be categorized as a 1/0 knapsack problem and specifically in the multidimensional multiple-choice knapsack problem (MMKP) category [102]. In the proposed energy model, the multidimensional property includes/refers to the magnitude and temporality of the supply and energy demands. Simultaneously, the multiple-choice characteristic refers to a time-shifting and/or quality degradation level applied while selecting an energy demand. In this context, we can conclude that the proposed adaptive consumption model has at least the computational complexity of an MMKP problem, proven to be NP-Hard. Since previous work [14] showed that the exact (optimal) method requires excessive execution (over 90 hours), large computational capacity, and has limited practical application (e.g., for $N > 9$, $Ts_i > 4$, or $q_i > 1$), we present a heuristic algorithm strategy to solve the adaptive management problem within a reasonable running time in this section.

6.1.4.1 Prepartitioning-based Strategy: PHRASE

The proposed strategy named Prepartitioning Home eneRgy mAnagement SystEm (PHRASE) is inspired by a divide-and-conquer approach. Instead of performing combinatorial analysis of energy allocation for the entire set of existing services as would be done in the optimal (exact) method, the suboptimal (approximate) PHRASE strategy iteratively analyzes subsets of services. Each subset z is denoted as a $partition_z$ ($partition_z \subset \mathcal{N}$) and has a length $lenPart_z$ that can be different for each partition, although previous work [9] demonstrated that equal size partitions produce better-quality results. The total number of partitions is identified as $NumPart$. For each $partition_z$ (i.e., services belonging to the partition), the strategy PHRASE analyses the application of management mechanisms (time-shifting and quality degradation) to produce variations, compute combinations, and obtain partial solutions. Later, PHRASE merges these solutions to obtain the total allocation of energy resources and the corresponding scheduling of services that enables adaptive energy management. The PHRASE algorithmic strategy for both the offline and online approaches is explained in Fig. 6.4 and Fig. 6.5, respectively. The implementation of PHRASE for the offline approach is summarized as follows:

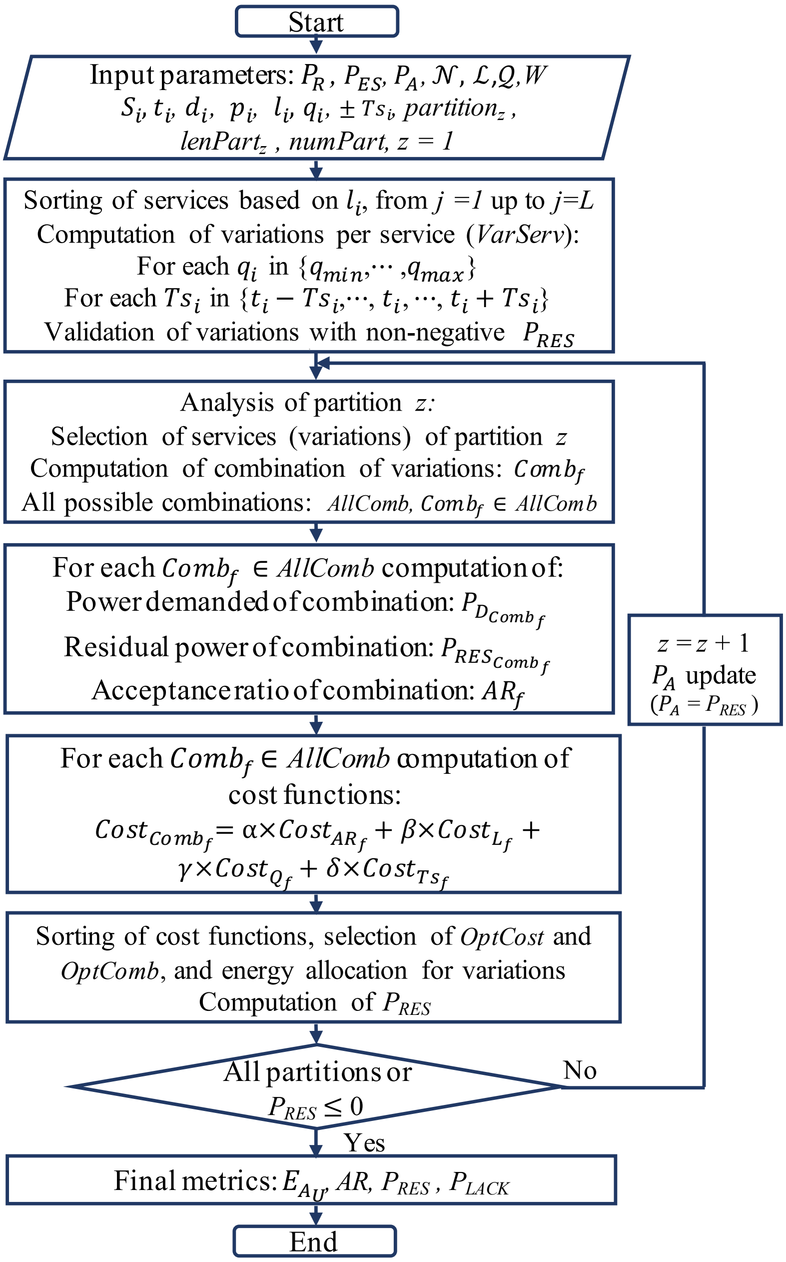


FIGURE 6.4: Flow chart PHRASE offline approach.

1. *Sorting of services and computation of variations:* As a first step, the algorithm sorts the services according to their l_i (from $j = 1$ to $j = L$) to ensure the energy allocation for CS. Then the algorithm computes the variations of services considering all the values of time-shifting and quality degradation. Furthermore, to reduce possible invalid combinations (with $P_{RES} < 0$), the algorithm eliminates variations whose isolated execution produces $P_{RES} < 0$. This procedure improves algorithm performance in terms of running time by reducing the size of the search space for the best combination per partition.

2. *Analysis of partitions and combination of variations:* The algorithm performs combinatorial analysis of variations belonging to each partition to obtain combinations. These combinations are made up of different variations of services. Regarding the complexity of PHRASE, we indicate that combinatorial analysis is dominant because as the values of Ts_i or q_i increases (and in the worst case both), the number of variations increases, and the generation of combinations increase exponentially [14].

3. *Selection of best combination and final performance metrics:* Once the algorithm obtains all the cost functions, it sorts them in increasing order (e.g., through a quicksort method) and selects the first cost in the sorted list (i.e., cost function with the lowest value). This function represents the best combination of services, which allows for optimal service scheduling in the HEMS. The process continues iteratively until all partitions are analyzed. Finally, the algorithm computes the performance metrics.

The implementation of PHRASE for the online approach is summarized as follows:

1. *Initial analysis of services:* Given that for the online approach, the dynamics in terms of demand and energy resources evolve as a function of time slots, the application of the prepartitioning strategy in this environment needs some modifications and additional procedures regarding the offline approach implementation. The first difference is the limited application of forward time-shifting, due to reasons of causality. In this context, the performance for the online approach, in terms of energy utilization, is expected to be lower than the achieved for the offline approach, at most the same (never better). The second change is the differentiation of resources and processing for CS (P_{CS}) and NCS (P_{NCS}) services to ensure CS execution. In this regard, the algorithm assumes that once the service is accepted (in its first slot), there is sufficient energy for its completion (the model does not accept partial processing). The last difference is the inclusion of a list named $waitingList$, which store information on the variations of services that were not processed in their natural starting time (t_i) due to lack of power or because the allocation was performed to higher-priority service.

2. *Analysis for CS:* If service S_i at time slot w is identified as a CS, the algorithm allocates the corresponding energy resource. Then, the algorithm updates the P_{CS} for the remaining services with this priority level.

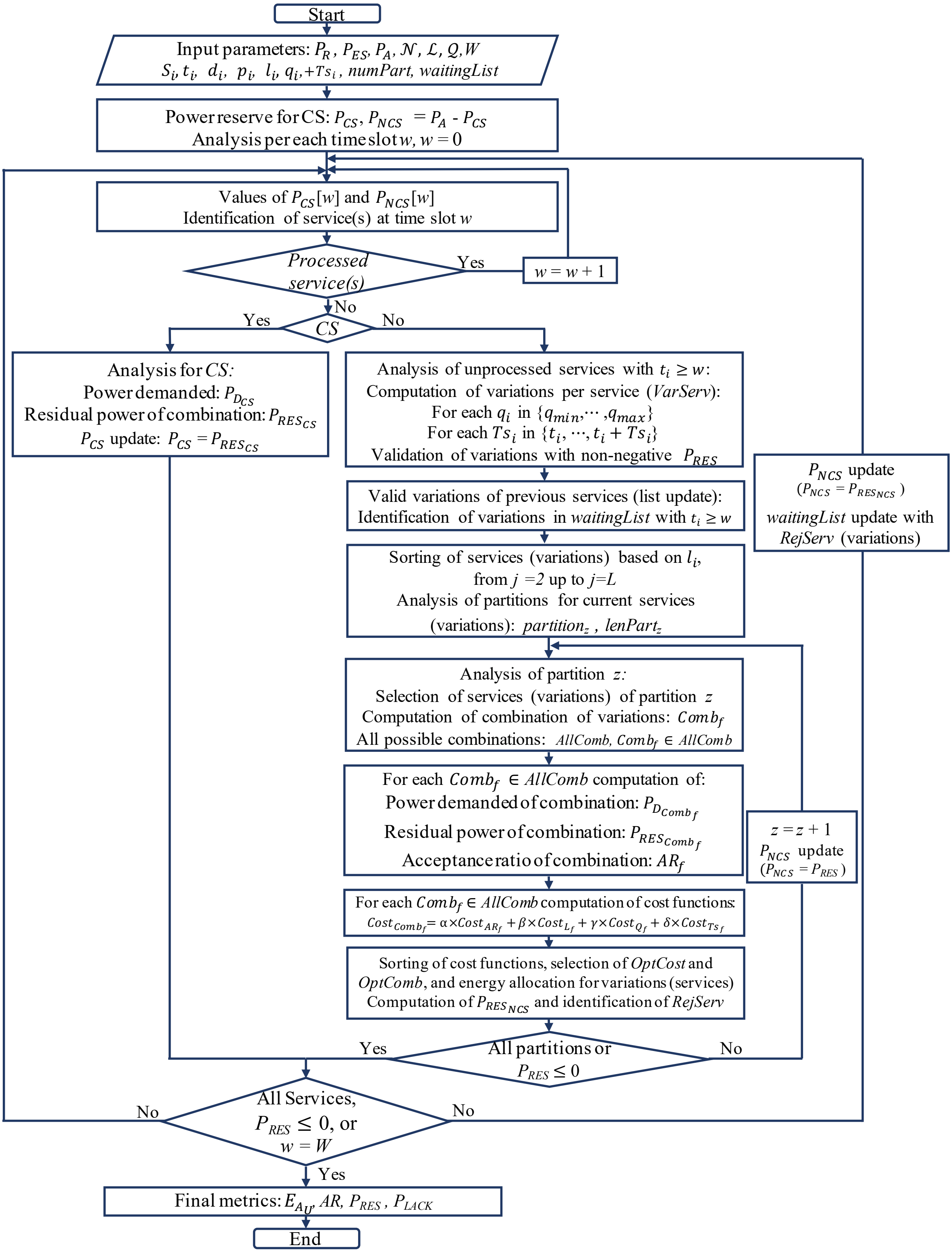


FIGURE 6.5: Flow chart PHRASE online approach.

3. *Analysis for NCS:* If service S_i at time slot w is identified as a NCS, the algorithm performs a similar analysis to carried out for the offline approach, considering the simulated services with the analyzed service and the services (variations) in $waitingList$. Once the algorithm selects the best combination, the energy allocation is made, and the energy resources for subsequent NCS are updated.

4. *Final metrics:* Once the analysis of all the services in the time horizon W has been carried out, the algorithm calculates the performance metrics.

6.1.4.2 Complexity Analysis of PHRASE

The complexity of PHRASE depends on the total number of analyzed combinations in the explored partitions. The most computationally demanding case arises when the approach is offline. Therefore, Eq. 6.28 expresses the complexity of PHRASE as a function of the number of services, the total number of quality degradation levels (Q), and the maximum forward and backward time-shifting value (Ts). In Eq. 6.28, the terms correspond to the original information of the services, the production of variations, and the generation of combinations of services in each partition, respectively. In Eq. 6.28, the last term is dominant and indicates that the growth rate of PHRASE is non-polynomial. However, as evidenced in previous studies [9], the number and size of partitions can be configured (e.g., $lenPart_z = 5$ services) such that complexity becomes tractable (i.e., feasible running time and usage of computational capacity) and hundreds or thousands of times smaller than the strategy without considering prepartitioning (i.e., when the exact method is applied).

$$f(N, Q, Ts) = N + Q \times (2 \times N \times Ts + N) + Q \times \left(\sum_{z=1}^{NumPart} (2 \times Ts + 1)^{lenPart_z} \right) \tag{6.28}$$

6.1.5 Evaluation

In this section, the proposed strategy will be validated and evaluated using a data set from real scenarios in order to show the benefits in terms of the metrics defined, specially regarding the capacity to accept more appliances consuming in the period, while smoothing the peaks of power consumed. In addition, to demonstrate the performance of PHRASE, this section presents a comparison with the optimal solution (i.e., without considering the prepartitioning method) and other similar strategies from the literature. The performance of the proposed heuristic strategies is evaluated according to the simulation setting indicated in Section 5.4.0.1. The results of metrics E_{A_U}, AR, P_{RES}, and P_{LACK} are compared to the baseline scenario, which has no management mechanism. Regarding the number of partitions, they are limited in PHRASE to five because, in previous work [9], this value has demonstrated to offer a trade-off in the running time and the accuracy of the results.

6.1.5.1 Scenario Description

Three scenarios have been considered for the evaluation of PHRASE. The first scenario aims to compare the performance of PHASE with the optimal solution that is obtained when no partitions are used (i.e., when a brute-force search method is implemented). Previous studies [14] have demonstrated that the optimal solution is limited to a small number of services and time-shifting values (e.g., $N < 9$ and $Ts_i < 6$). Consequently, this scenario has been limited to $N = 8$ services (i.e., a small-scale scenario) and $Ts_i = 5 \; \forall \; S_i$. Fig. 6.6 summarizes the generation and consumption (load distribution) profiles used to evaluate the optimal solution and the proposed heuristic.

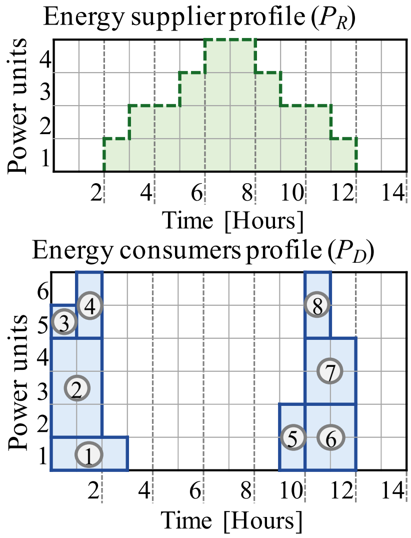


FIGURE 6.6: Power supply and consumption profiles on a small-scale in HEMS.

The second scenario corresponds to a real domestic environment in which the impact of PHRASE on adaptive energy consumption is exhaustively analyzed. For the evaluation of PHRASE in a domestic environment, a data set of real consumption over 24 hours obtained from the model provided in [116] is used as a reference. The consumption information has been adapted to the scope of the proposed adaptive model regarding the customized information of priority, time-shifting, and quality degradation, as summarized in Table 6.3. To simulate renewable energy generation (P_R) in a household, a generation profile following a Gaussian distribution has been used. In this profile, the total available energy is equal to the total energy demanded, and the peak value is approximately 6300 watts. This value is within the range of production of real renewable energy systems for domestic environments, as exemplified in [131]. Besides, to simulate the eventual contribution of the energy utility company to the total available supply (P_A), in the proposed scenario, the use of a small fraction of renewable energy P_{ES}^{R} is considered (although it may also be P_{ES}^{NR}) to ensure the execution of services with the highest level of priority ($j = 1$).

TABLE 6.3: Description of the main HEMS scenario.

Load description	**Quantity**	p_i **[W]**	t_i **[Hour]**	d_i **[Hour]**	l_i	Ts_i **[Hour]**	q_i
Fridge Freezer	1	190	00:00	24	1	0	1
Answer Machine	1	1	00:00	24	1	0	1
CD Player/Radio 1	1	17	06:00	3	1	0	1
CD Player/Radio 2	1	17	15:00	4	1	0	1
Clock	1	2	00:00	24	1	0	1
Phone	1	1	00:00	24	1	0	1
HiFi	1	109	18:00	5	3	0	2 (×0.75)
TV 1	1	127	12:00	2	3	0	2 (×0.75)
TV 2	1	127	18:00	5	3	0	2 (×0.75)
VCR DVD	1	36	20:00	3	3	0	2 (×0.75)
Laptop (charger)	1	146	09:00	8	2	±2	1
Hob 1	1	2401	08:00	1	2	±3	2 (×0.75)
Hob 2	1	2401	18:00	2	2	±3	2 (×0.75)
Oven 1	1	2128	08:00	2	2	±3	1
Oven 2	1	2128	16:00	3	2	±3	1
Microwave 1	1	1252	09:00	1	2	0	2 (×0.75)
Microwave 2	1	1252	13:00	1	2	0	2 (×0.75)
Microwave 3	1	1252	19:00	1	2	0	2 (×0.75)
Kettle 1	1	2001	10:00	1	2	±1	1
Kettle 2	1	2001	15:00	1	2	±1	1
Small Cooking	1	1002	11:00	2	1	0	1
Tumble Dryer	1	2501	08:00	2	2	+8	1
Washing Machine	1	407	17:00	3	2	−8/+4	1
Dish Washer	1	1131	09:00	2	2	±8	1
Lighting 1	2	50	06:00	2	1	0	1
Lighting 2	2	50	17:00	7	1	0	1
Lighting 3 (not indispensable)	5	50	06:00	2	4	0	3 (×0.5)
Lighting 4 (not indispensable)	5	50	17:00	7	4	0	3 (×0.5)

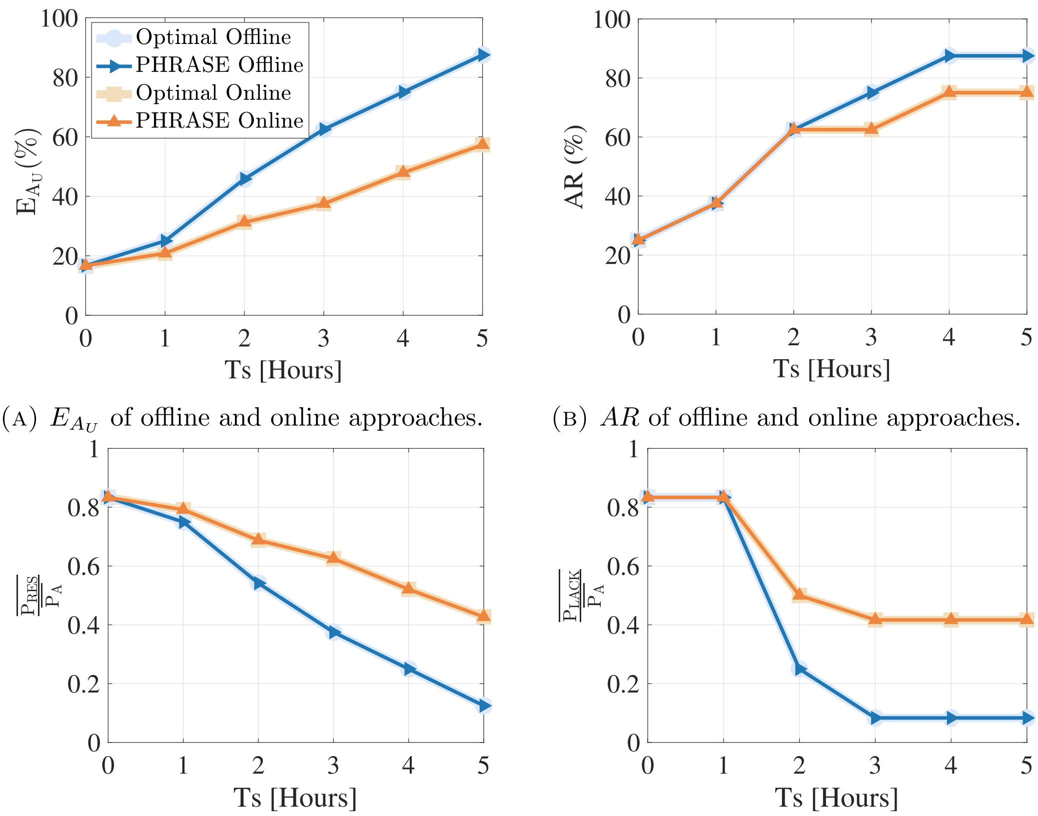
In the last scenario, a brief comparison of the results delivered by PHRASE with other similar approaches in the literature is carried out. To this end, the consumption data in [132] and summarized in Table 6.4 have been used. In this case, all services are NCS with a $l_i = 2$. Moreover, the results of adaptive consumption (reduction of peak power) of PHASE for offline and online approaches are compared with the results obtained with other three evolutionary algorithms-based in [132], that are denoted as Cuckoo (cuckoo search), GA (genetic algorithm), and BPSO (binary particle swarm optimization).

6.1.5.2 Numerical Results

In the small-scale scenario, the evaluation of PHRASE demonstrates that the proposed adaptive model leads to optimized energy consumption, which is reflected in an increase in E_{A_U} and AR and a decrease in P_{RES} and P_{LACK} as time-shifting increases, as shown in Fig. 6.7a, Fig. 6.7b, Fig. 6.7c, and Fig. 6.7d, respectively. Particularly, in this scenario, the evaluation of PHRASE confirms that this strategy produces adaptive consumption and high-quality results compared to those obtained with optimal strategy. In the analyzed scenario, the suboptimal results (from heuristic) are identical to those obtained with the optimal solution with a reduced running time and less computational capacity, as shown in Fig. 6.8.

TABLE 6.4: Description of the HEMS scenario for comparison of results (from [132]).

Load description	p_i **[KW]**	t_i **[Hour]**	d_i **[Hour]**	Ts_i **[Hour]**	q_i
Washing machine 1	1	01:00	1	±5	3 (×0.5)
Washing machine 2	1	13:00	2	±5	3 (×0.5)
Cloth dryer 1	4	01:00	4	±5	3 (×0.5)
Cloth dryer 2	4	09:00	2	±5	3 (×0.5)
Cloth dryer 3	4	19:00	2	±5	3 (×0.5)
Electric vehicle 1	3	05:00	2	±5	3 (×0.5)
Electric vehicle 2	3	09:00	1	±5	3 (×0.5)
Electric vehicle 3	3	20:00	4	±5	3 (×0.5)
Water heater 1	4.5	01:00	1	±5	3 (×0.5)
Water heater 2	4.5	04:00	1	±5	3 (×0.5)
Water heater 3	4.5	07:00	2	±5	3 (×0.5)
Water heater 4	4.5	17:00	5	±5	3 (×0.5)
Refrigerator	1	00:00	24	±5	3 (×0.5)
Lights	1.5	00:00	24	±5	3 (×0.5)



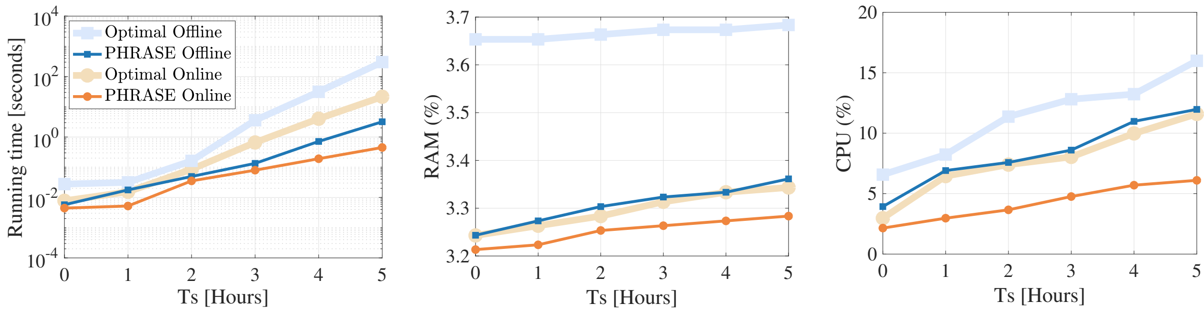
(A) E_{A_U} of offline and online approaches. (B) AR of offline and online approaches.

(C) P_{RES} of offline and online approaches. (D) P_{LACK} of offline and online approaches.

FIGURE 6.7: Summary of performance metrics obtained by optimal solution and PHRASE strategy in the online and online approaches considering the maximum quality degradation level $q_i = 3$ (×0.5) and two partitions.

Although in diverse scenarios, the performance of PHRASE may be lower than that of the optimal solution, its lower complexity (adjustable by varying the number of cabinets) makes it a feasible solution in a variety of application environments such as the HEMS.

As an example of comparison, Fig. 6.8a reports that PHRASE (for the offline approach) runs over 90 times faster than the optimal solution. Partitions make it possible to exceed the limits imposed on the optimal solution regarding the number of services.



(A) Running time. (B) RAM Usage. (C) CPU Usage.

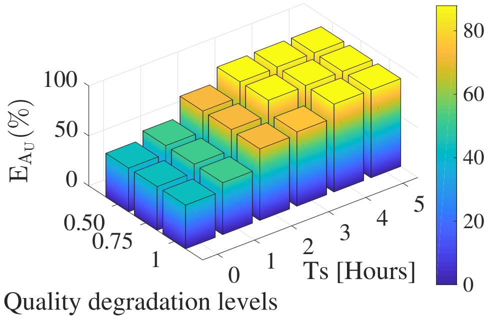
FIGURE 6.8: Running time and computational capacity used by the optimal solution and PHRASE strategy in the offline and online approaches considering a maximum quality degradation level $q_i = 3$ and a single core.

For the HEMS scenario in Table 6.3, the simulation results show that the proposed adaptive model and the PHASE strategy lead to optimized green energy consumption. As time-shifting and quality degradation levels increase, the algorithmic strategy improves the ability to allocate the consumption demands, preventing unnecessary energy waste. For practicality, Fig. 6.9 only shows the simultaneous action of time-shifting and quality degradation mechanisms for the offline approach. Fig. 6.9a, Fig. 6.9b, Fig. 6.9c and Fig. 6.9d show an increase in E_{A_U} and AR values and a decrease in P_{RES} and P_{LACK}, respectively, indicating that the P_A utilization improves with an increase in time-shifting and/or quality degradation value. The consumption profiles before (baseline scenario) and after the application of PHRASE is shown in Fig. 6.10a and Fig. 6.10b, respectively.

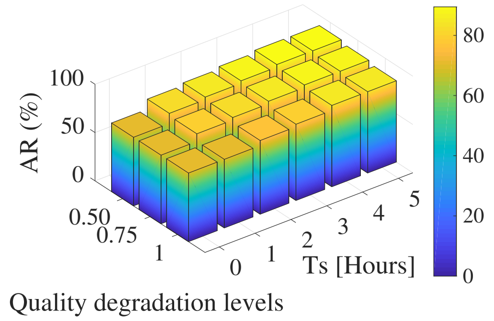
To compare the operation in offline and online approaches, the analysis of PHRASE is carried out for the maximum level of quality degradation (i.e., $q_i = 3\,\forall i \in \mathcal{N}$) and considering a variation in time-shifting (from 0 to 5 hours), as shown in Fig. 6.11. Simulation results for these conditions report that PHRASE achieves better metric values in the offline approach, as shown in Fig. 6.11a Fig. 6.11b, Fig. 6.11c, and Fig. 6.11d. This difference in the online approach obeys the limited use of forward-time shifting because of the causal principle and the lower amount of simultaneous variations analyzed. The metric results, specifically the AR, show the limitation of the PHRASE strategy, as the scenario conditions are designed to reach an optimal solution ($AR = 100\%$). A more sophisticated method might achieve an optimal solution. Therefore, future research could address the development and evaluation of adaptive solutions based on genetic algorithms or dynamic programming.

The metrics E_{A_U} and AR summarize the effectiveness of PHRASE in energy consumption. Fig. 6.11a indicates that the E_{A_U} metric shows a 44.17% improvement (from 43.71% to 87.88%) and 30.3% improvement (from 43.71% to 74.01%) for offline and online approaches, respectively. The AR metric improves by 18.42% (from 71.05% to 89.47%) and 15.79% (from 71.05% to 86.84%) for offline and online approaches, respectively, as seen in

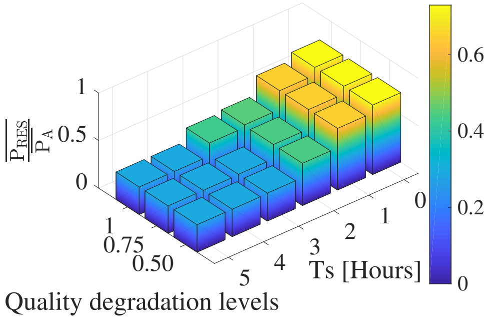
Fig. 6.11b. Although the improvement in AR does not appear to be significant because it is produced by the rejection of several small energy demands (with the lowest priority), the real operation of PHRASE is supported by the value of the E_{A_U} metric. Regarding the reduction of power peaks, Fig. 6.10b shows that PHRASE produces a reduction of 25.58% and 15.3% for offline and online approaches, respectively.



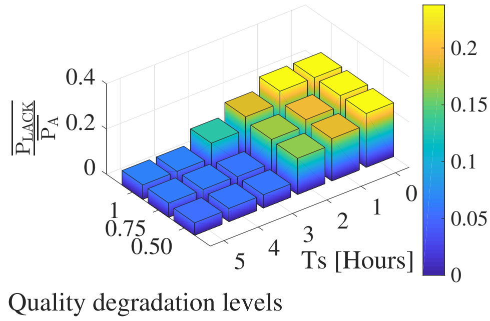
(A) E_{A_U} of the offline approach.



(B) AR of the offline approach.

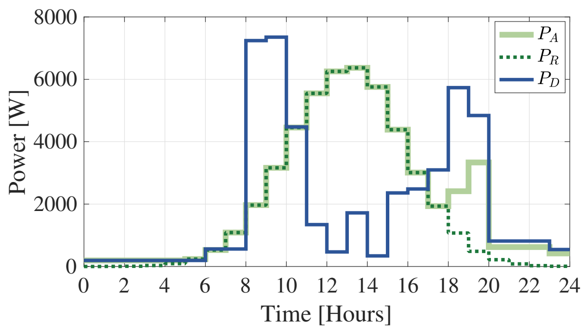


(C) P_{RES} of the offline approach.

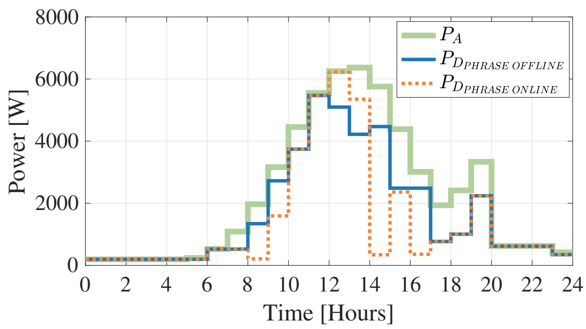


(D) P_{LACK} of the offline approach.

FIGURE 6.9: Performance evaluation of PHRASE for the domestic scenario described in Table 6.3 and metrics obtained of the offline approach.
The simulation of PHRASE in the offline approach exploits parallel processing, and the total running time for all values of Ts_i and q_i is 352.57 seconds.



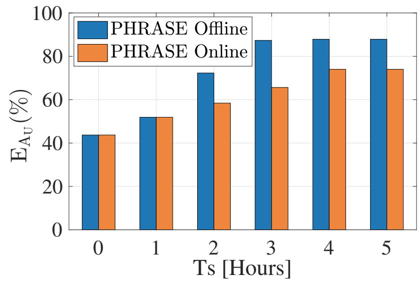
(A) Supply and consumption profiles before the application of the PHRASE strategy.



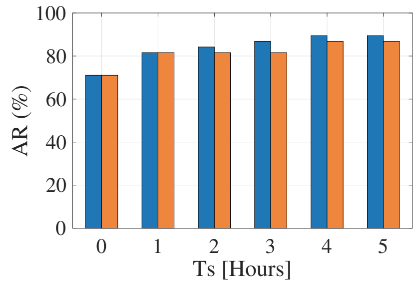
(B) Supply and consumption profiles after the application of the PHRASE strategy.

FIGURE 6.10: Comparison between the baseline scenario and PHRASE application in offline and online scenarios considering the maximum values of Ts_i and q_i.

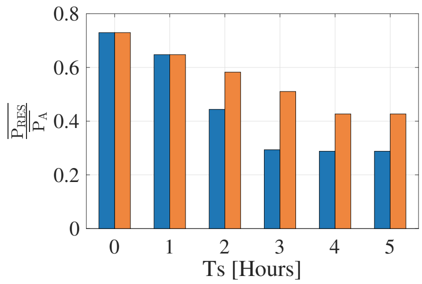
On the other hand, Fig. 6.12 reports that PHRASE solves the most demanding case ($Ts_i = 5$ and $q_i = 3$) with a running time of 165.31 seconds in the offline approach with the maximum usage of 4.69% RAM and 18.16% CPU. For the online scenario, the maximum running time is approximately 1 second using a maximum of 4.38% of RAM and 16.01% of CPU. The results in Fig. 6.12a, Fig. 6.12b, and Fig. 6.12c show that PHRASE can be executed with a reasonable running time using a small amount of computational capacity. Thus, this strategy can be applied to plan renewable and non-renewable energy consumption or energy management in real-time scenarios. Furthermore, the strategy could be deployed into an embedded device with limited computational resources such as a Raspberry Pi 3 Model B platform [141].



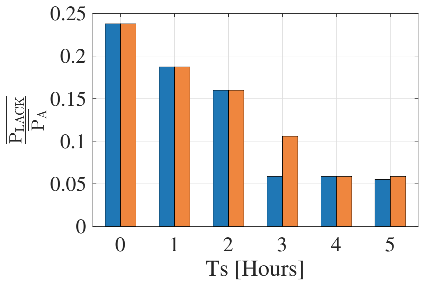
(A) E_{A_U} of offline and online approaches.



(B) AR of offline and online approaches.

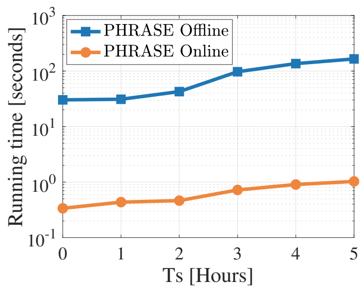


(C) P_{RES} of offline and online approaches.

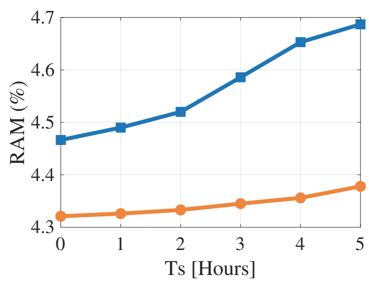


(D) P_{LACK} of offline and online approaches.

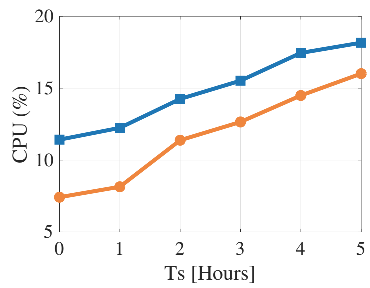
FIGURE 6.11: Summary of performance metrics obtained by the PHRASE strategy in the online and online approaches considering the maximum quality degradation level $q_i = 3$ ($\times 0.5$).



(A) Running time.



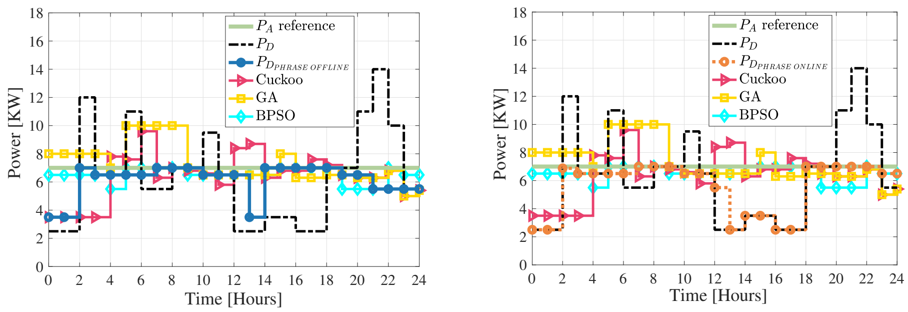
(B) RAM Usage.



(C) CPU Usage.

FIGURE 6.12: Running time and computational capacity used by the PHRASE strategy in the offline and online approaches considering a maximum quality degradation level $q_i = 3$ and a single core.

Regarding the comparison with existing strategies in the literature, the results of the third scenario in Fig. 6.13 report that PHRASE produces solutions similar to more sophisticated approaches such as those based on evolutionary algorithms [132]. The results in this scenario show that even though the structure of PHRASE is simple (based on prepartitioning), its internal management mechanisms (mainly time-shifting and quality degradation) allow obtaining an efficient adaptive consumption for offline (Fig. 6.13a) and online (Fig. 6.13b) approaches. Specifically, the results in Fig. 6.13 indicate that PHRASE, like the existing strategies (Cuckoo, GA, and BPSO), enables peaks power reduction. Even as shown in Fig. 6.13a and Fig. 6.13b, the proposed heuristic compared to the others strategies analyzed offers a lower level of power peaks (lower than 7[KW]) and a consumption conditioned to availability (P_A) throughout the time horizon. Moreover, PHRASE produces an improvement in service processing (use of P_A) of 42.86% (from $AR = 57,14\%$ up to $AR = 100\%$) for the offline approach and 35.72% (from $AR = 57,14\%$ up to $AR = 92.86\%$) for the online approach, verifying the validity of the model in terms of adaptive consumption conditional on availability.



(A) E_{A_U} of the offline approach. (B) AR of the offline approach.

FIGURE 6.13: Performance evaluation of PHRASE for the scenario described in Table 6.4 considering two partitions and comparison with similar approaches in [132].

6.1.6 Conclusions

This section proposes an IoT-enabled automated and adaptive HEMS that optimizes self-generated renewable energy utilization, which can use as a secondary source the provisioning from the energy utility company if necessary. The proposal includes a description of the architecture, the negotiation scheme for the consumption of IoT devices, and management mechanisms, such as time-shifting and quality degradation, to adapt the demand to the available power while maximizing its utilization. This section also provides the mathematical formulation associated with the adaptive consumption model. To solve the energy model, a heuristic called PHRASE is provided, which bases its operation on a divide and conquer approach.

To verify the validity of the proposed system and the operation of PHRASE, a simulation is carried out in a domestic environment based on real consumption data generated from [116]. The results of the metrics E_{A_U}, AR, P_{RES}, and P_{LACK} in the simulation performed reveal that the proposed energy model and algorithmic strategy deliver improvements in a way that available energy is used compared to the baseline scenario in which no management strategies are applied. Particularly, if PHRASE uses the maximum values of Ts_i and $q_i = 3$ (i.e., $Ts_i = 5$ and $q_i = 3$), E_{A_U} improves by 44.17% and AR by 18.42%, and the peak power is reduced by 25.58% for the offline scenario. In the online scenario, the improvement in E_{A_U} is 30.30% and AR is 15.79%, while the power peak is reduced by 15.30%.

Regarding the running time of the algorithmic strategy and computational resources used, the results of the simulations indicate that for the offline scenario, the maximum running time reached is 165.31 seconds (for $Ts_i = 5$ and $q_i = 3$) using a maximum of 4.69% of RAM and 18.16% of CPU. For the online scenario, the maximum execution time (for $Ts_i = 5$ and $q_i = 3$) is approximately 1 second using a maximum of 4.38% of RAM and 16.01% of CPU. The online scenario results demonstrate the feasibility of PHRASE in real-time applications and the possible deployment in current embedded devices of limited computational capacity. Therefore, it can be implemented in low cost devices, and attached to the smart meter.

Although the PHRASE algorithm produces improvements in renewable energy consumption in a reasonable running time, other strategies, such as genetic algorithms or dynamic programming, could be evaluated in future work. Furthermore, since the online operation of PHRASE is limited to the use of forward time-shifting, a prediction mechanism of the generated supply could be included in the adaptive model to potentially expect better energy allocation and demand processing. To this end, different techniques could be used, among which we can mention supervised learning techniques such as random forest or artificial neural networks, which are pretty popular today. Also, despite the proposed system encouraging battery-free adaptive management, a battery unit's possible inclusion could serve to store energy in periods of abundance and provide energy if the P_R is not sufficient to meet all demands. This would not only improve system performance but also reduce dependence on energy utility provisioning.

6.2 Adaptive Energy Management in 5G Network Slicing

Energy consumption is a critical issue for the communications network operators, impacting deeply the cost of the services, as well as the ecological footprint. Network slicing architecture for 5G mobile communications enables multiple independent virtual networks to be created on top of a common shared physical infrastructure. Each network slice needs different types of resources, including energy, to fulfill the demands requested by each application, operator, or vertical market. The existing literature on network slicing is mainly targeted at the partition of network resources; however, the corresponding management of energy consumption is an unconsidered critical concern. This section analyzes the requirements for an energy-aware 5G network slicing provisioning according to the 3GPP specifications, proposes an architecture, and studies the mechanisms to provide efficient energy consumption in terms of renewable and non-renewable sources. NFV and SDN technologies are the essential enablers and leverage the IoT connectivity provided by 5G networks. This section also presents the technical 5G technology documentation related to the proposal, the requirements for adaptive energy management, and the ILP formulation of the energy management model. To validate the improvements, an exact optimal algorithmic solution is presented.

6.2.1 Introduction

6.2.1.1 Background and Motivation

The deployment of 5G mobile networks introduces new services and applications to facilitate a wide range of end user demands. However, before these innovations can be made available to customers and vertical markets, some challenges need to be addressed, such as the efficient management of network resources, multi-tenancy approaches, and the management of energy consumption [142, 143]. Regarding the first two points, network slicing architecture has emerged as a means to efficiently support dynamic resource management in a multi-tenant environment [144]. Specifically, network slicing for 5G mobile communications leverages on the concepts of NFV and SDN to implement multiple virtual and independent logical networks, referred to as network slices, on a common shared physical network infrastructure [145, 146]. In network slicing, each slice (virtual network) is an isolated amount of end-to-end network resources and functions with different requirements, including energy, and with independent management and control, tailored to fulfill the diverse demands requested by a particular operator, application, service, customer, or vertical market [147].

The Mobile Network Operator (MNO) can configure and manage the control plane and user plane network functions and the corresponding resources (e.g., access, transport, and core networks) to support various Slice/Service Types (SST) [148]. These SST can be grouped according to their different requirements in functionality (e.g., priority, charging, security, and mobility), on performance requirements (e.g., reliability, latency, mobility, and data rate), or they can be targeted to specific users (e.g., public safety users, corporate customers, or virtual operators) [143]. In the technical specification TS 23.501 v16.4.0 (issued in March 2020), the 3rd Generation Partnership Project (3GPP) provides a standardized classification that groups different services, such as enhanced Mobile Broadband (eMBB) services, Ultra-Reliable Low-Latency Communications (URLLC) services, massive Internet of Things (IoT) services (MIoT), and Vehicular-to-everything communications (V2X, where the X means vehicle, infrastructure, pedestrians, etc.), within four SST categories [142], as shown in Table 6.5. This classification can be used as a baseline or template to implement network slices for most customer requirements. In addition, the TS 23.501 v16.4.0 does not limit the creation of other categories if necessary, and the assigned SST values do not give priority to one category over the others.

Table 6.5: Standardized SST values and examples [142].

Slice/Service type	**SST value**	**Characteristics**	**Examples of services**
eMBB	1	Slice suitable for the handling of 5G enhanced Mobile Broadband	4K/8K UHD, hologram, augmented/virtual reality
URLLC	2	Slice suitable for the handling of ultra reliable low latency (*e.g.*, 1 ms) communications	High-accuracy positioning systems, motion control, autonomous driving, automated factory, smart-grid service, augmented/virtual reality
MIoT	3	Slice suitable for the handling of massive IoT	Sensor-network services (*e.g.*, metering, agriculture, building, logistics, cite, home, etc.)
V2X	4	Slice suitable for the handling of V2X services	Vehicular communications systems (*e.g.*,vehicle-to-vehicle, vehicle-to-infrastructure, , etc.)

Regarding energy consumption management, this feature is a key factor in the deployment and evolution of mobile networks [149], and it is also a crucial consideration for the following reasons: (i) constant growth in energy consumption because of the increasing number of devices connected to mobile networks and the corresponding network densification from the deployment of a high number base stations and related infrastructures (legacy networks 2G, 3G, and 4G must coexist with 5G and beyond networks) [150]; (ii) increased traffic demand and heterogeneity of services with different requirements (e.g., high data rate, low latency, wide bandwidth, a high operating frequency of up to 60 GHz, reliability, or connectivity) [151]; (iii) increase in the OPEX for the MNO, because, more energy consumption means more costs of energy supply, which can produce an impact on tariffs for consumers or less margins for the operators [149]; and (iv) sustainability issues, which call for new energy generation and consumption principles [151].

Historically, the evolution of mobile networks has implied an increase in energy consumption. However, this current reality needs to change due to sustainability considerations, the increase in CO_2 emissions, and the impact on climate change caused by the use of fossil fuels for energy production (e.g., electricity) [152]. To guarantee energy sustainability and efficiency for mobile communications, there are different solutions that fall into two major groups: (i) increasing the use of renewable energy sources; and (ii) optimizing energy consumption (e.g., using energy saving mechanisms) [153].

The use of renewable energy sources (e.g., solar and wind), also known as green energy, to power ICT systems such as 5G networks, is a promising alternative that can reduce energy bills for the MNO and customers, the exclusive dependence on power grids, and is an opportunity to deliver a cleaner and more sustainable mobile communications ecosystem [151]. For instance, green energy can allow the deployment of base stations in remote areas where power grids are not available (e.g., photovoltaic installations), and it may be a choice to compensate the lack of energy capacity from the supplier. In addition, the adoption of renewable energy sources allows changing the traditional centralized energy supply scheme to distributed power grid architectures in which the energy harvesting processes can be used to improve energy distribution, as it has been demonstrated for mobile access networks [154]. For all these reasons, the use of green energy emerges as a feasible solution to deal with the ever-rising energy demand and sustainability requirements [153].

There are multiple benefits of using energy from renewable sources, but their intermittent nature may affect the continuity of services and the reliability of mobile networks. In addition, the use of green energy alone is not enough to make mobile networks more sustainable; they need to consume better and less. Consequently, there is a need to incorporate new methods for adaptive energy management consumption (i.e., mechanisms to adapt consumption to availability) [142]. In this regard, different strategies have been analyzed such as activation of network infrastructures on demand, the total or partial deactivation of services or devices (e.g., deactivation of sectors in base stations), periodical activation or deactivation of consumption (e.g., by using sleep or idle modes), degradation in the quality of service (e.g., decrease in transmitting power), or scale the energy consumption to the traffic dynamicity [155, 156]. These strategies can be applied to specific segments (e.g., to access networks) or to the whole system, and they can be performed through the ICT infrastructures of mobile networks [155]. In 5G, for example, the enabling technologies NFV and SDN can be used to deploy an energy management framework [157, 158], which offers high computing capabilities (data centers or cloud computing infrastructures), flexibility, and agility for executing management strategies (NFV benefits) [146], as well as the separation of control and data planes (SDN benefits) [30], and they can be applied for energy managing for different customers and scenarios transparently. Moreover, an NFV/SDN-based energy management architecture can potentially be used for managing virtual resources and networks [157, 158], a capability that can enable energy management in 5G network slicing.

In summary, new services and applications demanded by customers impose that the current and future mobile networks have sophisticated energy management and consumption schemes. These energy management approaches must be adapted to finite energy production and the dynamic generation–consumption conditions. Natively, mobile networks lack an efficient energy management scheme for the whole system. In addition, the redundant design of mobile networks (e.g., duplication of access, transport, or core devices) for keeping the reliability and performance in communications has produced a constant increase in energy consumption (mainly for non-renewable sources) and carbon footprint related. To promote sustainable and environmentally friendly development of mobile networks, the traditional network design and operation must incorporate efficient management of energy consumption and prioritize the use of renewable energy sources. In this regard, the evolution to 5G and specifically to network slicing is an opportunity to develop a sustainable and adaptive energy management ecosystem that is capable of working in a multi-tenant approach, encourages the use of green energy, and optimize the energy consumption.

6.2.1.2 Contributions

This section proposes an energy management solution applicable to 5G network slicing in which the service processing (i.e., the creation of network slices and corresponding services) is aware of the energy supply with the aim of optimizing power consumption, specifically by minimizing the power consumption from non-renewable energy sources. The proposal considers renewable energy sources, but non-renewable sources can also be used when extra energy is required. Efficient energy management involves minimizing consumption, achieving a specific reduction in consumption (e.g., 20% energy saving), or optimizing the available power consumption (utilization). In this work, we chose the last option, anticipating a future in which the penetration of green energy allows meeting all energy demand and requires adaptive mechanisms to leverages its generation. Moreover, we validated this approach in [9]. The considerations of the architectural framework and the enabling technologies needed for energy management, as well as the interaction between generation and consumption sides presented in [9], are taken as a baseline for the adaptive energy management solution presented in this section. In this regard, this section represents an evolution of the work in [9] towards an efficient and environmentally friendly energy-management for mobile networks.

The proposal in this section includes: (i) an architecture for adaptive energy management that considers provisioning from renewable and non-renewable sources, which is developed based on NFV and SDN technologies, and leverages the IoT massive connectivity provided by modern mobile networks; and (ii) various management mechanisms at service level such as an intra-slice prioritization scheme, service rejection, time shifting in service execution, and degradation in service quality (i.e., decrease in energy demand), all to optimize the available power consumption. In addition, this section provides modeling of the stakeholders involved in energy management and consumption (i.e., the MNO and consumers), and the ILP formulation corresponding to the problem of the adaptive energy management. To find the exact optimal solution, a brute-force search algorithmic strategy (OptTsNS) is proposed. Simulation results demonstrate that the proposed energy management solution allows for efficient energy consumption in the different slices and corresponding services in accordance with the available energy resources and customer requirements, thereby realizing feasible and efficient energy management for 5G networks.

6.2.2 Related Work

This section presents related work that addresses the two main areas of the proposal, i.e., the energy efficiency and energy management in 5G networks, and the use of network slicing architecture for resource management, including energy.

6.2.2.1 Energy Efficiency and Energy Management in 5G Networks

Improving energy efficiency has become a key pillar in the design of 5G networks due to economic and operational considerations, as well as environmental concerns [159]. In this regard, different solutions have been proposed to optimize or reduce energy consumption, which can be grouped under following broad categories: (i) resource allocation; (ii) network planning and deployment; (iii) activation of resources on-demand depending on traffic dynamic; (iv) hardware design; (v) improvements in the system operation (e.g., techniques to reduce interference); (vi) active user cooperation; and (vii) use of green energy complemented with energy-harvesting mechanisms (which can also be used to collect energy from radio signals over the air) [159]. Of these approaches, the use of green energy has gained momentum in recent years, and it represents a feasible and sustainable alternative to power mobile networks partially or even totally [160]. In this respect, different models have been proposed to characterize the production of renewable energy and the operation in the mobile networks. Even the possible interactions with smart grids to deliver demand-response schemas have been analyzed. Additionally, the need for management strategies in mobile networks to enhance integration and use of green energy has been evidenced [160], an issue that is solved with our proposal.

Regarding energy management within the 5G ecosystem, the literature shows that the enabling technologies NFV and SDN can be used as a platform to deploy optimization models (mainly based on heuristic approaches) and management applications targeting cost-efficient resource and energy usage [161]. Based on energy consumption estimations or network parameters information (e.g., traffic load, radio coverage, equipment activation intervals, or active users), the NFV/SDN architectural framework can carry out actions such as optimized routing of traffic flows or allocation of physical (networking, computing, and storage) and/or virtual (e.g., virtual machines) resources with the aim of achieving energy savings and an overall reduction of consumption in the mobile network [80, 81]. In addition, NFV technology facilitates in 5G networks that the VNFs can be dynamically scale-in/out to meet the desired performance level, a dynamic behavior, or to be adjusted to system capacity. These features can potentially reduce energy consumption, operating cost, and latency. In this regard, some adaptive and dynamic VNF scaling algorithmic strategies have been proposed in the literature [80, 81]. Although these procedures can reduce the energy footprint of 5G networks, they operate primarily on the network infrastructure and do not constitute an adaptive energy management system. Furthermore, these solutions do not consider the available energy supply for service processing or a multi-client approach, aspects which are considered in our proposal.

6.2.2.2 Use of Network Slicing Architecture for Resource Management

The network slicing architecture supported by NFV and SDN has demonstrated to be an effective solution for implementing resource allocation schemes and algorithms to meet the diverse and simultaneous demands of consumers and vertical markets [75]. For instance, network slicing can be used as a management solution to enhance the network resources sharing required for the dynamic operation of massive IoT infrastructures such as wearable devices [162]. The network slicing paradigm can be used to define entire network slices to cover the elastic demand for network resources through the day and according to different operating and customer requirements such as bandwidth or a desired reliability level for the customers [163]. The reconfiguration, scaling, and migration of virtual resources (e.g., VMs) needed for the dynamic operation of mobile networks corresponding to customer-specific workloads are considerations that can also be addressed efficiently (e.g., in terms of bandwidth and latency constraints) with network slicing technology [164]. In addition, the network slicing architectural framework is flexible enough to allow discrimination in network resources allocation among slices, customers, or services according to specific operational requirements; a feature can be exploited in different scenarios and for various purposes. In [165], for example, the authors proposed a two-level prioritization scheme (inter-slice and intra-slice) to implement a heuristic-based admission control mechanism able to dynamically allocate network resources to different slices customers needs and traffic loads. In our proposal, we also use an intra-slice scheme but focused on prioritizing the consumption of certain services if the energy supply is not enough to meet all demand.

The potential of network slicing for energy management has also been explored. In [166], Xiao et al. introduced a dynamic network slicing solution for large-scale energy-harvesting fog computing networks. In the proposed architecture, a regional orchestrator coordinates workload distribution among local fog nodes, providing slices of energy and computational resources to support various types of service requested by end users. The use of network slicing in this use case shows a maximization of the utilization of available resources, dynamic resource allocation according to service demands, and balance of workloads among fog nodes. This information provides insight into the possible improvements in energy efficiency that can be obtained with network slicing in 5G networks.

6.2.3 Energy Management Proposal for 5G Network Slicing

Section 6.2.3.1 describes the requirements for energy management in the context of 5G network slicing. Section 6.2.3.2 discusses the management mechanisms for adaptive energy consumption. Finally, Section 6.2.3.3 presents an overview of the proposed architecture and its operation.

6.2.3.1 Requirements for the Energy Management in 5G Network Slicing

To present a feasible proposal aligned with the needs of current and future mobile networks, we surveyed the main technical specifications and recommendations issued by the most representative standardization bodies in the mobile networks landscape: the 3GPP, the European Telecommunications Standards Institute (ETSI), and the Telecommunication standardization sector of the International Telecommunication Union (ITU-T). Table 6.6 shows the latest version of the technical specifications and reports reviewed related to the proposal. Regarding the proposed energy management approach, Table 6.7 summarizes the requirements and possible solutions for adaptive consumption in 5G network slicing.

Table 6.6: Technical documentation related to the proposal.

Area	Institution or organization	Documentation	Description
Requirements and features of 5G	3GPP	Technical specification TS 23.501, release 16	System architecture for the 5G System [143]
		Technical specification TS 22.261, release 17	Service requirements for the 5G system [142]
Network Slicing	3GPP	Technical specification TS 28.530, release 16	Management and orchestration; Concepts, use cases and requirements [144]
		Technical specification TS 28.531, release 16	Management and orchestration; Provisioning [145]
		Technical report TR 28.801, release 15	Study on management and orchestration of net. slicing for next generation network [147]
		Technical report TR 28.804, release 16	Study on tenancy concept in 5G networks and network slicing management [148]
	ETSI	Technical report GR NFV-EVE 012	Report on Network Slicing Support with ETSI NFV Architecture Framework [146]
Energy efficiency, management, and renewable energy sources	3GPP	Technical specification TS 28.310, release 16	Energy efficiency of 5G [149]
		Technical report TR 32.972, release 16	Study on system and functional aspects of energy efficiency in 5G networks [150]
	ITU-T	Recommendation L.1210	Sustainable power-feeding solutions for 5G networks [151]
		Recommendation L.1310	Study on methods and metrics to evaluate energy efficiency for future 5G systems [153]
		Recommendation L.1315	Standardization terms and trends in energy efficiency [155]
		Recommendation L.1331	Assessment of mobile network energy efficiency [156]
		Recommendation L.1360	Energy control for the software-defined networking architecture [157]
		Recommendation L.1361	Measurement method for energy efficiency of network functions virtualization [158]

Table 6.7: Requirements and proposed solutions for adaptive energy management in 5G Network Slicing.

Requirements: ID	Requirements: Description	Related technical documentation	Proposed solution
R1	Energy management for network slices and corresponding services throughout all components of the 5G system, in which the service processing is aware of energy supply with the aim of optimizing the consumption at all times.	3GPP TS 23.501 3GPP TS 22.261 3GPP TS 28.310 ITU-T L.1331	Adaptive energy management solution for 5G network slicing, in which the creation and configuration of networks slices is aware of the energy generation and consumption conditions.
R2	Definition of a priority order for services within a network slice in case the available energy at certain time period is insufficient to meet all demand. Based on this level of priority, the MNO must be able to differentiate the access to the energy resource and the application of management strategies for each service.	3GPP TS 22.261 3GPP TS 28.530 3GPP TS 28.531 3GPP TR 28.801 3GPP TR 28.804 ETSI NFV-EVE 012	Prioritization scheme at service level (intra-slice priority scheme) for the energy allocation process. In proposed architecture, the MNO decides the number of priority levels to implement and the assigned level to each service. The assignation of priority level can also be established through agreements with customers.
R3	Use of software-based platforms and virtualization to facilitate the creation and managing of network slices, the configuration of network functions to implement energy management strategies, and the incorporation of features for serving new SST.	3GPP TS 23.501 ITU-T L.1360 ITU-T L.1361	Use of NFV and SDN technologies, to bring agility, flexibility, and reconfigurability to the proposal.
R4	Use of energy from renewable energy sources, as a sustainable and environmental friendly alternative for powering mobile networks, and control of its contribution in the total energy supply.	3GPP TS 28.310 3GPP TR 32.972 ITU-T L 1210	Promotion and prioritization of the use of green energy, and implementation adaptive energy management strategies (see Section 6.2.3.2) to leverage its dynamic and intermittent generation.
R5	Energy management constrained to the availability. The proposed energy management model must be able to automatically adapt the consumption (*i.e.*, configuration of network slices) to the energy capacity, producing no or minimal impact on performance of network slices and corresponding services.	3GPP TS 23.501 3GPP TS 22.261 TS 3GPP TS 28.310	Implementation of strategies for adaptive energy consumption on network slices or services (see Section 6.2.3.2). In the proposed architecture, the customer is aware of the configuration of network slices and services (*e.g.*, modification of execution time, power demanded, or activation/deactivation state) performed by the MNO to optimize the energy consumption.
R6	Implementation of mechanisms to achieve energy efficiency through optimization of energy use and reduction in consumption (*i.e.*, energy saving). These mechanisms must be executed by the MNO and must produce no or minimal performance degradation for customers. Services belonging to network slices must be able to work in energy-saving states (*e.g.*, low energy consumption) or postpone their execution (*i.e.*, enter into sleep or idle mode) according the availability.	3GPP TS 22.261 3GPP TS 28.310 3GPP TR 32.972 ITU-T L.1310 ITU-T L.1315 ITU-T L.1331	In order to offer an adaptive energy management solution that optimizes the use of energy resources and reduce consumption, the following management strategies for services are considered in the proposed management model: (i) degradation of service quality, which represents a decrease of consumption (*i.e.*, low power consumption), (ii) rejection or non-execution of service(s) (*i.e.*, deactivation of service), and (iii) application of time shifting (forward or backward) in the service execution time.
R7	Definition of metrics or indicators to assess the energy efficiency achieved with the proposed energy management model.	3GPP TR 32.972 ITU-T L.1310 ITU-T L.1315	Definition of three performance metrics: (i) percentage of services processed, (ii) percentage of energy used by services processed, and (iii) amount of missing energy to process all services.

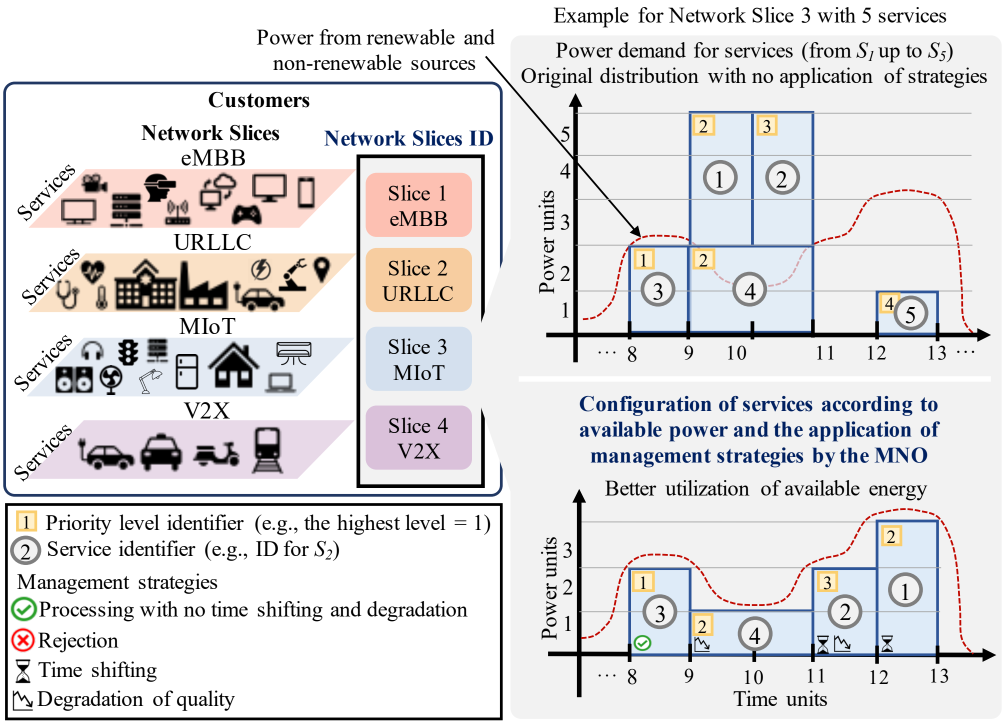
The structure of the proposed architecture and the energy management model described in Section 6.2.3.3 allow satisfying Requirements R1, R2, R3, and R4. The management strategies presented in Section 6.2.3.2 meet the need for adaptive consumption of available energy described in Requirements R5 and R6. Instead, the performance metrics to assess the optimization in energy consumption described in Requirement R7 are presented in Section 6.2.5.

6.2.3.2 Energy Management Mechanisms for Adaptive Consumption

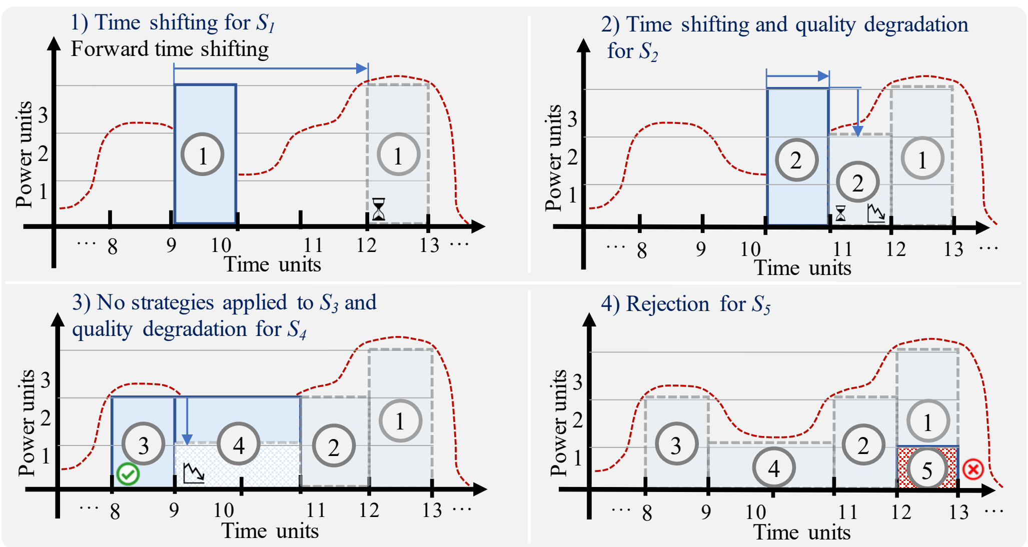
To carry out adaptive energy management in 5G network slicing, five mechanisms have been considered. They have been previously presented in Section 4.1.2 and Section 6.1.3.3 and are adapted to the context of network slicing. These mechanisms are executed by the MNO, are implemented through algorithmic solutions, and they seek offer energy efficiency to the 5G mobile networks through optimization of power consumption (i.e., less and better energy use), while promoting the reduction of OPEX and related environmental impacts. In the energy management process, the customers are aware of the configuration of network slices and/or services performed by the MNO to optimize the consumption, and they are tolerant of the possible modifications/actions on network slices or services. Figure 6.14 shows an example of the implementation of the proposed strategies and the corresponding description is shown below.

- *Prioritization of services in network slices:* In the energy management proposal, the MNO can establish priority levels to differentiate the services, as shown in Fig. 6.14a. With this information, the MNO can prioritize the resource allocation (e.g., energy) for the configuration of a service or a set of services. The prioritization schema can be established automatically by the MNO or can be agreed with customers through contractual terms, and it may depend on several factors such as: (i) environment of applicability of services (e.g., allocation of a higher level for emergency services, services associated with search and rescue operations in disaster, or services for public safety); (ii) characteristics of the services (e.g., the number of end-users, location, average consumption, etc.); or (iii) specific requirements from customers.

- *Use of time shifting capability for service execution:* To efficiently use the energy resource and adapt consumption to generation, different strategies can be applied optionally to services belonging to a network slice. The application of the strategies on services can seek several objectives such as: (i) maximization in energy utilization, as shown in Fig. 6.14a; (ii) maximization in service processing (acceptance rate); (iii) multi-metric objectives, e.g., maximize consumption and also service processing; or (iv) it may be linked to a scheme that prioritizes the processing of one type of service over others. These objectives, as well as priority in network slices, can be established automatically by the MNO or can be defined in contractual agreements with consumers.



(A) Application of management strategies for the network slice 3. In this example, the initial distribution of services (initial time, duration, and power demanded) causes an inefficient use of energy and allows the execution of only two services (S_3 and S_5). The use of management strategies instead allows optimization in the consumption and processing of four services (S_1, S_2, S_3, and S_4).



(B) Disaggregated representation of the management strategies applied to services belonging to network slice 3.

FIGURE 6.14: Example of application of the management strategies for adaptive energy management in 5G network slicing.

In the context of energy efficiency, a strategy that adapts consumption to existing supply is the temporal displacement in the service execution time [14]. Thus, the *time shifting* is helpfu to encourage the anticipated consumption or a deferral in the use of energy for service execution within a finite time horizon and according to the energy availability, as shown in Fig. 6.14b for service S_1. In the proposed management model, the *time shifting* strategy, similar to the others described in this section, can be implemented because the service processing (and corresponding energy consumption) is not carried out immediately upon customer requests, but rather there exists a calculation process executed by the MNO (specifically at network slice management and orchestration domain), in which the configuration of network slices and services is performed by applying management strategies and considering energy generation and consumption conditions. This procedure enables the MNO to find the efficient distribution of services (service scheduling) that optimizes energy consumption, as shown in the example in Fig. 6.14a.

- *Degradation in service quality:* If the available power level is not sufficient to satisfy the power demanded by a certain service (or services), the MNO may choose to apply a *quality degradation* to that service, i.e., allocate a lower amount of energy than requested, as observed for the service S_4 in Fig. 6.14b, in which the assigned power level is half of the demanded. This strategy allows reducing consumption to a level tolerable by the service, which allows maintaining its functionality. Another possible action that the MNO can execute on the services consists in the implementation of a combined strategy, in which *time shifting* and *quality degradation* are applied to the service simultaneously, as shown in the example in Fig. 6.14b for service S_2. On the other hand, considering that generation and consumption fluctuate constantly, during periods of energy surplus (due to high generation or low demand), one strategy that could be adopted is the increase in quality (i.e., deliver more energy of the demanded). Thus, the implementation of this strategy on services that tolerate the increase in energy supply would prevent the energy produced from being wasted if it is not consumed.

- *Normal Processing of Services:* Although the normal processing of a service (i.e., execution without modifications or actions on it) is not a management strategy as such, it is an alternative configuration mode. If there is the availability of the supply, the analyzed service can be processed in its natural execution time and with its power level demanded, as shown in Fig. 6.14b for service S_3.

- *Service Rejection:* If the energy supply is not sufficient to guarantee the execution of a service, either due to an energy shortage or because the energy has already been allocated to another service, and also the application of the strategies described above (i.e., *time shifting* or *quality degradation*) does not allow its processing, the service in the admission stage is not processed, as shown in Fig. 6.14b for service S_5. The criteria for service rejection may be established by the MNO and in agreement with the customer.

  In the example in Fig. 6.14, each energy management strategy acts on a specific service. However, in a real implementation, the MNO with information of the services (required execution time, duration, and power demanded) and considering the use of the different management strategies described above, must be able to apply the best strategy for each service. This process is performed with the aim that the total or aggregate power of all services optimizes the use of the available supply. To this end, the MNO executes service scheduling algorithms, which allows achieving the energy efficiency required by 5G networks in a network slicing context.

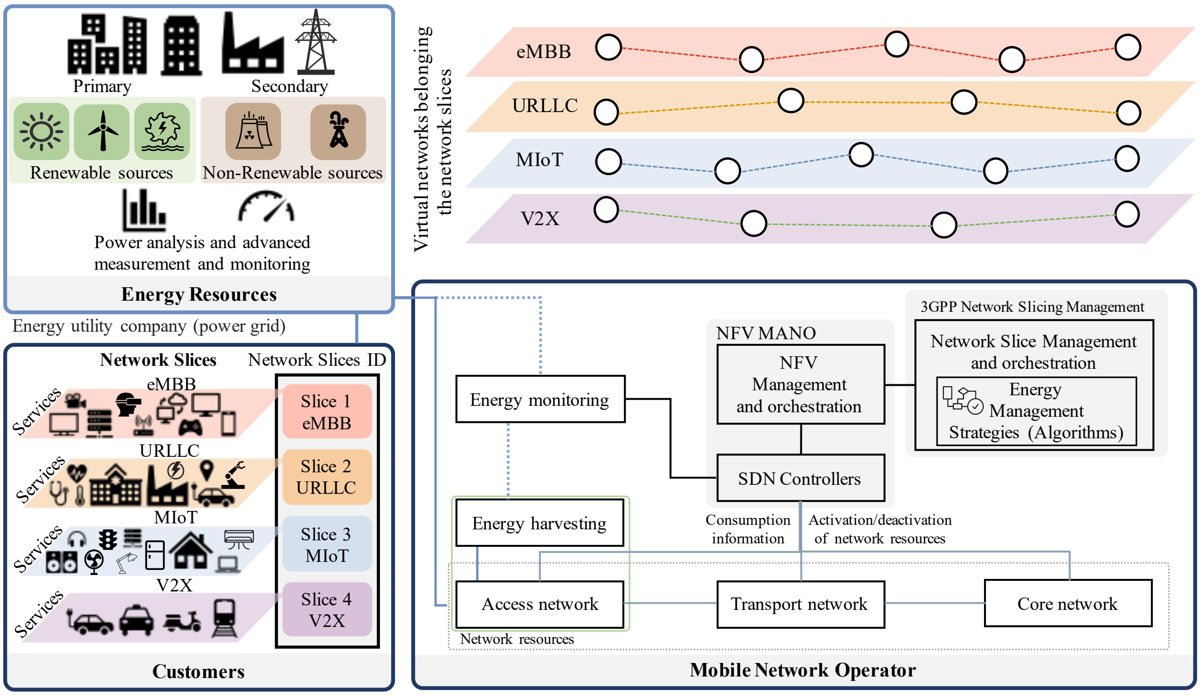
6.2.3.3 Architecture for Adaptive Energy Management

This section presents an architecture for adaptive energy management in 5G network slicing based on the established management and orchestration framework by the 3GPP and ETSI [146, 147, 150]. This architecture is composed of two stakeholders, the customers and the MNO, as shown in Fig. 6.15. The architectural framework is focused on the efficient energy resource management from renewable and non-renewable energy sources for the creation, configuration, and deployment of network slices and the corresponding services on the mobile network infrastructure.

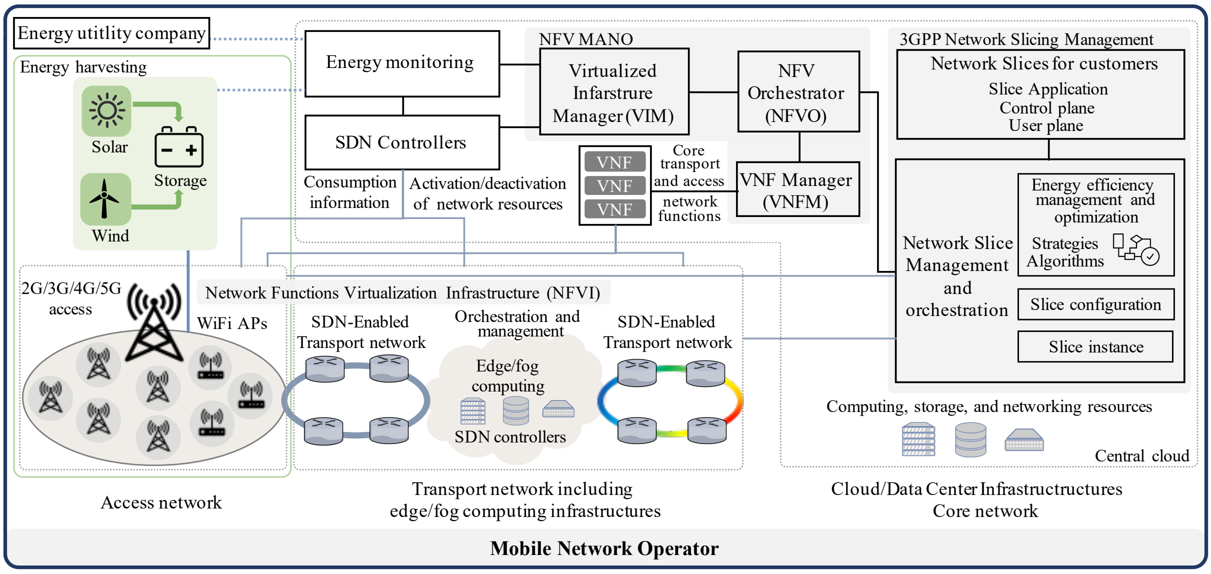
The architecture leverages the concepts of NFV and SDN as well as the IoT connectivity of customers to carry out the energy-aware realization of 5G network slices. From the operational perspective, the network slicing architecture can be considered to be comprised of two main blocks: (i) the first block is integrated with the NFV MANO framework and by 3GPP Network Slice Management framework, as shown in Fig. 6.15a, dedicated to the network slices management and configuration considering energy management requirements (optimization of power consumption); and (ii) the second block is composed by the NFV and SDN frameworks as well as by the underlying network resources dedicated to the network slice implementation.

In the first block (MANO), with the information on customers (e.g., number of network slices and services, the power consumption of services, SST values, etc.) and the conditions on power generation (available from non-renewable and renewable energy sources), the MNO proceeds to create the network slice(s), starting from a base template, similar to the example shown in [167], or selecting a network slice profile from a network service catalog [148]. This template can be then customized to meet the specific requirement of the end users. To carry out the network slice(s) creation (by choosing any of the above methods) on the MANO framework, the MNO first analyzes the amount power demanded (consumed) for service processing (execution), considering the consumption in all segments of the mobile network (i.e., the power consumed in the core, transport, and access networks). This consumption estimation procedure can be performed using historical data or predictive models. The consumption estimation process is carried out for all services and for all network slices. It has two objectives: (i) estimate/verify if the available power (managed by the MNO) is sufficient to allow the processing of all services in all network slices; and (ii) establish the actions to be performed by the MNO to optimize power consumption (optimize the service processing).

Traditionally, the increase in energy demand has been faced with the contracting or production of additional energy. However, this alternative involves an increase in OPEX and possible environmental impacts, and it is not a sustainable solution. In this regard, modern management systems demand adaptive consumption schemes restricted to availability and that prioritize the use of renewable energy, as described in Requirements R1, R4, and R5 in Section 6.2.3.1.



(A) Schematic of the proposed architecture.



(B) Schematic of the network operator and management entities.

FIGURE 6.15: Overview of the architecture for adaptive energy management in 5G Network Slicing.

Therefore, in the proposed architecture, the network slice process is aware of an available energy resource (i.e., services/slices and power supply considered jointly). The MNO if needed can apply various management mechanisms on services, which are deployed as algorithmic solutions on the Network Slice MANO (see Fig. 6.15b) to adapt consumption to availability, optimize power consumption (avoiding peaks loads or energy shortage), and reduce dependence on non-renewable energy as much as possible. As a result of the analysis of service parameters (customer information), the available supply, and the possible management mechanisms at the service level (network slices), the MNO obtains the optimal service scheduling (number of processed services, service execution time, and power consumption levels) in each network slice that leads the optimization of power consumption (minimization of power from renewable energy consumption) and the energy-efficient network slice creation and configuration. A summary of the energy-aware creation and configuration of network slices in the proposed architecture is shown in Fig. 6.16.

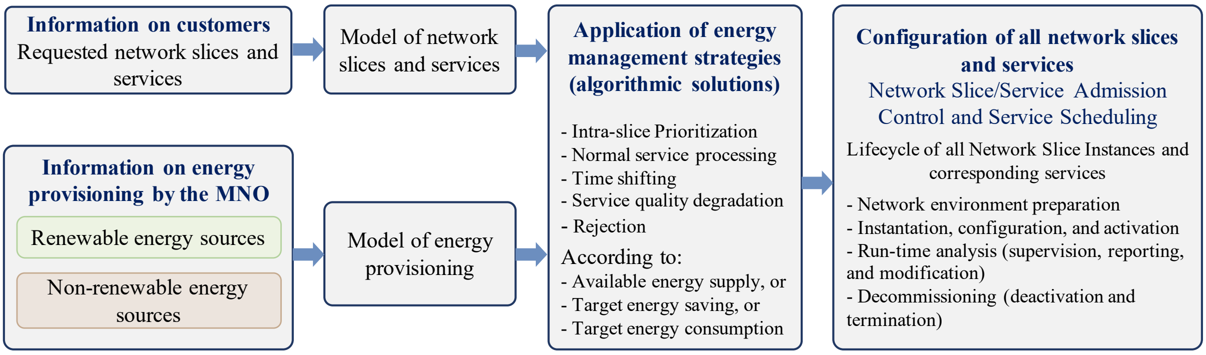


FIGURE 6.16: Description of the energy-aware network slice creation in the adaptive energy management architecture. The network slices and corresponding services are configured to achieve energy efficiency, considering a finite energy provisioning and the optimization of power consumption, specifically through the minimization of the consumption of non-renewable energy sources.

Once the general structure of the network slicing (network slice profile) has been selected, the MNO can deploy in the network slice MANO entity (at core network) the corresponding network slice instance, which refers to a set of network function instances (physical or VNFs, e.g., belonging access, transport, and core networks), the connectivity between them, and the required resources (e.g., compute, storage, and networking resources) that form a deployed network slice [148]. Typically, a network slice instance is designed (preparation phase), then it is instantiated (instantiation, configuration, and activation phase), later it is operated (run time phase), and finally it may be decommissioned (decommissioning phase) when the slice is no longer needed [142]. The complete NSI lifecycle is managed by the MNO, specifically by the network slice MANO. In the process of configuring and deploying network slices, the MNO sends the NSI information to the second block of the architecture, specifically to the NFVO, which is in charge of coordinating with the VNFM and with the VIM the creation of VNFs (that provide specific network capabilities to support and realize the particular service(s)) and virtual networks (access, transport, and core) need for the deployment of network slices. Then, the requests from the VIM entity are sent to the SDN controllers (access, transport, and core controllers), which coordinate with the NFVI the creation of network slices on the underlying network infrastructure (physical network resources). At this point, all created slices compose a single (end-to-end) network slice for specific customers. All of the network slices are managed and orchestrated first by an NFVO and in an upper level by the NFV and 3GPP MANO frameworks.

- *Customers (Network Slices) Modeling:* In the proposed architecture, the customers correspond to the network slices owners. They have connectivity capabilities and demand from the MNO all network resources (virtual, physical, and energy) to carry out the services, applications, or verticals. In the context of the adaptive energy management model, a service is characterized by its power consumption (i.e., by the amount of power that is used in the core, transport, and access network when the service is in execution), and it is tolerant to the possible actions (management mechanisms) that the MNO can execute on it to carry out the adaptive energy management. Considering the parameters related to power consumption and the management mechanisms described in Section 6.2.3.2, a service k, with $k \in \{1, \ldots, N\}$ of priority level j, with $j \in \{1, \ldots, M\}$ belonging to the network slice i, with $i \in \{1, \ldots, L\}$, denoted as $S^i_{k,j}$ is fully characterized by the parameters of Table 6.8. Figure 6.17 depicts an example of a service and its corresponding parameters.

TABLE 6.8: Parameters of network slices and services.

Parameter	Description	Unit/Comment
L	Number of network slices	Integer number
i	Network slice identifier	$i \in \{1, \ldots, L\}$
M	Number of priority levels of services	Integer number
j	Priority level identifier	$j \in \{1, \ldots, M\}$
N	Total number of services in the system	Integer number
k	Service identifier	$k \in \{1, \ldots, N\}$
Q	Number of quality degradation levels	Integer number
$t^i_{k,j}$	Starting time of service $S^i_{k,j}$	Time units
$d^i_{k,j}$	Duration of service $S^i_{k,j}$	Time units
$p^i_{k,j}$	Power demanded of service $S^i_{k,j}$	Power units
$q^i_{k,j}$	Quality level of service $S^i_{k,j}$	Discrete values (e.g., $[q^i_{k,j\,min} = 0.1, \ldots, q^i_{k,j\,max} = 1]$)
$u^i_{k,j}$	Time shifting value of service $S^i_{k,j}$	Time units (*backward*:$t^i_{k,j} - u^i_{k,j}$, or *forward*: $t^i_{k,j} + u^i_{k,j}$)
v^i_k	Priority level of service k belonging to network slice i	Integer number

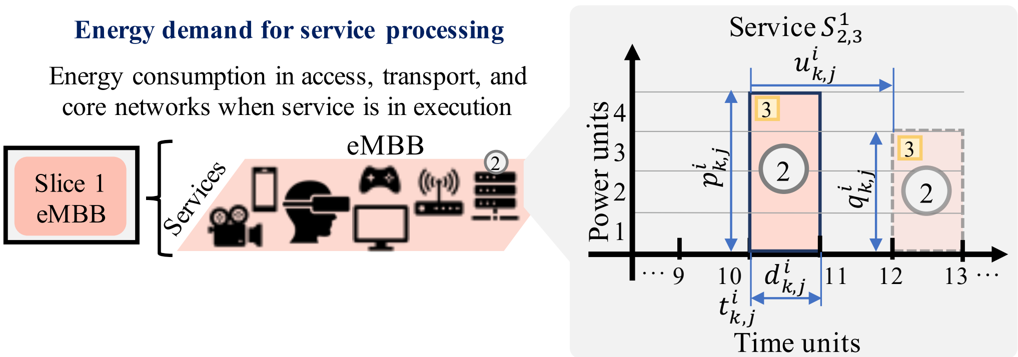


FIGURE 6.17: Graphical representation of the characterization of a service that is part of a network slice. Description: Service 2, with a priority level 3, and belonging to the network slice 1. Parameters: $t^1_{2,3} = 10$, $d^1_{2,3} = 1$, $p^1_{2,3} = 4$, $u^1_{2,3} = +2$ (*forward*), $v^1_2 = 3$, $q^1_{2,3} = 0.75$ ($p^1_{2,3} \times q^1_{2,3} = 3$).

In terms of power consumption, the total or aggregated power demanded by all network slices and corresponding services (P_D) can be expressed as:

$$\forall i \in L, \forall j \in M, \forall k \in N: \; P_D = \sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{N} p_{k,j}^{i} \times q_{k,j}^{i} \tag{6.29}$$

- *Mobile Network Operator (MNO) Modeling:* The MNO in the network slicing architecture is the entity that provides network resources (physical and virtual), is responsible for the creation, modification, and deletion of network slices, and is in charge of managing the energy resource for the operation of the entire mobile network. In this context, the total available power (P_A) in the mobile network ecosystem comes from the contribution of energy from renewable and non-renewable energy. The MNO is then able to control the contribution of one source over another, and particularly can promote the primary or majority use of green energy. The mathematical model of P_A in the proposed network management model is given by:

$$P_A = P_R + P_{NR} \tag{6.30}$$

where P_R represents the power obtained from renewable energy sources, while P_{NR} stands for the power from non-renewable energy sources. These parameters are given by:

$$P_R = P_A \times w_R \tag{6.31}$$

$$P_{NR} = P_A \times (1 - w_R) \tag{6.32}$$

where the factor $w_R \in [0, 1]$ denotes the weight related to the contribution of renewable energy in the total generated power P_A, which can be controlled by the MNO according to the green energy availability and the application scenario.

6.2.4 Energy Management Model Mathematical Formulation

This section presents the mathematical formulation of the proposed energy management solution. Section 6.2.4.1 describes the assumptions considered in the proposed model, while Section 6.2.4.2 presents the ILP formulation of the proposal.

6.2.4.1 Assumptions Related to the Energy Management Model

The proposed adaptive energy management model is summarized in Fig. 6.18. To provide a practical implementation of this proposal, the assumptions (simplifications) in Section 5.1.1 have been considered, including the following ones:

1. Use of discrete values for service quality degradation. To provide a feasible energy management model and maintain the linear condition of the problem, in the proposal, the values of degradation, also called levels of degradation that can be

applied to a service, are restricted to a finite set of possibilities. Analogously to the time shifting value, this sequence of values should be small (e.g., up to three or four degradation levels) because an increase in levels corresponds to a non-linear increase in the complexity associated with optimal energy management.

2. In 5G network slicing, a customer (user equipment) may be served by at most eight network slices at a time [142]. However, in our proposal for simplicity, a customer can only belong to one network slice. Moreover, the customers in each network slice are different from each other (i.e., different network slices specified to different customers).

3. If needed, the MNO can deploy multiple network slices of the same slice/service type (e.g., eMBB with the same features but for different groups of customers). In this case, the MNO is able to differentiate the slices according to the network slice identifier, as shown in the example in Fig. 6.14a.

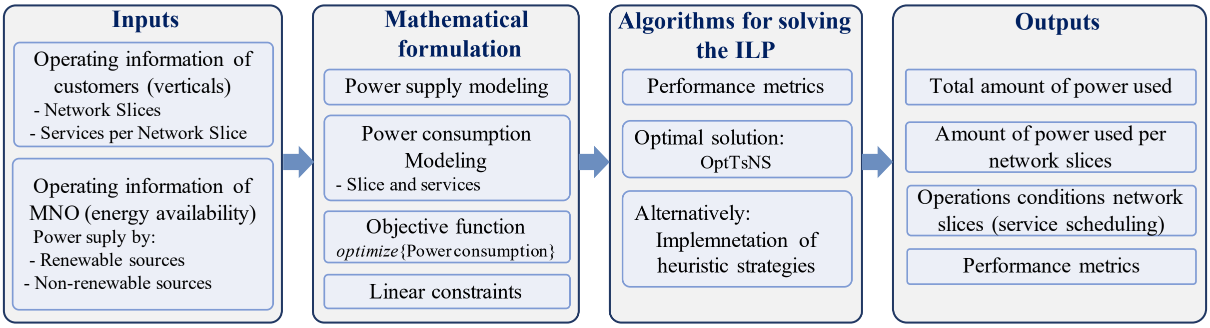


FIGURE 6.18: Schematic of the adaptive energy management model for 5G Network Slicing.

6.2.4.2 ILP Problem Formulation: OptTsNS

- *Objective function:* In our proposal, the creation and configuration of network slices and associated services is energy-aware. Thereby, the proposed management model seeks to improve energy efficiency through the optimal use of the available supply and specifically through the minimization of power consumption. Technically, this objective is expressed as indicated in Eq. 6.33. Nevertheless, considering that an important requirement in an energy management system for mobile networks is the promotion and use of green energy (as mentioned in Requirement R4 in Section 6.2.3.1 and as shown in the energy provision model in Eq. 6.30), the objective of the proposed energy model should be focused on minimizing the consumption of energy from non-renewable sources. To this end, we have established a cost function associated with consumption in the 5G ecosystem from renewable and non-renewable sources as shown in Eq. 6.34, where w_1 and w_2 can be in the range [0,1] and represent the weights associated with the individual cost functions that are proportional to renewable (c_{P_R}) and non-renewable ($c_{P_{NR}}$) energy consumption, respectively.

To promote the primary utilization of renewable energy, the cost associated with the consumption of this kind of energy in Eq. 6.34 can be set to a minimal value ($w_1 \ll w_2$) and even to zero. In this model, we chose the second option (i.e., $w_1 = 0$) with the aim of moving toward a green mobile network ecosystem. Therefore, the objective function of minimizing power consumption in Eq. 6.33 considering the cost function in Eq. 6.34 translates into the minimization of the cost associated with non-renewable energy consumption as indicated in Eq. 6.35, which in turn is equivalent to the objective of minimizing the P_{NR} in the total P_A as shown in Eq. 6.36. This objective function must respect the available energy resources, consider the parameters of the services, the management mechanisms described in Section 6.2.3.1, and the constraints presented below.

$$minimize\ \{P_D\} \tag{6.33}$$

$$Cost_{PD} = w_1 \times Cost_{P_R} + w_2 \times Cost_{P_{NR}} \tag{6.34}$$

$$minimize\ \{Cost_{P_{NR}}\} \tag{6.35}$$

$$minimize\ \{P_{NR}\} \tag{6.36}$$

- *Constraints:* The following constraints are linked to the proposed adaptive energy management model.

$$C1 : P_A[t] \geq 0 \tag{6.37}$$

$$C2 : (P_A[t] - P_D[t]) \geq 0 \tag{6.38}$$

$$C3 : \sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{k=1}^{N} \left(p_i^{k,j} \times q_i^{k,j}\right) \times x_{ijk}[t] \leq P_A[t],\ x_{ijk} \in \{0,1\} \tag{6.39}$$

$$C4 : t_{k,j}^i \geq 0 \tag{6.40}$$

$$C5 : \{t_{k,j}^i - u_i^{j,k}\} \geq 0 \tag{6.41}$$

$$C6 : W \geq \max\{t_{k,j}^i + d_{k,j}^i + u_i^{k,j}\} \tag{6.42}$$

  - *Domain constraints:* The energy supply by the MNO for service processing is assured by $C1$. Instead, $C2$ guarantees a non-negative difference between the power demanded and power provisioning. In Constraints $C1$, $C2$, and $C3$, the power variables are specified at time slot t, because the energy provisioning and consumption may vary at each time slot. Thus, the objective of the proposed model is the minimization of power consumption (from non-renewable sources) during all time slots within a finite time horizon W.
  - *Capacity Constraint:* In the mobile communication system, the maximum energy capacity is limited by $C3$, in which the decision variable x_{ijk} stands for the allocation of energy resources for the processing of the service $S_{k,j}^i$, as shown in Eq. 6.43.

$$x_{ijk}[t] = \begin{cases} 1 & \text{if the service } k \text{ belonging to the network slice } j \text{ with} \\ & \text{priority level } j\ (S_{k,j}^i) \text{ is processed at time slot } t, \\ 0 & \text{otherwise.} \end{cases} \tag{6.43}$$

The correspondence between the service, its priority level, and the network slice to which it belongs is validated by Eq. 6.44.

$$p^i_{k,j} = \begin{cases} \text{Power demanded by service } k \text{ with priority } j \text{ and within the network slice } i & \text{if service } k \text{ with priority level } j \text{ belonging to a network slice i exists,} \\ 0 & \text{otherwise.} \end{cases} \tag{6.44}$$

- *Time constraints:* $C4$ and $C5$ ensure a non-negative starting time for the entire system ($t = 0$). $C6$ guarantees a finite time horizon for the analysis of services. The linear condition of the objective function, the constraints, and the decision variable deliver the linear nature to the problem.

- *Adaptive Management of Network Slices and Services:* Taking into account that the energy provisioning (mainly from renewable sources) in the mobile network is finite, the consumption demanded for service execution must be adapted to availability as defined in Requirement R5 in Section 6.2.3.1. Depending on the amount of energy available and demanded (e.g., $P_D > P_A$) and the features of customers, the MNO has the possibility of executing different management mechanisms on the services such as intra-slice priority scheme, service rejection, degradation in service quality, and use of time shifting in service execution, as indicated in Section 6.2.3.2. The execution of these strategies allows the MNO to establish an access control to the energy resources associated with the creation of the network slices and configuration of corresponding services.

  To find the optimal allocation of energy resources for service execution, the MNO must analyze the action of the different strategies on the services, considering the particularities of each service and/or network slice. This procedure is carried out through algorithms implemented in the core network, as shown in Fig. 6.15b, and can be computationally very demanding. At the end of this analysis, the MNO obtains the optimal scheduling for the set of N services. Through optimal scheduling of the set of N services (set defined in the proposal as *combination of services*, *comb*) the MNO tries to process as many services as possible (in the worst case the services with lower priority can be rejected if $P_D > P_A$) and with the minimum impact on the requirements from customers (respecting as much as possible the original execution time and power level demanded by services). In our proposed model, the adaptive energy consumption is represented by a cost function, as shown Eq. 6.45. In this equation, α, β, γ, and δ can be in the range [0,1] and correspond to the weights of the individual cost functions related to the management mechanisms, which are listed as follows:

$$Cost^{i,j,k}_{comb} = \alpha \times Cost_{AR_{comb}} + \beta \times Cost_{pri_{comb}} + \gamma \times Cost_{u_{comb}} + \delta \times Cost_{q_{comb}} \tag{6.45}$$

  - $Cost_{AR_{comb}}$: Cost function related to the processing of services and defined by Eq. 6.46.
  - $Cost_{pri_{comb}}$: Cost function related to the priority level of services and defined by Eq. 6.47.

- $Cost_{u_{comb}}$: Cost function related to the time shifting application in service execution and defined by Eq. 6.48.
- $Cost_{q_{comb}}$: Cost function related to the quality degradation of services and defined by Eq. 6.49.

$$Cost_{AR_{comb}} = \begin{cases} 0 & \text{if all services are processed,} \\ \text{Total rejected services} & \text{otherwise.} \end{cases} \tag{6.46}$$

$$Cost_{pri_{comb}} = \begin{cases} 0 & \text{if all services have the maximum priority level,} \\ \sum_{i=1}^{L} \sum_{k=1}^{N} v_k^i & \text{otherwise.} \end{cases} \tag{6.47}$$

$$Cost_{u_{comb}} = \begin{cases} 0 & \text{if all services are processed without time shifting,} \\ \sum_{i=1}^{L} \sum_{j=2}^{M} \sum_{k=1}^{N} u_{k,j}^i & \text{otherwise.} \end{cases} \tag{6.48}$$

$$Cost_{q_{comb}} = \begin{cases} 0 & \text{if all services have no quality degradation,} \\ \sum_{i=1}^{L} \sum_{j=2}^{M} \sum_{k=1}^{N} q_{k,j}^i & \text{otherwise.} \end{cases} \tag{6.49}$$

In Eq. 6.45, the weights of individual cost functions can be set based on specific requirements from the MNO and/or from the customer. However, a possible configuration can be $\alpha > \beta > \gamma > \delta$, because this relationship promotes the processing of a higher number of services and the minimum impact on services due to the application of time shifting in service execution or degradation in service quality. In addition, the optimal scheduling of N services (i.e., the optimal combination of services) that leads to optimal energy consumption is given by the minimum value of cost function (minimum value in Eq. 6.45). Thus, the objective of the proposed energy management solution is to minimize the total cost function, as shown in Eq. 6.50. For the implementation of the adaptive energy management model represented by its cost function in Eq. 6.50 there are a number of different possible algorithmic solutions (optimal and sub-optimal). In this section, we present an exact or optimal algorithmic strategy, which is described in detail in Section 6.2.5.

$$\forall i \in L, \forall j \in M, \forall k \in N : minimize\{Cost_{comb}^{j,j,k}\} \tag{6.50}$$

The energy-aware resource allocation for service processing in the context of 5G network slicing, considering the finite energy supply, an intra-slice priority scheme, and different management strategies with the aim of minimizing energy consumption through optimal energy utilization is equivalent to the objective of multi-dimensional multi-choice knapsack problem of choosing the most valuable items of a set of classes (one item per class) without overloading the knapsack [102]. The literature has proven that the complexity linked to this kind of problem is NP-hard. Establishing an analogy with our proposal, the multidimensional behavior is given by the power and time parameters of services, and the multiple-choice feature corresponds to the selection of a specific time shifting value and/or quality degradation level from a possible set of options. Thus, we can conclude that the optimal adaptive energy management in 5G network slicing falls into the NP-hard classification.

6.2.5 Evaluation

To validate the operation of the proposed adaptive energy management model, in this section, we present an optimal service scheduling algorithmic strategy denoted as OptTsNS. The objective of OptTsNS is to minimize power consumption from non-renewable sources considering finite energy provisioning and the management mechanisms described in Section 6.2.3.2. Concerning the optimization of power consumption, the task of OptTsNS is to find the best actions (strategies) for each service in such a way that the processed services (i.e., the combination of services that demands P_D) enable the efficient use of the available energy. The proposed algorithmic strategy, described in Section 6.2.5.1, bases its operation on an exhaustive search method. In this brute-force method, all possible combinations of N services, executed simultaneously, are explored considering all possible values of degradation and time shifting in service execution. To choose the best combination of services (i.e., the optimal distribution of services in time, with a P_D (P_{Dcomb}) that produces minimization of P_{NR} and, consequently, the optimal use of P_R), the cost functions are used. Specifically, the optimal solution delivered by OptTsNS corresponds to the combination of services that produces the minimum cost function, as shown in Eq. 6.50. To quantitatively verify the improvements obtained with OptTsNS, several performance metrics are used and a numerical analysis is performed on a particular case study. In this context, an analysis in different scenarios, as well as the online version of OptTsNS and the development of more sophisticated and efficient methods (that use as the optimal solutions as upper bounds), could be addressed in future work.

The performance metrics that allow meeting Requirement R7 are the acceptance ratio AR, available energy utilization (E_{A_U}), and missing power (P_{LACK}), which have been previously defined in Section 5.1.2.4 and in Eq. 6.21.

6.2.5.1 Optimal Algorithmic Strategy: OptTsNS

Figure 6.19 explains the proposed algorithmic strategy OptTsNS and the main steps carried out are summarized below.

- *Variations per service* ($VarServ$)*:* A variation of a service is the result of the application of a specific discrete time shifting value to the $t^i_{k,j}$ and/or the application of a specific quality degradation level to the $p^i_{k,j}$ of a service $S^i_{k,j}$. The analysis of all N services for each value of time shifting and for each quality degradation level produces a total number of variations $AllVarServ$. Considering that, for simplicity, the time shifting forward and backward have the same value, this number is given by:
$$AllVarServ = Q \times (2 \times N \times max\{u^i_{k,j}\} + N) \tag{6.51}$$
- *Combinations of services* ($CombServ$) *and computation of cost functions:* In the algorithmic strategy, the set of N different variations of services ($VarServ$) is named as a *combination of services* ($Combserv$). Each $Combserv$ has specific characteristics and requires a certain power level P_{Dcomb}. The algorithmic strategy evaluates

the cost functions for each $CombServ$. Regarding the AR metric, the algorithm prioritizes the execution of the highest priority services (with $j = 1$ the highest priority level). If the services have the same priority level, the algorithm selects the set of services that produce an optimization in the consumption of available power (maximization of energy use). The combinatorial analysis of all ($VarServ$) delivers a total number of combinations of services $AllCombServ$, which is given by:

$$AllCombServ = Q \times (2 \times max\{u^i_{k,j}\} + 1)^N \tag{6.52}$$

The computation of combinations of services (variations) contributes largely to the growth of complexity of the problem. For instance, $N = 10$, $max\{u^i_{k,j}\} = 4$, and $Q = 3$ produce over 10 billion combinations. If a computer is able to process one $Combserv$ each millisecond, it would need over 2900 hours to explore the complete search space.

- *Sorting of combinations and selection of the best combination:* In this step, a quick-sort method is applied to all combinations, according to the descending value of $Cost^{i,j,k}_{comb}$. Then, the best combination (which is the first in the sorted list) is chosen. Finally, the energy is allocated to the services that can be processed, and the performance metrics are computed.

- *Complexity Analysis of* OptTsNS*:* The complexity of the exact solution is conditioned to the processing of $AllVarServ$ and $AllCombServ$. As a function of N, the growth rate can be expressed as:

$$f(N) = N + Q \times (2 \times N \times max\{u^i_{k,j}\} + N) + Q \times (2 \times max\{u^i_{k,j}\} + 1)^N \tag{6.53}$$

where the third term in Eq. 6.53 is dominant and represents the size of the search space that must be explored to find the optimal combination of services ($VarServ$) that lead to the minimization of power consumption (optimal power consumption). Therefore, the complexity the algorithmic strategy OptTsNS is exponential with an order of growth $\mathcal{O}(2^N)$ that depends on the selected values of N, Ts, $max\{u^i_{k,j}\}$, and Q.

6.2.5.2 Numerical Results

- *Simulation Setting:* The simulation conditions for OptTsNS is detailed in Section 5.4.0.1. The evaluation of the optimal solution is given in terms of metrics AR, E_{A_U}, and P_{LACK}. The results obtained are compared with a traditional scenario in which no management strategy is applied (i.e., when $v^i_k = 1$, $q^i_{k,j} = 1$, $u^i_{k,j} = 0$, $\forall i \in L$, $\forall j \in M$, $\forall k \in N$). The total execution time for the simulation was approximately 90 min.

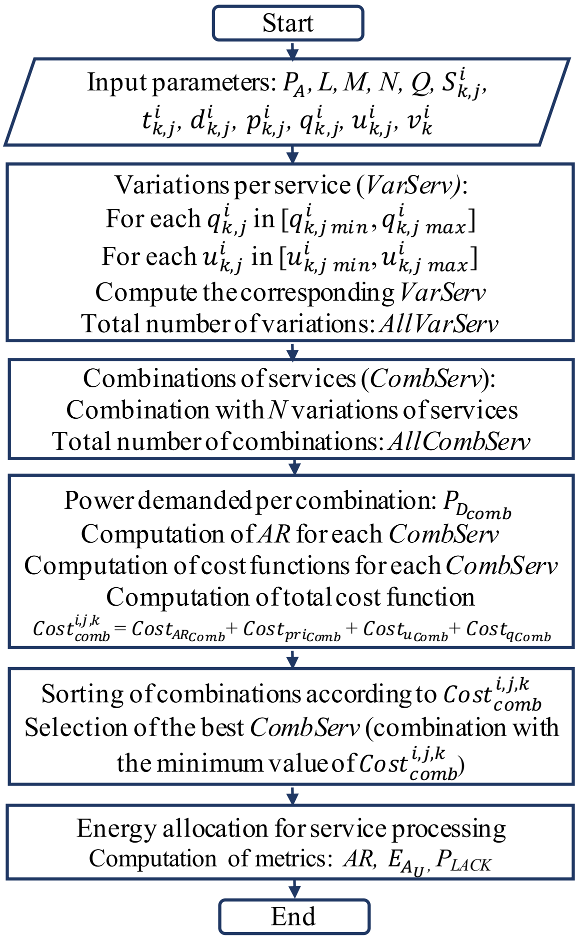


FIGURE 6.19: Flow chart of the exact algorithm strategy OPTTSNS.

- *Case Study:* Figure 6.20 shows the scenario that has been considered for the quantitative evaluation of OPTTSNS. This scenario allows the analysis of the different management strategies during states of shortage (high load) and surplus of energy produced by the lack of synchronization between energy generation and consumption. In this particular case, we also analyze the minimization of power consumption by considering a 100% green energy supply (i.e., promoting the use of energy from renewable sources). Due to the high computational requirements for the execution of OPTTSNS, the simulation was limited to $N = 8$ services, $max\{u^i_{k,j}\} = 4$ time slots (backward and forward), and $Q = 3$ quality degradation levels; the rest of parameter used are detailed in Table 6.9 and Fig. 6.20. To simplify the analysis, services with equal duration and a flat energy profile and were selected. However, this does not mean a limitation for the developed algorithm, which can work with any profile of energy generation and consumption if needed.

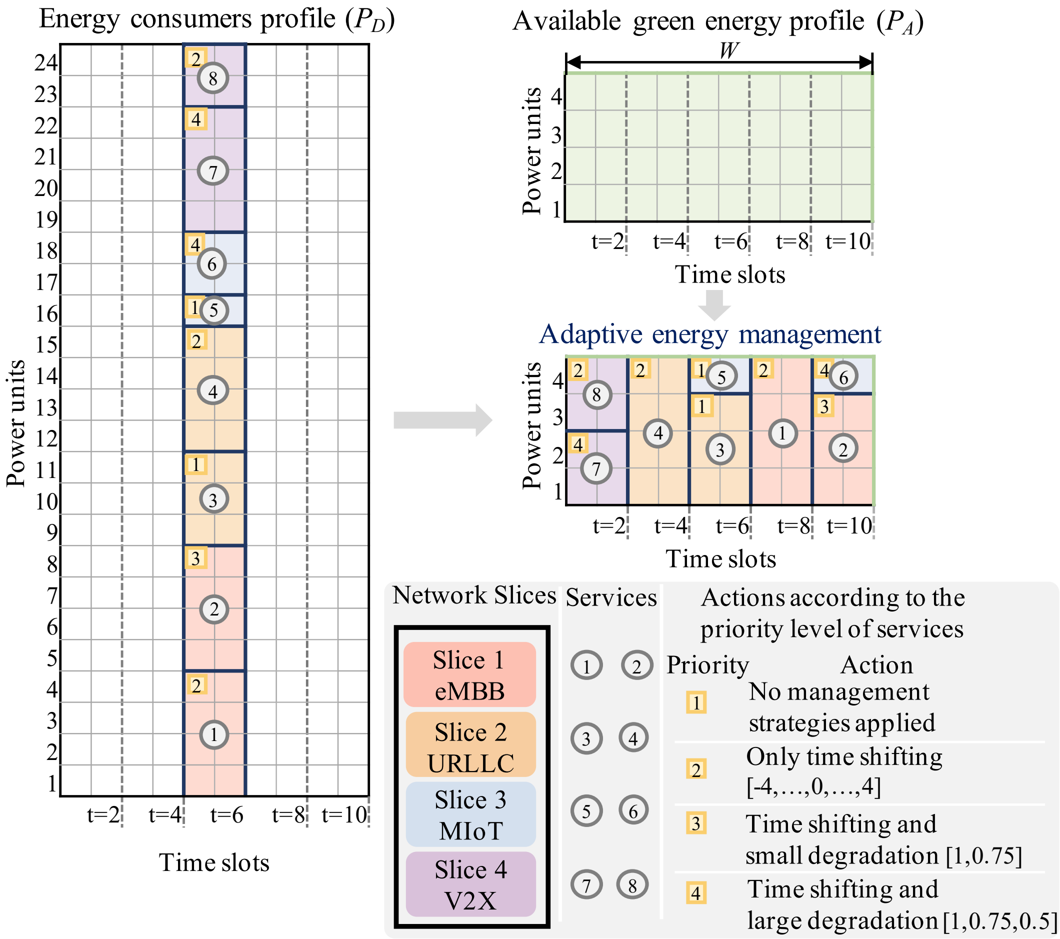
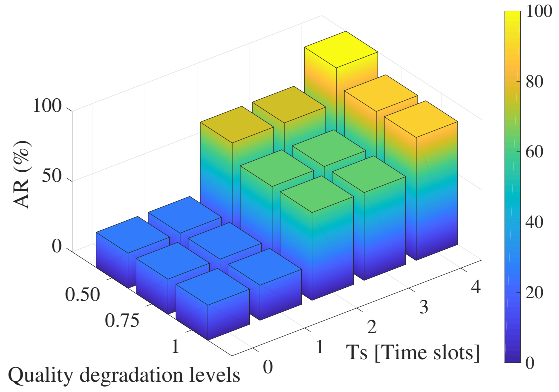


FIGURE 6.20: Energy provisioning and consumption profiles of the case study, and optimal energy allocation obtained with OPTTSNS.
Parameters: According to Table 6.9.

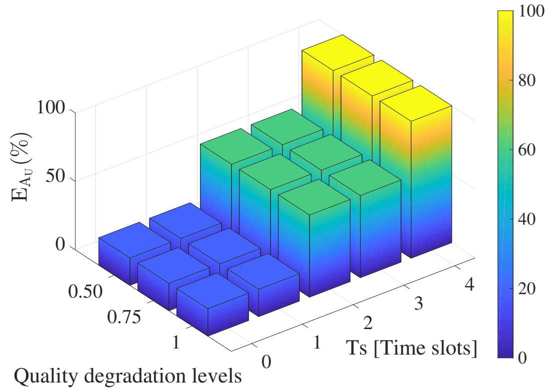
TABLE 6.9: Parameters of OPTTSNS and values regarding the case study.

Parameter	L	M	N	W	$t^i_{k,j}$	$d^i_{k,j}$	$max\{u^i_{k,j}\}$	$p^i_{k,j}$	Q	$q^i_{k,j}$
Value	4	4	8	10	4 $\forall k$	2 $\forall k$	4, see Fig. 6.20	[1-4], see Fig. 6.20	3	[1, 0.75, 0.5], see Fig. 6.20

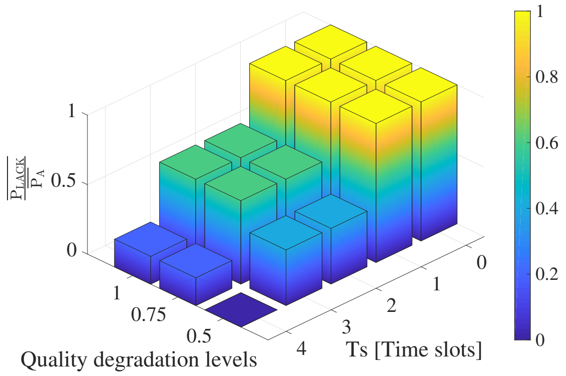
- *Results and Discussion:* The simulation results in Fig. 6.21 confirm the effectiveness of OPTTSNS to improve power consumption, as reported in values of metrics AR, E_{A_U}, and P_{LACK} in Fig. 6.21a–c, respectively. As the time shifting value increases (from 0 to 4) and as the quality degradation level decrease (from 1 to 0.5), OPTTSNS has the ability to distribute the services in time and minimize the power consumption, which consequently leads to efficient use all P_A. In this regard, the algorithmic strategy enables reducing the peak loads (i.e., the peaks of power consumption) and obtain a flat consumption profile, which can be better adapted to energy generation. Thus, the use of management strategies enables the processing of services that, in normal conditions (i.e., without a management mechanism) would be inevitably rejected. For instance, in Fig. 6.20, only services $S^2_{3,1}$ and $S^3_{5,1}$ could be processed, which corresponds to an AR = 25%, in this case the most of the energy produced could not be used and would be wasted. In addition, the simulation of OptTsNS shows that, when this algorithmic solution is deployed at the mobile network MANO entities, as shown in Fig. 6.15, it can offer efficient and adaptive energy management for 5G systems, considering the context of network slicing and with the ability to manage and exploit renewable energy generation.



(A) AR metric OptTsNS.



(B) E_{A_U} metric OptTsNS.



(C) P_{LACK} metric OptTsNS.

Figure 6.21: Performance evaluation of OptTsNS for case study of Fig **??**. Parameters: According Table 6.9.

For the analyzed use case, the improvements obtained with the algorithmic strategy are given in terms of energy utilization (as shown in Fig. 6.21b), increased service processing (from $AR = 25\%$ until reaching an $AR = 100\%$, as shown in Fig. 6.21a), and in reducing the use of energy from non-renewable sources to meet the complete energy demand (Fig. 6.21c). The latter is of utmost importance in modern energy management systems that aim to achieve a reduction in OPEX and with minimal environmental impact.

6.2.6 Conclusions

Network slicing is specified for 5G networks to provide new opportunities for service provisioning in order to increase efficiencies and improve revenue. Regarding energy consumption management, this feature is a key factor in the deployment and evolution of mobile networks, and it is also a crucial consideration. This section analyzes the requirements, proposes an architecture, and presents a feasible and efficient solution for adaptive energy management in 5G network slicing that meets the requirements of energy efficiency and sustainability by current and future mobile communications networks. The proposal aims to optimize the consumption of available power, and specifically seeks for the minimization of power consumption of non-renewable energy sources. In this regard, our proposed energy management model prioritizes the use of renewable energy sources, but non-renewable energy sources can be used when extra energy is required.

The architecture proposal covers the requirements for efficient, adaptive, and sustainable energy management; the mathematical model of the stakeholders involved in energy management and consumption in the 5G ecosystem; and several management mechanisms at service level such as an intra-slice prioritization scheme, service rejection, time-shifting in service execution, and degradation in service quality (i.e., decrease in energy demand) with the aim of optimizing available power consumption. The proposal also presents the ILP formulation of the adaptive energy management model and its complexity analysis, which has been proven to be NP-hard.

To validate the operation of the proposed adaptive energy management model an exact optimal algorithmic strategy, OptTsNS, is presented, of exponential complexity that depends on the values of N, $u^i_{k,j}$, and Q, as shown in Eq. 6.53, and which reveals the need for heuristic approaches for scalable, faster, and less computationally demanding implementations. The evaluation of the exact solution for a particular case study allows us to verify the improvements obtained with the proposal in power consumption and utilization, in the increase of service processing, and in the minimization of the use of non-renewable energy sources. This latter feature is of paramount importance because it can potentially reduce the OPEX for the MNO and the energy footprint because of the operation of the mobile networks.

6.3 Adaptive Energy Management in UAV-Enabled Communications

5G is expected to provide diverse and stringent improvements such as greater connectivity, bandwidth, throughput, availability, improved coverage, and lower latency. Considering this, drones or Unmanned Aerial Vehicles (UAVs) and IoT devices are perfect examples of existing technology that can take advantage of the capabilities provided by 5G technology. In particular, UAVs are expected to be an important component of 5G networks implementations and support different communication requirements and applications. UAVs working together with 5G can potentially facilitate the deployment of standalone or complementary communications infrastructures. Due to its rapid deployment, UAV-based solutions are suitable candidates to provide network services in emergency scenarios, natural disasters, and search and rescue missions. An important consideration in deploying a programmable drone fleet is to guarantee the reliability and performance of the services through consistent monitoring, control, and management scheme. In this regard, the NFV paradigm, a key technology within the 5G ecosystem, can perform automation, management, and orchestration tasks. In addition, to ensure the coordination and reliability in the communications systems, and considering that the UAVs have a finite lifetime and eventually they must be replaced, a scheduling scheme is needed to guarantee the availability of services and efficient resource utilization. To this end, this section presents a UAV scheduling scheme that leverages the potential offered by NFV. Based on a brute-force search combinatorial algorithm, the proposed strategy allows obtaining the optimal scheduling of UAVs in time to deploy network services efficiently. Simulation results validate the performance of the proposed strategy by providing the number of drones needed to meet certain levels of service availability. Furthermore, the strategy allows knowing the sequence of replacement of UAVs to ensure optimal resource utilization.

6.3.1 Introduction to UAV-Enabled Communications

Recent evolution in UAV boosted by the miniaturization of electronic and sensors have allowed the use of UAVs in different civilian applications. Their shrinking size in combination with price reductions has increased the popularity of these devices both in the amateur community as well as in professional applications. Accordingly, we are now witnessing the fast deployment of a new categorization in the UAV area: Small Unmanned Aerial Vehicles (SUAV), commonly known as drones (that will be the preferred name in this article), which are low-cost devices with reduced payload capacities, restricted communication range and limited battery time, but still powerful enough so as to carry small computers on-board.

Drone applications are spreading throughout a plethora of different fields covering from smart agriculture scenarios to road traffic monitoring, public safety, sensor information retrieving or even unmanned cargo. In general, these use cases are normally scheduled as relatively fixed missions of standalone drones [168]. This section, in particular, is focused on the energy management challenge that although is obviously present in standalone drone short missions, its complexity is exponentially exacerbated when dealing with multi-drone long-term operations.

The use of multi-drone network is apparently a cost-effective solution which enables a fast and agile deployment in hard-to-reach locations and can straightforwardly be integrated into existing networks and adapt to unexpected changes. This flexibility can be certainly improved with the usage of (NFV) 5G technology enabled drones as we have shown in [169]. However, drones have also several challenges that should be addressed. The weight a drone can carry determines the payload equipment and the size of the on-board battery. In consequence, we deal with low-resource payload (e.g. Single Board Computers) equipment and small batteries that provide limited service time. Because of these limitations, we need multi-drone systems to cover areas of significant size and a fleet of reserve drones for replacements in order to provide long-term services. Apart from connectivity requirements, such as latency and bandwidth, a communication system provided by drones needs innovative management solutions (e.g., NFV) that enable the use of resources and energy efficiently. For this reason, this section presents a strategy for the efficient management of resources in a communications system that provides services or network functions through the deployment of drones. The proposed solution leverages the potential offered by NFV and the 5G capabilities. In the context of the proposal, 5G technology is used to meet connectivity requirements, such as very low latency and high bandwidth in order to guarantee a correct migration procedure, and also to provide communication between the different components in the system. Instead, NFV is in charge of the management tasks in the system. Specifically, in order to carry out the management tasks related to the replacement of drones and allocation of drones to services, a energy-aware scheduling algorithm has been developed.

The proposed algorithm, based on a brute-force search combinatorial method, explores all possible combination of drones and service with the aim of providing the exact or optimal scheduling of drones to execute services. This exact allocation of drones over time ensures the continuity of services during a finite time interval, while leads to the optimal resource utilization. Apart from the replacement sequence the algorithm can inform the total number of drones (or batteries) to use to reach a certain level of availability. In addition, within an NFV scope, the drone scheduling strategy can be considered as a network service. To validate the performance of our solution, two small-scale scenarios have been analyzed, one generic and one realistic, whose results can be applied in the planning a design stages in a variety of real use cases, such as: services in emergency or natural disasters and relief services in search and rescue missions. In addition, the information obtained with our implementation is also useful as a baseline to develop mathematical models and faster sub-optimal or heuristics methods for real-time practical implementations.

6.3.2 Related Work

In the last years, drone uses have evolved from the basic on-board video camera applications to a wide range of novelty functions such as drones acting as first responders in an accident or drone swarming intelligence to provide network services. To conduct these assignments efficiently, on account of drones limitations, the use of 5G technologies such as NFV or SDN seems essential as they will enable an accurate operation. In particular, in this proposal for energy management, NFV is used to deploy the execution of the proposed algorithm.. There are several examples of the use of NFV in the UAV domain in the literature. In [170] an UAV platform provides to external controllers the opportunity to adapt the telemetry monitoring. In [171] is presented an NFV programmable infrastructure that enables the agile unification of services and functions, which may be determined by the operator of the UAVs at deployment time. NFV is used to decouple the drone hardware infrastructure from the control layer that virtualizes the infrastructure resources for the higher layers [172]. NFV is also used to enable multi-mission drones and supports a flexible deployment of network services[169]. Finally, NFV allows the migration of VNFs [170] which are the responsible units for providing the network functionality through the software implementation. The VNF migration enables an agile and flexible execution of the network services encompassing those VNFs that can be accommodated by the drones. Basically, the migration of VNFs consists on moving a virtual machine from one drone unit to another. There are different migration types: Non-live migration, where the VNF is down and it is moved to a different compute node, and Live-migration, where the VNF is running throughout the migration. Well-known tools that are key in the NFV framework development, such as OpenStack[1] or VMware[2], support migration. The use of NFV is reinforced by the appearance of multi-drone systems. Drones can run different VNFs, endowing a huge versatility to the drone swarm. Nonetheless, virtualization is a resource intensive process, and because of the limited on-board equipment, it is necessary to use solutions such as LXC Linux Containers[3] to provide a similar environment as a VM but reducing the overhead that comes with running a separate kernel and simulating all the hardware. It should be noted that the migration of container-based VNFs presents an additional challenge since this virtualization unit is stateless and in principle, cannot be migrated. However, there are recent works in which they aim at addressing migration issues with containers like CRIU (A project to implement checkpoint/restore functionality for Linux: `https://github.com/checkpoint-restore/criu`). For this reason, in this proposal, because of the selected VNFs, the migration process is not mandatory. Routing VNFs recover their status proactively, collecting all the necessary routes in a few seconds but, in complex network scenarios, the use of VNF migration is crucial for correct operation.

Different alternatives for drone communication has been proposed in [173] being the WiFi in Ad-Hoc mode one of the most popular solutions. Regarding routing strategies, there are also several options that extend from Mobile ad-hoc networks (MANETs) and vehicular ad-hoc networks and also some innovative proposal like SDN.

The main focus of the proposal presented in this section is set on battery power consumption. All the drones are normally equipped with a Single Board Computer as payload (Raspberry Pi 3B[4] (RPi) in our case), with an autonomous battery giving rise an independent service of the drone. In standard drone applications (drone flying as expected), flight-engines consume most part of the energy [174] while the power consumption by

[1] OpenStack: https://docs.openstack.org/ocata
[2] VMware: https://www.vmware.com/es.html
[3] Linux Containers: https://linuxcontainers.org/
[4] Raspberry Pi 3B: `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`

network services is practically negligible, so that, the service time is limited by the drone battery time (around 20 minutes following the technical specifications[5]). Even so, when a drone has a static position, it tends to land whenever possible to save energy (a drone that is providing a WiFi access point service does not necessarily has to be flying). In this case, service time will be limited by the SBC battery time and network services should be taken into account and will play an important role in modeling battery consumption.

Regarding power consumption in mobile and portable devices, there are different examples studying the impact of hardware components on the energy consumption [175][176] and also the impact related with Wireless Communication [177]. In [178] is presented a method for wirelessly charging the drone battery when it lands, without the need to remove it and replace it. Ground task automation has come to the attention of researchers during the past few years [179][180] reducing the human operators at the Ground Control Station (GCS).

In addition, in order to efficiently manage the available resources (e.g., energy), various techniques, mechanisms and procedures have been developed. One of the most widely used is the combinatorial analysis, in which all possible combinations of resources to be used are analyzed. In this proposal, this mechanism is used to analyze all possible combinations of drones to run services. Considering a procedure similar to that described in [14], the proposed technique, by analyzing the whole set of possible cases, ensures the best (exact) result by providing the information of specific resources (drones and batteries) to be used. This optimal scheduling of drones guarantees an efficient use and management of available energy at every moment.

6.3.3 Problem Statement and System Model

In this section, first the statement of the problem is formally presented in Section 6.3.3.1. Then, the system model and notations are described in Section 6.3.3.2 followed by the definitions in Section 6.3.3.3 and the performance metrics in Section 6.3.3.4. Finally, the assumptions are presented in Section 6.3.3.5.

6.3.3.1 Problem Statement

Maintaining a certain degree or level of availability can become an important and even critical consideration in the deployment of network services. Especially in communication systems provided by drones, whose capacities in terms of processing and energy may have limitations, the efficient use and management of resources must be guaranteed in order to provide or maintain a desired level of availability. Therefore, this metric is an important factor in the design, planning and deployment phases, considering that some applications may demand specific values for their operation.

In order to provide network services, by leveraging the connectivity capabilities offered by 5G networks and within an NFV context, a set of programmable drones can run

[5]DJI Phantom 3 Pro: https://www.dji.com/es/phantom-3-pro

VNFs, and thus, provide the required services. In this sense, a fleet of programmable drones can offer different network services simultaneously, such as: routing tasks, Internet connectivity, video surveillance services, telemetry, multimedia services, among others. To ensure proper coordination and management of the devices that implement the VNFs or services, it is necessary for an entity or component to perform the corresponding management tasks. In this way, and in an NFV environment, the core management entity, i.e., the orchestrator, can perform the orchestration and management of available resources [44].

Besides, because the provision of services provided by drones is constrained to their autonomy or battery duration, an efficient energy management scheme is of paramount importance in both short and long-term applications. In this regard, a policy or scheme that allows the coordination and replacement of drones, to keep the service in an active state while ensuring a certain level of availability, is essential. As a result of all aforementioned, this section presents a scheme or management system for the deployment and replacement of drones, in which an optimal scheduling algorithm is implemented in order to guarantee the continuity of services, i.e., a level of availability, during a finite time interval.

The proposed scheme is shown in Fig. 6.22 and is composed of two components: (i) a set of drones, which are in charge of executing the VNFs and that constitute the NFVI and (ii) a GCS, where the elements and entities responsible for monitoring, control and management of resources and network services are located. The latter is precisely the component (NFV orchestrator) where the developed algorithm is intended to be executed. In this regard, the complete system can be considered as a network service, which in an NFV environment and using the connectivity benefits provided by 5G such as very low latency and high bandwidth, can offer the optimal or exact drone scheduling for services execution. In addition, because 5G is considered in the design of the system, the proposed solution can also be categorized as a novel 5G use case.

The goal of the proposed algorithm is to carry out an optimal drone scheduling over time, in order to maximize the use of available resources, drones, while providing a reliable communications system guaranteeing continuity in the execution of services. In addition, the information provided by the algorithm can be used as a toolkit in mission planning. Apart from the replacement sequence, the algorithm can inform the services availability level obtained with the deployment of a given number of drones, or in turn, the results can be used to know the number of drones that must be deployed to obtain a given service level. The proposed scheme is characterized based on two different states, which are described below:

- *Service execution state:* In this state the drones, which are equipped with processing and communication devices, execute the VNFs to provide the demanded services. For its operation, the drones are battery powered. Therefore, the autonomy time or drone lifetime is constrained to the capacity of the power supply, the energy consumption of the services to be executed and the consumption of all the elements that allow the operation of the device.

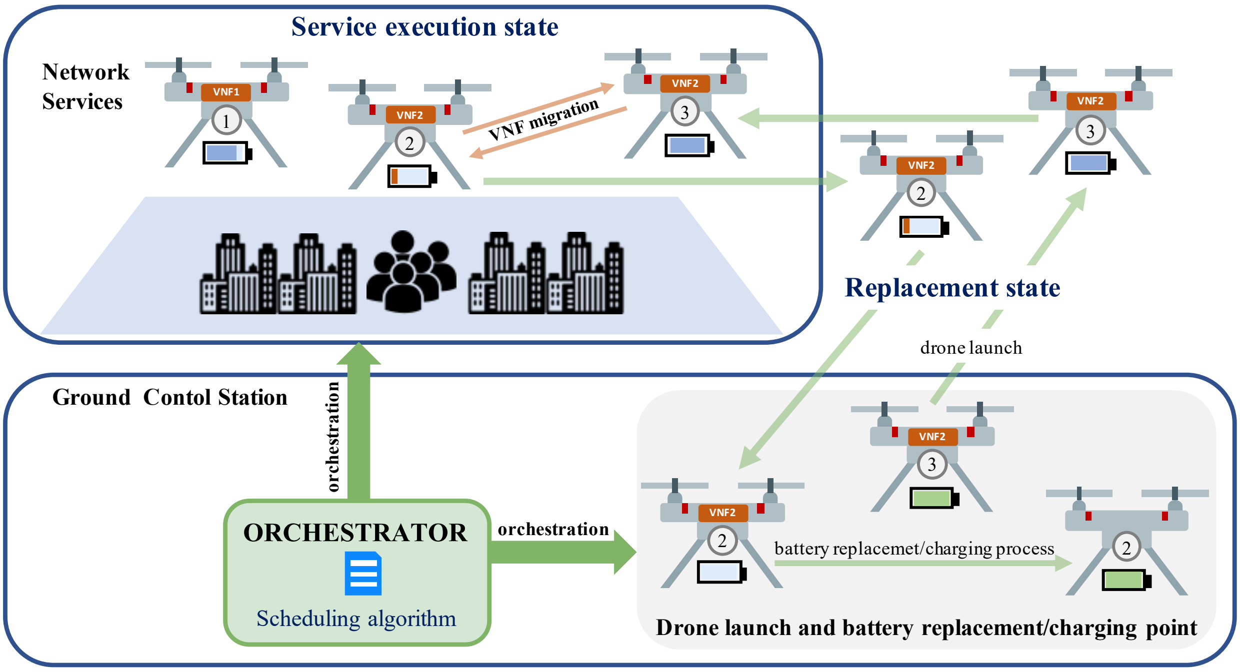


FIGURE 6.22: Overview of the proposed approach for energy management in UAV-enabled communications.

In the proposed approach, the drones can execute the VNFs or services while they are in flight, as shown in the example of Fig. 6.22. Also, for strategic reasons and with the aim of extending the service lifetime, it is possible to consider scenarios in which not all drones remain in flight. For example, for certain applications, such as the provision of connectivity services, some drones after their launch may land on specific locations. In the latter scenario, the service provided by the drones on the ground is not limited to the time that the drone can remain in the air. Even in this case, it is possible to consider the use of a secondary energy source to further lengthen the time of service provision. In any of the proposed scenarios, flying drones or drones on land, the algorithm guarantees the optimal drone scheduling overtime over time. Of course, depending on the scenario, for example if all the drones are flying, the drone replacement procedure should be performed more frequently.

- *Replacement state:* In this state the drones do not provide the services. However, this phase is necessary to guarantee both the migration (transition) of VNFs and the replacement or recharging of drone batteries. Since the battery duration has a finite lifetime, it must be recharged or replaced, being the replacement a more useful and practical option in most cases, due to the agility involved in the process.

In the proposal, the management system located at GCS has all information about available resources (drones and batteries) and service requirements (power demanded by each service and the total required availability time) because all is provided by the users of the system. Through the execution of a scheduling algorithm, the system is able to provide the optimal allocation of drones to cover the demanded services. The scheduling algorithm is executed in the NFV domain, specifically in the NFV orchestrator as depicted in Fig. 6.22.

For the computation of the optimal allocation of drones, the algorithm considers both the consumption related to the service (*service execution state*) and the necessary power to perform the replacement process (*replacement state*). In the proposal, these power consumption are represented by time variables, a *service time* related to the *service state* and a *replacement time* associated with the *replacement state* (see Fig. 6.23). Thus, a percentage of the total battery capacity (majority) is assigned to service execution and the rest is dedicated to the execution of the replacement tasks (service migration). The time variables associated with the operating states of the system are described in detail in Section 6.3.3.3. Regarding the *service time*, this value depends on the power demanded by the service and can vary greatly from one service to another, for example, considering the same battery capacity, a drone running a service that demands high power consumption will have a shorter *service time* compared to another running a service whose power consumption is lower. On the other hand, the *replacement time* is composed of the time needed to perform the migration of the VNFs, from old drone (low battery level) to new drone (high battery level), and the time associated with the round trip flight to (old drone) and from (new drone) the GCS. Since the replacement time is not directly linked to the execution of the service, the value of this parameter could be similar or the same for the different services.

In order to guarantee the continuity of the services in the proposed system, the service migration process starts when the drone that is going to be replaced is active, i.e., when it is running the service, specifically, the migration process begins at the end of the *service time* or at the beginning of the *replacement time* (for a better understanding see the Fig. 6.23 and Fig. 6.25e). After the migration process has been carried out, i.e., when the *replacement time* is over, the replaced drone returns to the GCS, at this time this drone is no longer active but is still part of the system. Once the drone reaches the ground, its battery is replaced or recharged so that according to the indications received by the GCS, it can be assigned for the execution of another service.

According to the aforementioned, in the system, the replacement time is sufficient to guarantee the transition of the services as well as the launching and landing of the drones. In addition, regarding the migration process, among the important aspects to consider are the service hand-off process, from one drone to another, and the exchange of information associated with this procedure. Regarding the latter, in the proposal the exchange of information is accomplished thanks to features such as high connectivity and low latency time provided by 5G technology. Instead, the procedure related to the service transition is a process linked to the type and features of each service. However, it is worth mentioning that Section 6.3.5 presents the results of the application of the proposal in a real case whose values of both the *service state* and the *replacement state* (including the migration process) have been obtained through measurements.

At all times, the management system coordinates the resources that must be allocated (drones to be launched from the ground), because based on the initial information of services and drones, as well as the computations performed by the algorithm, the system is able to estimate the number of available drones, the status of the services, the sequence of replacement to be performed and the availability level reached. Hence, the characterization of the system through the *service state*, the *replacement state* and their corresponding time variables enable the system to operate with the appropriate margins so that the services can be executed continuously during a required time interval while the resource utilization is optimized.

For a better description and understanding of the different states of the proposed energy management scheme, an example is presented below. In Fig. 6.22 is considered an application environment composed by two VNFs, which are expected to be active during a finite time interval. To this end, the system initially uses two drones, drone 1 and drone 2, which execute VNF1 and VNF2, respectively. As time goes by, the management system evidences that drone 2 is draining its battery due to the consumption of the service and the consumption related to its flight. In response to this, and before the drone stops providing the service or in the worst case it stops working and collapses to ground, the system coordinates the sending of another drone. In this case drone 3 is selected, whose energy level is adequate to guarantee the execution of the VNF 2 for a subsequent time interval. At the moment that drone 3 is located at a suitable distance for the establishment of communication with drone 2, the migration of VNF2, from drone 2 to drone 3 is performed, so that the service is not interrupted and remains available. Subsequently, drone 2 returns to the ground station to recharge or replace its battery, so that it can be ready for a new allocation. Thus, drone 2 is available to run the VNF2 or a different VNF, it depends on the decision that is made by the scheduling algorithm and the corresponding management system.

During all the time of operation of the service, all the actions both on land and in the air are coordinated by the management and orchestration systems. In summary, the replacement state includes the launching of the new drone (with high battery level) from the ground station, the return of the old drone (with low battery level) to the ground station, and the service migration process (VNF migration).

In addition, from the example described above, it can be observed that to guarantee a continuous execution of the service and a total availability level (100%), the number of available drones must be at least one unit greater than the number of services. In the example it is verified that to guarantee the continuous operation of VNF1 and VNF2, it is necessary to use 3 drones, drone 1 (VNF1), drone 2 (VNF2) and drone 3 (VNF2).

Also, as aforementioned, the replacement state may include the battery replacement or recharge of it. In the first case, the battery replacement is a process that can generally take less time, for example, in a search or rescue mission, it is possible to use a limited number of drones and a large number of batteries. Meanwhile, in the second case, the battery charging commonly is a slower process, but necessary if the drone is tampering resistant, or if the number of available batteries is limited.

In the proposal, regarding to the replacement state, for practical reasons, has been considered the battery replacement procedure. Nonetheless, the algorithm developed has the flexibility to considering a battery charging procedure. In fact, within the characterization of the system, the battery charging phase could be considered as an additional state, the *battery charging state*. In summary, the proposed strategy bases its operation on a drone scheduling algorithm, which allows to know how many drones are going to be used, how they should be replaced and when the replacement should be made.

6.3.3.2 System Model

The drone scheduling algorithm is intended for providing the information of the optimal drone scheduling over time. In the proposal, the time variable has been divided in time slots of equal duration. Thus, a drone is able to run a service over time (service execution state) during one or several slots according to its capabilities (available energy) and the features of the services that it can run. Similarly, the drone replacement state can last one or more time slots depending on the features of the drones (battery replacement procedure) and the scope of application of network services. A summary of notations that describe the drone scheduling strategy is shown in Table 6.10. Then, these parameters are defined in Section 6.3.3.3.

TABLE 6.10: System parameters of the proposed approach for energy management in UAV-enabled communications.

Parameter	**Description**	Comments/Units
T_A^E	Expected availability time	Time units
T_A^R	Reached availability time	Time units
A_v	Service availability	Percentage, $A_v \in \{0, ..., 100\}$
A_vS	Service availability per services	Percentage, $A_vS \in \{0, ..., 100\}$
NS	Number of services (VNFs)	Integer number
S_j	Service indentifier	$j \in \{1, ..., NS\}$
T_{init}^j	Initial time of service j	Time units
P_d^j	Power demanded by the service j	Power units
ND	Number of available drones	Integer number
D_k	Drone identifier	$k \in \{1, ..., ND\}$
C_B^k	Battery capacity of drone k	Power x Time units
$T_B^{d,k}(P_d^j)$	Battery lifetime of drone k for service j	Time units
$T_B^{r,k}$	Battery replacement time of drone k	Time units

6.3.3.3 Definitions

1. *Expected availability time* (T_A^E): Also defined as service availability time, it represents the time interval where the services are expected to be active/available.

2. *Reached availability time* (T_A^R): Time interval during which the services are active/available.

3. *Number of services* (NS): The set of services or VNFs that are executed by the drones during a certain time period.

4. *Initial time of service j* (T_{init}^j): Time instant from which the service j is available/active. Initial time in which the service availability is analyzed.

5. *Power demanded by the service j* (P_d^j) : Power demanded by each service j to be executed. Each service may demand a different amount of power. In general, this parameter represents the consumption demanded by the services, and in practical implementations, its units can also be given in terms of electric current (e.g., [mA]).

6. *Number of drones available* (ND): Set of drones that are part of the system.

7. *Battery capacity of drone k* (C_B^k): It represents the amount of energy that can be stored in a battery of each drone k. Moreover, in practical implementations this capacity can be expressed in terms of electric charge, i.e., electric current per time units (e.g., [mAh]).

8. *Drone battery lifetime* ($T_B^{d,k}(P_d^j)$): Each drone k with a battery capacity (C_B^k), can execute a service that demands a power level (P_d^j) during a time period $T_B^{d,k}$. This relationship can be expressed as follows:

$$T_B^{d,k}(P_d^j) = \frac{C_B^k}{P_d^j} \tag{6.54}$$

   This time variable represents the time interval linked to the *service execution state.*

9. *Battery replacement time* ($T_B^{r,k}$): This time variable is linked to the *replacement state* of a drone k. The ($T_B^{r,k}$) includes the time associated with the sending of the new drone (drone with high level of energy supply), the time demanded to perform the migration process of the services, and the time needed for the old drone (drone with low level of energy supply) to reach the ground station (charging point).

   A pictorial representation of the time variables related to the two states that characterize the system is shown in Fig. 6.23.

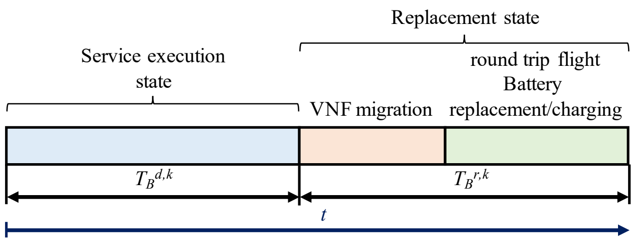


FIGURE 6.23: Time variables of the drone scheduling strategy.

6.3.3.4 Metrics

To assess the performance of the drone scheduling algorithm two metrics have been defined.

1. *Services availability* (A_v): Also defined as the total availability of services, and expressed as a percentage, this metric shows the ratio between the time that all

the services are available and the expected availability time. If the $(T_A^E) = (T_A^R)$, i.e., all the services are available during all the time required, the $(A_v) = 100$ %, otherwise this value will be lower. The service availability can be expressed as:

$$A_v = \frac{T_A^R}{T_A^E} \cdot 100\% \tag{6.55}$$

2. *Services availability per services* (A_vS): This metric provides the information of the mean availability value of all services. A service j can reach an availability level equal to $A_{v,j}$, if this value is small compared to the availability of the other services, the A_v value will also be small and equal to $A_{v,j}$. For this reason, the (A_vS) metric is defined, because it is less restrictive and weights all availability values, in order to provide information on the behavior of all the services that are part of the system. As a consequence, the A_vS value will always be greater or at most equal to the A_v value. The service availability per services is defined by:

$$A_vS = \frac{\sum_{j=1}^{NS} A_{v,j}}{NS} \cdot 100\% \tag{6.56}$$

6.3.3.5 Assumptions

The following assumptions are made for the practical implementation of the algorithm:

1. In practical implementations each programmable drone can execute more than one VNF concurrently. However, to simplify the analysis, in the proposed scheduling scheme, each drone k can run only one VNF or service j. This consideration is valid, since the execution of several services in the same drone would correspond to the consumption of different power levels. Thus, the processing of only one VNF and the analysis of its consumption could represent a summarized value of all the services that are executed in the drone.

2. Similar to the previous consideration, the strategy considers that a service j can only be executed by one drone k at the same time. This with the aim of simplifying the analysis in the distribution of drones and services.

3. In the proposed system, any drone has the ability to execute any VNF. Likewise, any battery can power any available drone. In this sense, all available resources, drones and batteries, can be reused when demanded. It is clear that the services execution is limited to the capabilities of drones and the features of services.

4. In the proposal it is considered that all services work simultaneously, i.e., all services are available as long as the system has the resources for their execution.

5. The ND must be at least equal to NS. Although, as discussed in Section 6.3.3.1, to ensure the execution of services without interruption, ND should be at least greater than or equal to $NS+1$ $(NS \geq ND+1)$. If $ND < NS$, then $A_v = 0\%$ and

$A_vS = 0\%$. The aforementioned consideration is mandatory in the initial execution or first drone allocation process, after this stage the algorithm is continuously evaluating the amount of available resources. Therefore, in the following allocation processes ND could be smaller than NS, in which case the algorithm analyzes the requirements of services to perform the corresponding allocations.

6. In the *replacement state*, the drone that is replaced arrives at the ground station, with a very low battery level (fully discharged battery). For the replacement process, a battery that has previously been charged up to 100% of its capacity is used (fully charged battery). Similarly, if the complete drone must be replaced and not just its battery, the device that replaces it will be equipped with a battery charged to the maximum level. This consideration is also valid for the drones that are assigned for the first time, i.e., the drones used in the first allocation process have their batteries fully charged. In addition, the system has enough batteries to guarantee the replacement process of all the drones that demand them.

7. In the proposal is assumed that communication requirements such as very low latency and high bandwidth capabilities are provided by 5G technology. Moreover, the level of connectivity provided by 5G allows for proper communication and coordination between the different components within the system.

6.3.4 Drone Scheduling Strategy

In this section, first the drone scheduling procedure is presented in Section 6.3.4.1; then the complexity of the problem is discussed in Section 6.3.4.2.

6.3.4.1 Drone Scheduling Algorithm Procedure

The drone scheduling strategy consists of systematically computing the optimal set of available of drones to execute the services. To this end, the strategy follows the guidelines described in the *execution* and *replacement* states. The scheduling process stars with the individual analysis of the execution of each service for each available drone $(T_B^{d,k}(P_d^j))$, then goes through the following three phases to obtain the optimal allocation of drones to run services.

a) Computation of Combinations: In order to find the exact or optimal allocation of drones to run services, the algorithm, based on a brute-force search combinatorial method, explores all possible combinations of drones and services. Once all the possible combinations are obtained the algorithm determines those that meet the system requirements (valid combinations). Subsequently, this set of combinations are sorted in descending order according to the A_v metric. At the end of this phase, the best combination of drones (the first combination in the list from the top) is selected. In this context, this best combination represents the optimal set of drones whose services execution produce the highest A_v value in the system.

b) *Resource allocation and services evaluation:* Once the best combination of drones and services is obtained, the drones are allocated to their corresponding services. Afterwards, the A_v and A_vS metrics are computed; specifically, the A_vS metric is computed because the A_v metric was already obtained in the previous phase. If the A_v reached is equal or greater than the desired value, i.e., $T_A^R \geq T_A^E$, the algorithm stops its execution; otherwise it analyzes the current availability level of all services ($A_{v,j}$) and the available resources (drones that have not been used) to proceed with the next allocations.

   The analysis of the available resources is carried out in the following phase. While, the analysis of availability per services is part of this phase and corresponds to the services evaluation, which is a procedure performed in order to reach the highest possible A_v or T_A^R value (both parameters completely equivalent), from the second allocation process. In this regard, based on the information of the last allocation made, the algorithm lists the services in descending order according to the $A_{v,j}$ reached, so that this information can be used in the computation and subsequent allocation of the best combination of drones. In specific, the objective of this process is to provide additional information to the algorithm in order to allocate the drones with the highest battery capacity to the services with the lowest current $A_{v,j}$ values. In summary, the services evaluation contributes that the drones are allocated starting with the service with the lowest $A_{v,j}$ value. The mechanism described in this phase ensures an increasing A_v and an efficient resource utilization.

c) *Verification of available resources:* Throughout the scheduling process, the algorithm must know the status of the executed services (T_A^R) and the information of the resources in the system. Especially from the second allocation process, the algorithm has to identify the resources used and available in order to perform the computation of combinations and the subsequent allocation of resources. In this regard, the algorithm has to evaluate at each time the information of the drones in the system, considering that this information consists of: drones used, drones that have not been used, drones that must replace their battery and drones whose battery has been replaced and are ready for a new allocation.

In an iterative process, the algorithm follows the phases described above and continuously calculates the best scheduling of drones to execute services. This procedure is carried out constantly until any of the two stopping criteria is met. The first criterion is the A_v value reached, if after an allocation process the $A_v = 100\%$, the algorithm stops its execution. The second stop criterion is related to the number of available drones in the system, considering that this number is made up of drones that have not been used (not allocated yet) and drones whose battery has been replaced (or charged). In the event that the system does not have the necessary resources (drones) to perform the corresponding allocations, the algorithm stops its execution, under this condition an $A_v \neq 100\%$ will be achieved. Finally, the algorithm provides the information of the (T_A^R), A_v and A_vS reached. The developed algorithm guarantees the best drone scheduling for services execution over time, by analyzing all possible drones-services combinations. However, the problem tends to growth as the NS and ND increase, which can be a problem if the capacity or processing time are constrains within the system.

The phases discussed above are implemented in the algorithm through different steps. The drone scheduling algorithm is explained in Fig. 6.24 and each step is described in detail based on the example depicted in Fig. 6.25. In this example $T_A^E = 7$ time slots, and for simplicity, the NS and ND are limited to 2 and 4, respectively. A pictorial representation of the required services is shown in Fig. 6.25a. Moreover, the example considers a $T_B^{r,k} = 2$ time slots, the first time slot corresponding to the VNF migration process and the round trip time of the drone, and the second time slot referred to the time for battery replacement. In addition, to better understand the proposed strategy in Table 6.11 is provided a summary of parameters related to the processing of the combinations and the analysis of the drones in the system. The different steps that are part of the algorithm are explained as follows.

TABLE 6.11: Parameters related to the processing of combinations and drones in the scheduling algorithm.

Parameter	**Description**
$PairDronServ$	Pair of a drone and a service. A pair is used to describe the individual analysis of the execution of a service S_j running on a drone D_k
$TotalDronServ$	Total number of pairs of drones and services
$CombDronServ$	Combination of a set of drones to run a set of services. A combination of drones and services, which is commonly referred to as a combination, is composed of different pairs of drones and services
$AllCombDronServ$	Set of all possible combinations of drones and services
$NumAllComb$	Total number of all possible combinations. This number is given by Eq. 6.58
$ValCombDronServ$	Set of valid combinations of drones and services. The $ValCombDronServ$ is a subset of $AllCombDronServ$
$NumValComb$	Total number of valid combinations. This number is given by Eq. 6.59
$SortCombDronServIDs$	Sorted list of the identifiers of the analyzed combinations. To obtain this list, the combinations are sorted in descending order according to the A_v reached
$ReplaceDrones$	Set of drones whose battery must be replaced
$AvDrones$	Set of drones that has neither been used nor allocated in the system
$ReadyDrones$	Set of drones whose battery has been replaced. These drones can be used for a new allocation process
$TotAvaDrones$	Total number of drones that are available in the system. This number is given by Eq. 6.62

- *Input parameters:* The input parameters comprise the T_A^E value, and the information of services and drones, as shown in Table 6.12a and Table 6.12b, respectively.

TABLE 6.12: Information of services and drones.

(A) Information of services.

S_j	T_{init}^j [$Time\ slots$]	P_d^j [$Power\ units$]
1	0	1
2	0	1

(B) Information of drones.

D_k	C_B^k [$Power \times time\ slots$]	$T_B^{r,k}$ [$Time\ slots$]
1	4	2
1	3	2
3	3	2
4	1	2

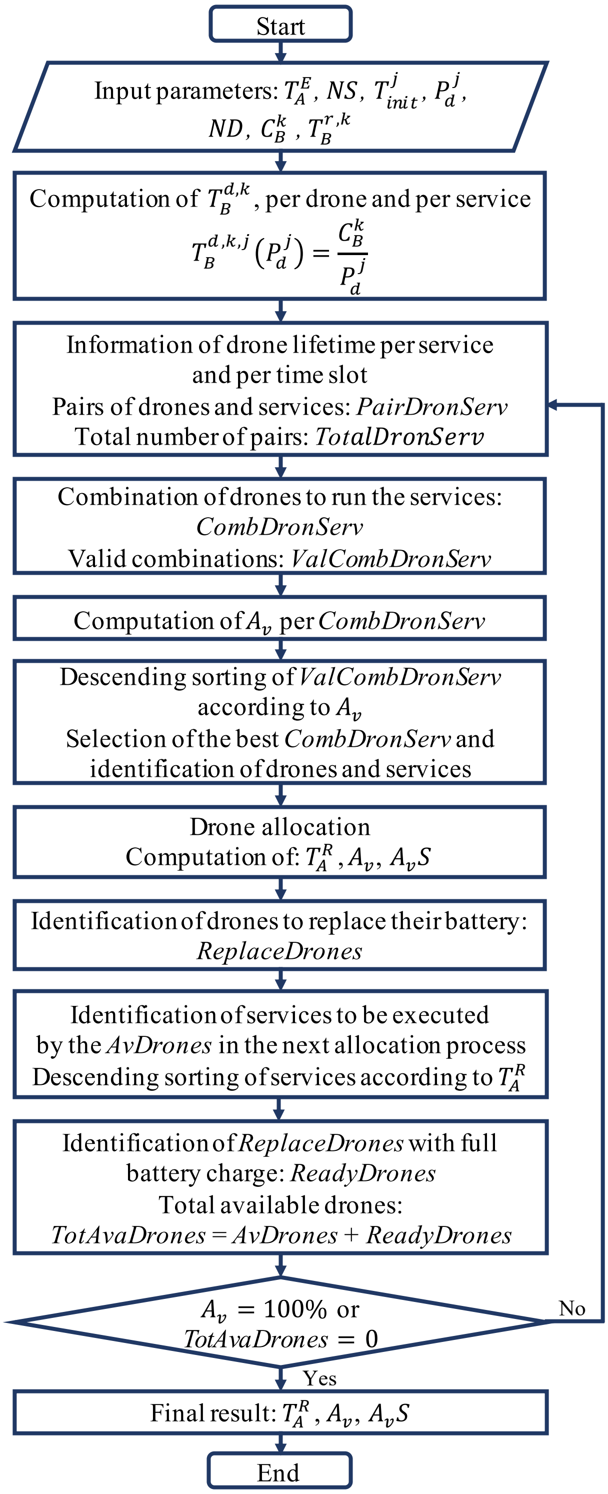


FIGURE 6.24: Energy-aware algorithm for drone scheduling.

- *Computation of* $T_B^{d,k}(P_d^j)$*, per drone and per service:* For the computation of these values, Eq. 6.54 is used. The Table 6.13a presents all possible values of battery lifetime per drones $(D_1, ..., D_4)$ and per services (S_1, S_2).

- *Information of drone lifetime per service and per time slot:* In this step is provided the same information displayed in Table 6.13a, but disaggregated in pairs of drones and services ($PairDronServ$), as shown in Table 6.13b. Further, in this step the information of $T_B^{d,k}(P_d^j)$ is presented in time slots, following the discrete time model discussed in Section 6.3.3.2. At this point, the algorithm provides the information of all possible drone options to execute the services individually. The total number of pairs drone-services ($TotalDronSer$) is given by:

$$TotalDronServ = NS \cdot ND \tag{6.57}$$

  In the proposed example, with $NS = 2$ and $ND = 4$ the $TotalDronServ = 8$ pairs of drones-services, from $PairDronServ\,1$ to $PairDronServ\,8$, see Table 6.13b. Therefore, in this table are represented all possible service lifetime values depending on the drones to be used. For instance, if S_1 would be executed by drone D_4 ($PairDronServ\,7$) the service lifetime would be $T_B^{d,4}(P_d^1) = 1$ time slot, instead if S_1 would be run on drone D_1 ($PairDronServ\,1$) the service lifetime would several times greater and equal to $T_B^{d,1}(P_d^1) = 4$ time slots. The information of each $PairDronServ$ will be used in the next step of the algorithm to analyze the joint action of drones to run services, in order to obtain the maximum possible A_v and A_vS values in every allocation process.

- *Combination of drones to run the services:* Since all services run simultaneously, the different combinations of drones and services ($CombDronServ$) must be analyzed. Hence, the algorithm from a group of ND drones must obtain a set of all possible combinations of drones to run NS services ($AllCombDronServ$). In this context, the total number of combinations ($NumAllComb$) to be processed is given by the analysis of $NS \cdot ND$ pairs ($TotalDronSer$, see Table 6.13b) taken NS at a time, and can be expressed as:

$$NumAllComb = \binom{ND \cdot NS}{NS} = \frac{(NS \cdot ND)!}{NS! \cdot (NS \cdot ND - NS)!} \tag{6.58}$$

  The $NumAllComb$ obtained in this step is critical, because it contributes largely to the growth of complexity of the problem. For instance, $NS = 8$ and $ND = 10$, produce over 28 billions of combinations to be processed.

  In accordance with the criteria adopted for the drone scheduling strategy, see Section 6.3.3.5, not all $AllCombDronServ$ are valid. For instance, in the example (see Table 6.13b), the combination of $PairDronServ\,1$ (D_1, S_1) and $PairDronServ\,2$ (D_1, S_2) is not valid, because the same drone (D_1) is assigned to different services (S_1, S_2) at the same time. Likewise, the combination of $PairDronServ\,3$ (D_2, S_1) and $PairDronServ\,5$ (D_3, S_1) is also not valid, because the same service (S_1) is executed by two drones (D_2, D_3) concurrently. In contrast, the combination of $PairDronServ\,1$ (D_1, S_1) and $PairDronServ\,4$ (D_2, S_2) is valid, because once selected the drones D_1 and D_2 can be allocated to the services S_1 and S_2, respectively. Considering this, the total number of valid combinations ($NumValComb$) is given by:

$$NumValComb = \frac{ND!}{(ND - NS)!} \tag{6.59}$$

In the example, $NS = 2$ and $ND = 4$ produce $NumAllComb = 28$ combinations and $NumValComb = 12$ combinations, the latter shown in Table 6.13c.

TABLE 6.13: Information of drones, services and combinations.

(A) Information about battery lifetime of drones for services S_1 and S_2.

D_k	$T_B^{d,k}(P_d^1)\,[Time\,slots]$	$T_B^{d,k}(P_d^2)\,[Time\,slots]$
1	4	4
2	3	3
3	3	3
4	1	1

(B) Information of pairs of drones and services and drone lifetime per time slot.

$No.Pair$	D_k	S_j	$Drone\,per\,time\,slot$
1	1	1	1 1 1 1
2	1	2	1 1 1 1
3	2	1	2 2 2 0
4	2	2	2 2 2 0
5	3	1	3 3 3 0
6	3	2	3 3 3 0
7	4	1	4 0 0 0
8	4	2	4 0 0 0
		Time Slot	1 2 3 4

(C) Combinations among available drones to run the services.

$No.Comb.$	$Combination\,(No.Pair)$	$Drones$	$Services$
1	1, 4	1, 2	1, 2
2	1, 6	1, 3	1, 2
3	1, 8	1, 4	1, 2
4	2, 3	1, 2	2, 1
5	2, 5	1, 3	2, 1
6	2, 7	1, 4	2, 1
7	3, 6	2, 3	1, 2
8	3, 8	2, 4	1, 2
9	4, 5	2, 3	2, 1
10	4, 7	2, 4	2, 1
11	5, 8	3, 4	1, 2
12	6, 7	3, 4	2, 1

- *Computation of A_v and A_vS per $CombDronServ$:* Using the information obtained of $ValCombDronServ$ in Table 6.13c and the $T_B^{d,k}(P_d^j)$ value, given in terms of time slots, in Table 6.13b, it is possible to compute the A_v and A_vS metrics, for each of the $CombDronServ$. The Table 6.14a shows the different A_v and A_vS

values for each $CombDronServ$ of Table 6.13c. In this table, the metrics have been rounded to the lower bound. An example of the computation of these metrics for the $CombDronServ\,1$ is provided below. In the case of $CombDronServ\,1$ (composed by $PairDronServ\,1$ and $PairDronServ\,4$), D_1 is allocated to S_1 during $T_B^{d,1}(P_d^1) = 4$ time slots, while D_2 is allocated to S_2 during $T_B^{d,2}(P_d^2) = 3$ time slots.

$$A_{v\,Comb1} = \frac{3\,[time\,slots]}{7\,[time\,slots]} \cdot 100\% = 42.86\% \tag{6.60}$$

$$A_vS_{\,Comb1} = \frac{\frac{4\,[time\,slots]}{7\,[time\,slots]} + \frac{3\,[time\,slots]}{7\,[time\,slots]}}{2} \cdot 100\% = 50\% \tag{6.61}$$

- *Selection of the best CombDronServ:* The $ValCombDronServ$ are sorted in descending order according to their A_v value (see Table 6.14a), using a quick sort method. The algorithm provides a list with these values and the combination with the best value, the first in the list is selected. Even if there is more than one better combination, the first one in the list is always selected. In the example, the sorted list of all $ValCombDronServ$ is:

  $Sort\,CombDronServ\,IDs$:= {1,2,4,5,7,9,3,6,8,10,11,12}

  Where, the $CombDronServ\,1$ is the best combination, while the $CombDronServ\,12$ has the lowest A_v level. After selecting the best $CombDronServ$, in the example $CombDronServ\,1$, the algorithm proceeds to identify the drones and services belonging to that combination, as shown in Table 6.14b.

- *Drone allocation and computation of T_A^R, A_v and A_vS:* In this step, with the information of the best combination, the algorithm allocates the drones to their corresponding services over time. As shown in Table 6.15a, D_1 is allocated to S_1 and D_2 is allocated to S_2. The Fig. 6.25c illustrates this allocation procedure, and in this figure is not only represented the *execution state* ($T_B^{d,k}(P_d^j)$, blue color), but also the *replacement state* ($T_B^{r,k}$, orange and green colors). Then, the performance metrics A_v and A_vS are computed. In this particular example the values reached are $A_v = 42\%$ and $A_vS = 50\%$. This is the initial allocation of drones, and since none of the stop criteria have been met, i.e., $A_v \neq 100\%$ and the system has available resources (drones D_3 and D_4), the algorithm continues its execution process.

- *Identification of drones to replace their battery (ReplaceDrones):* The algorithm must constantly monitor the drones used, so that when they finish their execution (T_A^R), they have to change to the *replacement state* (*ReplaceDrones*). In the example, in the first allocation drones D_1 and D_2 have been used, and as can be seen in Fig. 6.25d, drone D_2 must start its replacement process in time slot 4, instead D_1 must perform this procedure in time slot 5.

TABLE 6.14: Computation of metrics for $ValCombDronServ$ and selection of the best combination.

(A) Computation of A_v and A_v for all $ValCombDronServ$.

$No.Comb.$	$Combination\,(No.Pair)$	$A_v(\%)$	$A_vS(\%)$
1	1, 4	42	50
2	1, 6	42	50
3	1, 8	14	35
4	2, 3	42	50
5	2, 5	42	50
6	2, 7	14	35
7	3, 6	42	42
8	3, 8	14	28
9	4, 5	42	42
42	4, 7	14	28
11	5, 8	14	28
12	6, 7	14	28

(B) Information of pairs of drones and services belonging to the best combination.

$No.Pair$	D_k	S_j	$Drone\,per\,time\,slot$
1	1	1	1 1 1 1
4	2	2	2 2 2 0
		Time Slot	1 2 3 4

- *Identification of services to be executed by the available drones* ($AvDrones$) *in the next allocation process:* To continue with the drone scheduling process, from the first allocation, an analysis of the priority of the executed services is carried out. This level of priority is given based on the (T_A^R) parameter of the services. Thus, a service with a lower (T_A^R) value will have a higher priority level to be processed in the next drone allocation step. In this way, the algorithm will allocate the drones with the highest ($T_B^{d,k}(P_d^j)$) values to the services with the highest priority levels. The priority in the execution time of the services is analyzed at the end of the first allocation process, since the start time of all the services is the same ($T_{init}^j = 0$).

  The priority information of the services is used in the computation of the combinations and in the selection of the best combination, from the second allocation. This process is carried out to guarantee an efficient and uniform allocation of the drones, otherwise a specific service could achieve a T_A^R value much higher than the others. This situation must be avoided since the services must be executed simultaneously with the objective of always reaching an A_v as large as possible. In the example (see Fig. 6.25d), S_2 ($T_A^R = 3$ time slots) has higher priority level compared to S_1 ($T_A^R = 4$ time slots). Therefore, later in the computation of all combinations this information will be considered so that the selection of the best combination allows a drone allocation ($AvDrones = 2$, $D3$ and $D4$) that favors the execution of S_2, as shown in Fig. 6.25e.

The priority level of services is also useful in the case that $ND < NS$, situation that may occur after the first drone allocation. In this particular case, the priority allows to know the services that must be attended, with the available resources. In this scenario, the combinations are computed with the number of available drones.

- *Identification of total available drones (TotAvaDrones):* The total number of drones available in the system include: drones that have not been used, which in the case of the example are D_3 and D_4 ($AvDrones$), and drones whose battery has been replaced and are ready ($ReadyDrones$) to be used when the system demands them. The $ReadyDrones$ are the $ReplaceDrones$ that have completed the *replacement state*. In the example proposed, at this point, the first drone allocation has been performed, therefore the system does not have $ReadyDrones$. Under these conditions, the drones available are D_3 and D_4. In the next stages, the algorithm could have $ReadyDrones$ available, once they exceed the *replacement state*.

  The total number of available drones ($TotAvaDrones$) in the system can be expressed as:

$$TotAvaDrones = AvDrones + ReadyDrones \tag{6.62}$$

- *Iterative drone allocation:* In an iterative process of computation of: combinations, selection of the best combination, drone allocation, priorities of services and metrics; the algorithm continues its execution until either of the two stopping criteria is met ($A_v = 100\%$ or $TotAvaDrones = 0$). Continuing with the example, in the second iteration the algorithm allocates D_3 to S_2 and D_4 to S_1, as shown in Table 6.15b and in Fig. 6.25e. The results after this procedure are $A_v = 71\%$ and $A_vS = 78\%$. In a similar way to the process carried out at the end of the first iteration, once the services have been completely executed by drones D_3 and D_4, these devices pass to the *replacement state*. Specifically, this status will be reached at time slot 5 for D_4 and at time slot 6 for D_3, as represented in Fig. 6.25f. For practical reasons, the time window in proposed example has been limited to 8 time slots.

  After the second iteration has been completed, the algorithm checks the total number of available drones. At this point, given that $AvDrones = 0$, the algorithm checks if any of $ReplaceDrones$ has become $ReadyDrones$. In the example, the unique drone that meets this condition is D_2, which after the corresponding computations is allocated to S_1, as shown in Table 6.15c, and in Fig. 6.25g. At the end of the third iteration the $A_v = 85\%$ and $A_vS = 92\%$. Once the third iteration is finished, the scheduling process continues in the same way as in the previous iterations and according to the steps described in the Fig. 6.24. The fourth iteration is the last in the example. In this iteration $D1$, whose battery has been replaced, is allocated to S_2, as seen in Fig. 6.25h and in Table 6.15d. Resulting in the final values: $T_A^R = T_A^E = 7$ time slots for both services (S_1 and S_2), $A_v = 100\%$ and $A_vS = 100\%$. Once these values are obtained, the algorithm stops its execution.

A summary of the complete drone allocation procedure is depicted in Fig. 6.25b. The final drones sequence is: D_1, D_4 and D_2 for S_1; and D_2, D_3 and D_1 for S_2. Moreover, as a relevant result the algorithm allows to know the number of drones (resources) needed to reach a certain availability level. In the example, during a time window of $T_A^E = 7$ time slots, with 4 drones and an optimal scheduling, the system can reach an $A_v = 100\%$ in the services execution.

TABLE 6.15: Progressive allocation of drones to fulfill the network services.

(A) Initial drone allocation.

S_j	$Drone\,information\,per\,time\,slot$
1	1 1 1 1 0 0 0
2	2 2 2 0 0 0 0
Time Slot	1 2 3 4 5 6 7

(B) Second drone allocation.

S_j	$Drone\,information\,per\,time\,slot$
1	1 1 1 1 4 0 0
2	2 2 2 3 3 3 0
Time Slot	1 2 3 4 5 6 7

(C) Third drone allocation.

S_j	$Drone\,information\,per\,time\,slot$
1	1 1 1 1 4 2 2
2	2 2 2 3 3 3 0
Time Slot	1 2 3 4 5 6 7

(D) Final drone allocation.

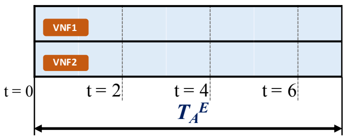
S_j	$Drone\,information\,per\,time\,slot$
1	1 1 1 1 4 2 2
2	2 2 2 3 3 3 1
Time Slot	1 2 3 4 5 6 7

6.3.4.2 Complexity Analysis

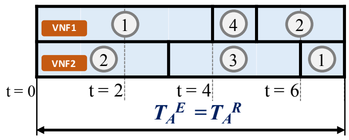
The NS and ND values have an impact on the growth of complexity of the algorithm. The growth rate of the problem is not linear, because it depends on the product of $NS \cdot ND$, as shown in Eq. 6.57 and in Table 6.13b. The steps 3 and 4 define the growth of the algorithm. This growth rate as a function of NS and ND can be expressed as:

$$f(NS \cdot ND) = NS \cdot ND + C(NS \cdot ND, NS) \tag{6.63}$$

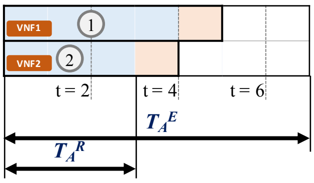
Where the second term is the dominant term within the expression. As described in Section 6.3.4 it represents total set of all combinations (*AllCombDronServ*) that the algorithm must analyze in a mandatory manner in order to find the valid combinations $ValCombDronServ$ to be processed.



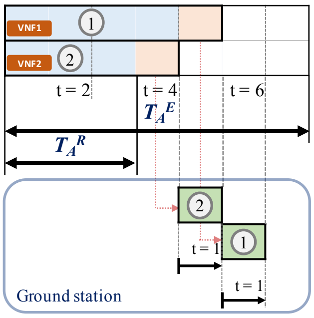
(A) Required network service



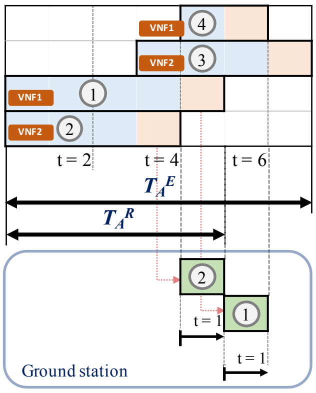
(B) Drone scheduling to execute the services S_1 and S_2 during $T_A^E = 7$ time slots



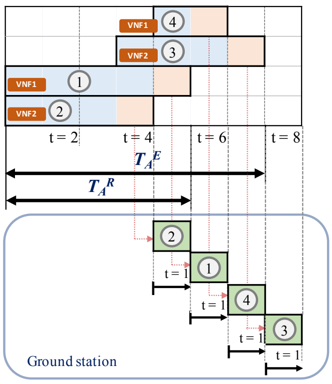
(C) First drone allocation per service, D_1 for S_1 and D_2 for S_2, $A_v = 42\%$ and $A_vS = 50\%$



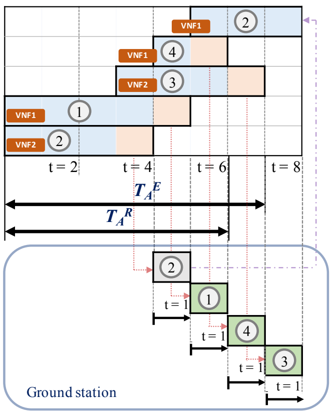
(D) *Replacement state*, battery replacement of drones D_1 and D_2



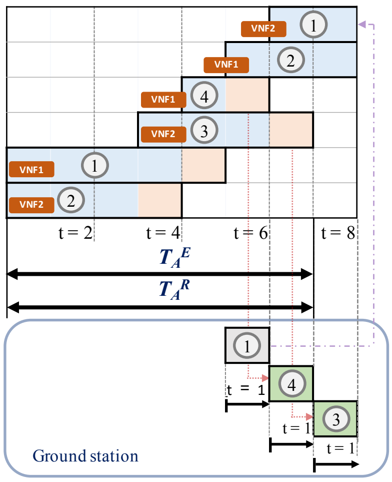
(E) Second drone allocation per service, D_4 for S_1 and D_3 for S_2, $A_v = 71\%$, $A_vS = 78\%$



(F) *Replacement state*, battery replacement of drones D_3 and D_4



(G) Third drone allocation per service, D_2 for S_1, $A_v = 85\%$ and $A_vS = 92\%$



(H) Fourth drone allocation per service, D_1 for S_2, $A_v = 100\%$ and $A_vS = 100\%$

FIGURE 6.25: Example of the energy-aware drone scheduling algorithm.

Thus, according the Big-O classification [91], ignoring the low-order terms, i.e., the first term in Eq. 6.63, the order of growth of the scheduling algorithm is $\mathcal{O}(C(NS \cdot ND,\ NS))$. Hence, this complexity reveals the drawbacks that the algorithm has for the selection of NS and ND values.

6.3.5 Performance Evaluation

To validate the resource planning algorithm proposed in the previous section, it is necessary to define reasonable scenarios that can integrate all the different parameters that should be assessed and provide the complex environment where this type of algorithms is normally applied. The evaluation will then be carried out through extensive simulations using these scenarios. Section 6.3.5.1 describes the simulation setup and the two application scenarios and then, the evaluation results are presented and discussed in Section 6.3.5.2.

6.3.5.1 Simulation Setup

The drone scheduling strategy is evaluated in two different application scenarios: a *Generic Scenario* that uses random values for some parameters, and a *Realistic Scenario* whose values are based on experiments and measurements, as shown in Table 6.16. These scenarios are described in detail below, and for practical reasons, in both scenarios the NS has been limited up to $NS = 7$ services. The scheduling algorithm has been implemented using Matlab (Matlab R2017b).

The *Generic Scenario* has been run on a computer equipped with a 3.33 GHz x 12 cores Intel Core i7 Extreme processor and 12 GB RAM. This simulation leverages parallel processing and multiple CPU cores (up to 6 cores) have been used in each simulation. In order to ensure stability of the results, each case (NS) was repeated 50 times except for $NS = 7$, which was executed only once due to its excessive running time. Results are shown, in Fig. 6.30 and in Fig. 6.30c, with a confidence interval of 95%. The total running time for this scenario (all cases) exceeded 300 hours.

Meanwhile, the *Realistic Scenario* has been executed on a machine with a 3GHz x 4 cores Intel Core i5-7400 processor and 64 GB RAM. In this case parallel processing has also been exploited, all 4 cores have been used during the simulation. Because this scenario is deterministic, only one simulation has been executed for each case ($T_A^E = 5$, $T_A^E = 10$ and $T_A^E = 15$ hours). The execution time for this scenario is around 80 hours, and results are shown in Fig. 6.31.

TABLE 6.16: Summary of simulation parameters for the drone scheduling algorithm.

Scenario	T_A^E	NS	ND	T_{init}^j	C_B^k	P_d^j	$T_B^{r,k}$
Generic	10 hours	1-7	0-11	0, for all j	uniform distributed random value [1- 5] [Ah], for all k	uniform distributed, random value [1- 5] [A], for all j	10 [min], for all k
Realistic	5, 10, 15 hours	7	0-10	0, for all j	3 [Ah], for all k	S1 Router: 292.02 [mA] S2 Router: 292.35 [mA] S3 AP + Router: 371.82 [mA] S4 AP + Router: 373.62 [mA] S5 Telemetry TX: 288.76 [mA] S6 Telemetry TX: 288.23 [mA] S7 Flying: 9000 [mA]	10 [min], for all k

- *Generic scenario:* This scenario corresponds to a very general application environment with the intention of performing an initial validation of the proposed solution. To this end, it is assumed that the different drones can execute the services in the air (with a much higher power consumption), a subset can land on the ground after its launch from the ground control station, or even a hybrid situation can also be possible (different cases are discussed in Section 6.3.3.1).

  The scenario is not particularized for specific applications and it is considered that applications can vary from the provision of video surveillance services to the provision of connectivity services, etc. (this is modeled in the scenario by considering the power demanded by the different services P_d^j to be a random value between 1 and 5 A). In addition, to provide more diversity, batteries capacities C_B^k are considered to be different for each drone, choosing for them random values between 1 and 5 Ah. This assumptions (see Table 6.16), will produce services with different T_A^R values (*service execution state*) varying from $T_A^R = 0.2$ hours (minimum value) to $T_A^R = 5$ hours (maximum value).

  Considering the parameters described above, a $T_B^{r,k} = 10$ minutes (*replacement state*) and a time window of $(T_A^E) = 10$ hours, the algorithm has to perform many transitions among the *service execution state* and the *replacement state* in order to optimally allocate the available resources to the corresponding services which is exactly the situation that is wanted to be forced in this scenario in order to test the algorithm capabilities. The metrics achieved for the different NS and ND values, are shown later in Section 6.3.5.2.

- *Realistic scenario:* The objective of this scenario is to test the algorithm under more real conditions replacing the random values used in the Generic scenario by some other values that may be closer to some real ones.

  In particular the scenario that will be described in this section (Fig. 6.26) shows a set of drones each one with an onboard SBC (they carry an RPi with its own battery) linked through an ad-hoc WiFi network and using a certain FANET (Flying Adhoc Network) routing protocol to guarantee connectivity. The drones are including different VNFs depending on the role they are assuming (Access Point (AP), router, or telemetry transmitter (i.e., video or sensor data).

  In this scenario the energy consumption for a particular drone may depend on many diverse factors. In first place there are two different types of batteries and also drones that are flying and drones that are landed (so depending on the situation the battery that limits the service maybe either one or the other). For the drones that are not flying (the drone battery is not presenting any limitations for them and only the RPi battery is used) the measurements must consider the different WiFi interfaces, the WiFi communications (different traffic including Video, telemetry, routing messages, etc.), CPU load, external hardware etc.

  As it can be appreciated it is not easy in this environment to evaluate how the energy consumption curve will perform for the different drones, how many drones are in fact needed in order to guarantee that the service can be maintained over time (considering a certain replacement time), etc. This is considered to be a suitable scenario so as to validate the combinatory algorithm and the rest of the section will provide more details on the scenario itself and about the validation methodology.

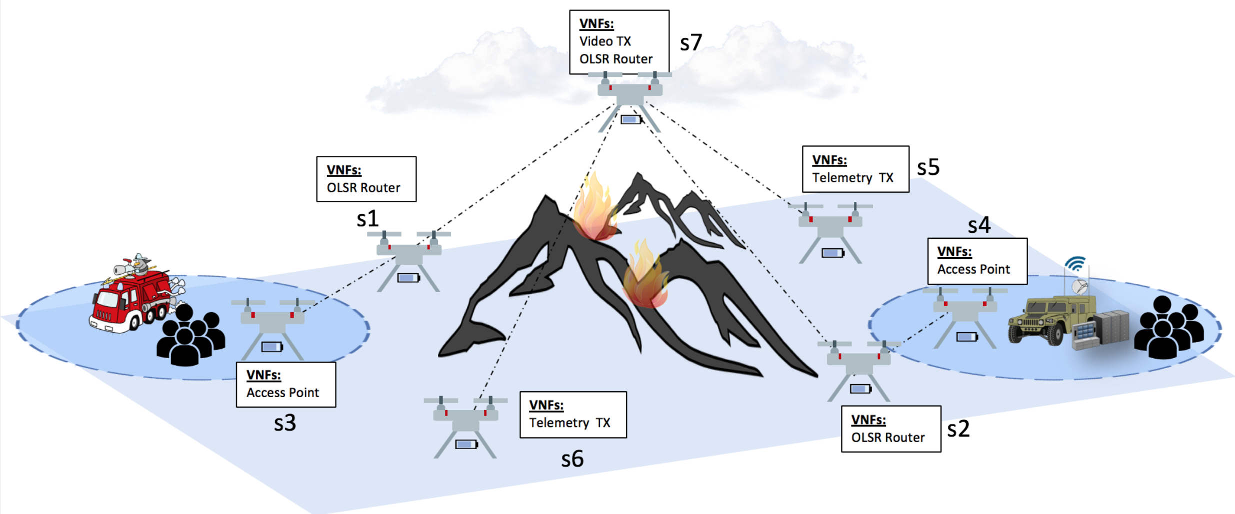


FIGURE 6.26: Drone swarm providing network connectivity in a disaster situation.

A total of seven drones have been considered in this scenario that may represent a natural disaster use case (e.g., earthquake, fire, flood) where drones can enable communications between emergency services islands as seen in Fig. 6.26 drones accommodate different VNFs and play different roles within the network to perform the overall service that will be taken into account by the algorithm:

1. *Optimized Link State Routing (OLSR) Router VNF:* Because of the drone network nature (e.g., node mobile, volatile network) OLSR [181] has been selected as routing protocol. OLSR is a distributed and proactive routing protocol used to establish connections between participant nodes in an ad-hoc wireless network proposed for MANETs and extended for Flying Ad-hoc Networks (FANETs). The main advantage of this type of routing protocol is its dynamic discovery allowing state-less VNFs that prevent the costly process of VNF migration. Drones s1 and s2 in Fig. 6.26 are OLSR routers.
2. *Access Point VNF:* Wireless AP VNFs are used to interconnect wireless communication equipment from emergency services (end user terminals). The selected technology has been the normal 2.4 GHz IEEE 802.11. Drones s3 and s4 in Fig. 6.26 are APs.
3. *Telemetry Transmitter VNF:* Data transmission VNFs have been used in different drones within the network. As it can be appreciated in Fig. 6.26, two types of data transmitters have been specified, (a) telemetry transmitter (32 Kbps flow, that can either represent GPS information or sensor data such as temperature or humidity) and (b) video transmitter in standard quality (200 Kbps flow) that is enough to have an overview of the disaster area by the emergency services. Drones s5 and s6 in Fig. 6.26 are telemetry transmitters.

Drones s1 to s6 are landed and the energy demanded is only related to the network services while drone s7 is flying (Figure 6.26).

In order to validate the algorithm, it is necessary to provide as input a rough estimation of the power demanded by each drone. However, in this realistic environment, besides user traffic (video, telemetry, etc.), numerous factors may affect battery lifetimes. Parameters like the pattern of energy consumption, environmental conditions or battery status are significant factors to take into account in real applications, but there is extremely complex to model them in simulated environments therefore despised. However, there is unpredictable network traffic which is considered to measure energy consumption, such as packets re-transmission, WiFi management packets, routing messages, etc.

The power consumption will be directly measured using a real RPi and a specific power meter. In order to do so, it is required that the RPi resembles the real conditions as stated in the scenario definition in terms of traffic, CPU load and consider the necessary hardware to enable wireless communication since the consumption depends heavily on these parameters.

To calculate all these values, a simulation using ns-3[6] network simulator has been performed. As it can be seen in Fig. 6.26, the simulation includes the seven drones (one of them is flying (s7) while the rest of them are landed). The drones that accommodate the telemetry transmitter VNFs generate one flow to each one emergency service involved on the scene. More details can be consulted in Table 6.17 where the simulation parameters are specified. The trajectory of the flying node (s7) has been precalculated using Matlab and included in the simulation using traces with the ns format.

TABLE 6.17: Simulation parameters for drone operation.

Parameter	**Values**
Traffic	CBR
Telemetry Transmission Rate	32 Kbps
Video Transmission Rate	200 Kbps
Network Protocol	UDP
Routing Protocol	OLSR
Simulation Time	3600 seconds
Number of drones	7
Mobility Model	Static

After this simulation, it will be possible to calculate the traffic that will be processed by each drone, including all the different components that have been previously mentioned. To be able to properly analyze the traffic at each drone the following characterization has been done for the traffic depending on the source and the destination as represented in Fig. 6.28.

1. *Transit traffic:* Received traffic to be forwarded because that drone is not the final destination of the packet. This traffic is telemetry data.
2. *Sink traffic:* Traffic that is consumed by that particular drone. This traffic corresponds to OLSR management or WiFi management packets since telemetry data is consumed out of the analyzed network by the emergency services.

[6] ns-3: https://www.nsnam.org/

3. *Source traffic:* Traffic that is generated at that particular drone. This traffic can be either OLSR management, WiFi management or telemetry data.

Fig. 6.27 shows the average throughput for each drone obtained by the simulation. These results will be replicated into real RPis in order to perform the power measurements. Note that the flying node s7 transmitting video is not represented in the figure. Flight-engines are demanding all the available energy in this drone and power consumption due to traffic processing is negligible in comparison. No power measurement is performed here since the battery that limits the service execution is the one of the drone itself (in Table 6.16 this power consumption is modeled as 9000 [mA] since together with the battery capacity that is used, a 20 minutes flight estimation is obtained which is quite normal for regular drones). In addition, as expected the traffic consumed by drone (only OLSR management) is insignificant compared with transit and generated traffic.

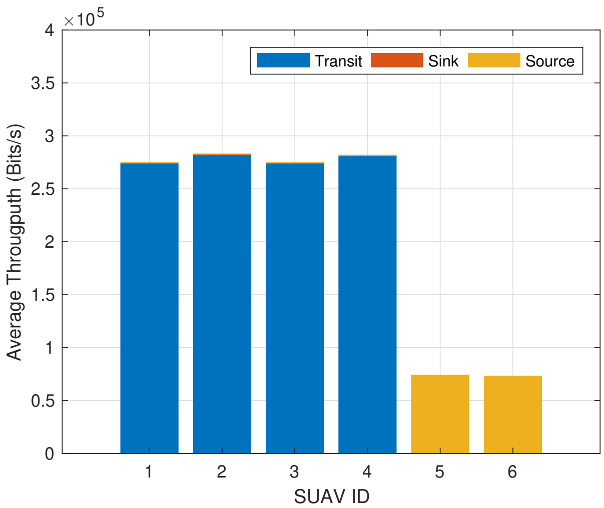


FIGURE 6.27: Average throughput for realistic scenario.

As it has been mentioned, the main purpose of the simulations described in the previous section is to estimate the different data flows (Fig. 6.27) that have to be injected into the RPi in order to emulate the power consumption that it would have in a real scenario.

In order to perform these measurements, the testbed depicted in Fig 6.28 has been built. It includes three different RPis 3B and the Monsoon FTA22D Power Meter[7] (which provides a robust power measurement solution for mobile devices with high accuracy (±1)).

The RPi Source generates two flows, one of them consumed by the RPi hosting the VNF and the second one consumed by the RPi Destination. The RPi that accommodates the VNF is also generating another flow that is consumed by the RPi destination. In this way, we can emulate the traffic involved with each device on the network. To generate this traffic the Iperf[8] tool.

[7]Monsoon FTA22D Power Meter: https://www.msoon.com/
[8]Iperf: https://iperf.fr/

The RPi VNF is then powered by the Monsoon (Vout voltage of 4.2 V) main channel and then the average power is derived from instantaneous current (Fig. 6.29) and voltage, and divided by the duration of the sampling run (200 seconds). After conducting the measurements, it has been verified that the power consumption is not significantly increased unless the network interface is close to the maximum bandwidth. The authors in [175] are finding similar results. The most relevant power increase is due to the use of an external hardware (extra WiFi card to create the AP) carried by drones s3 and s4.

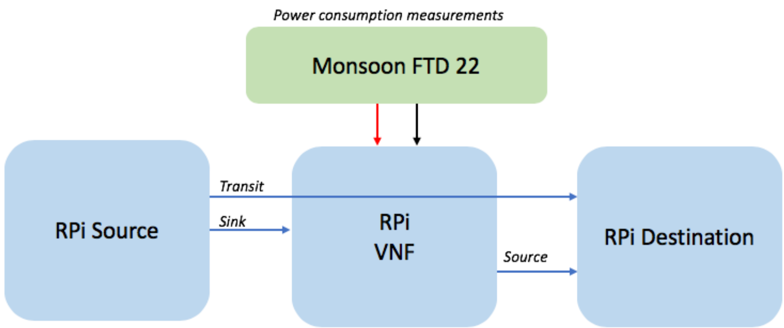


FIGURE 6.28: Power consumption measurements methodology.

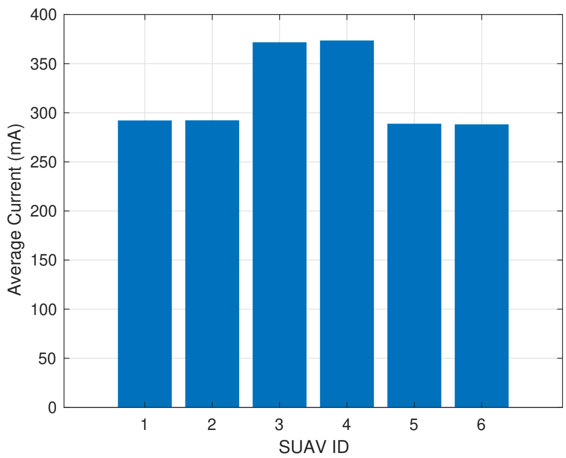
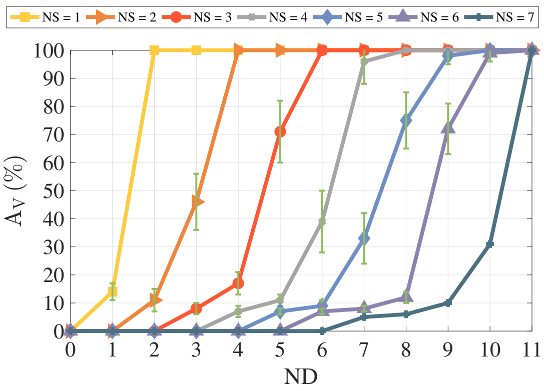


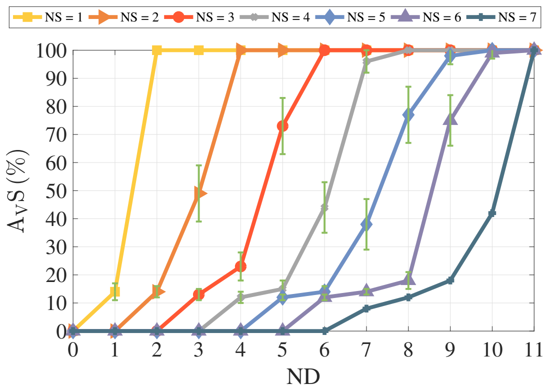
FIGURE 6.29: Average current for realistic scenario.

6.3.5.2 Results

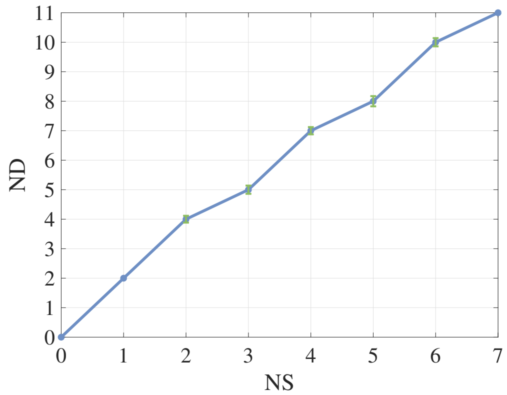
In both scenarios, *Generic* and *Realistic*, the optimal drone scheduling is performed. As a result, the algorithm allows to know the level of services availability achieved when using a given number of available drones. Seen in another way, the proposed strategy can be used to know how many drones need to be deployed to reach a certain services availability level (in the simulations from $A_v = 0\%$ to $A_v = 100\%$). In this context, the results provided by the drone scheduling algorithm can be used in the design, planning and deployment stages of network services executed by drones. Therefore, the information provided by the scheduling strategy can be used for both performance evaluation and system sizing.



(A) Total services availability.



(B) Service availability per services.

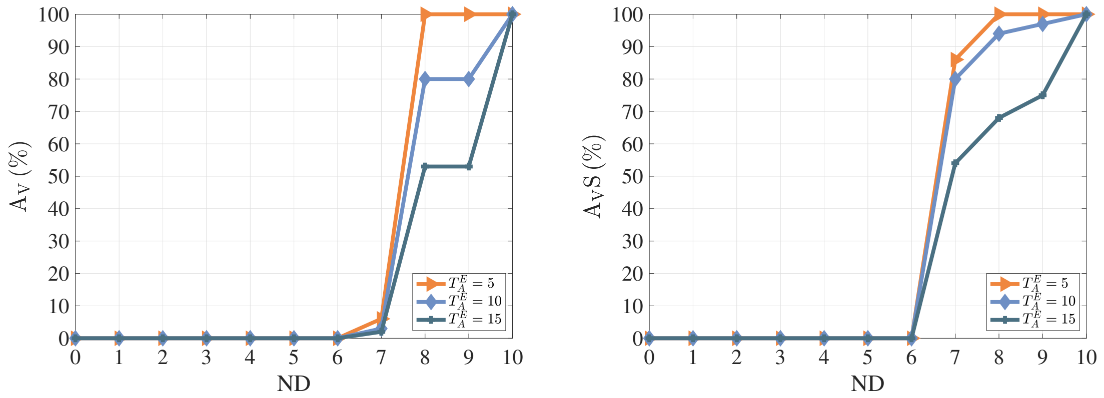


(C) ND to reach an $A_v = 100\%$ - Generic scenario.

FIGURE 6.30: Performance evaluation of the drone scheduling strategy in Generic scenario.

Regarding the *Generic scenario*, whose results are shown in Fig. 6.30a (A_v), in Fig. 6.30b (A_vS) and in Fig. 6.30c ($A_v = 100\%$), the algorithm provides the different availability levels (metrics) for each of the services from $NS = 1$ up to $NS = 7$. As discussed above, this information allows to know the number of drones need to reach a certain A_v value, or the A_v obtained with a given amount of available resources (drones). For example, as shown in Fig. 6.30a, with $NS = 3$ services are needed $ND = 6$ drones to reach an $A_v = 100\%$ (also $A_vS = 100\%$ in this case). Similarly, if as system consists of $NS = 5$ services and demand an $A_v = 90\%$, the number of drones required is $ND = 9$ drones. In the latter example, as shown in Fig. 6.30a, for $NS = 5$ there is no an exact ND value that meets an $A_v = 90\%$, in this case the selection should be rounded the upper bound value ($ND = 9$), due to the ND is an integer number. This rounding operation also ensures that the value obtained is greater than or equal to the requested value.

Another relevant result that can be extracted from the results provided by the algorithm (Fig. 6.30a) is ND as a function of NS to reach an $A_v = 100\%$, as shown in Fig. 6.30c (also $A_vS = 100\%$). This summarized information represents a practical and useful tool that can be used in the design and planning phases of services that have as a constraint the 100% of availability for their operation (e.g., provision of communications services in emergency or search scenarios). For example, in the *Generic scenario*, it is appreciated that for $NS = 6$ are needed at least $ND = 10$ drones to reach and $A_v = 100\%$.



(A) Total services availability.

(B) Service availability per services.

FIGURE 6.31: Performance evaluation of the drone scheduling strategy in a Realistic scenario.

In addition, the results obtained help corroborate the criteria that were considered in the design of the algorithm. For example, as discussed in Section 5.1.2.4, all A_vS values must be equal or greater than A_v values. This condition is verified by establishing a comparison between Fig. 6.30a (A_v) and Fig. 6.30b (A_vS) for the *Generic scenario*, and between Fig. 6.31a (A_v) and Fig. 6.31b (A_vS) for the *Realistic scenario*. For example, for the (*Generic scenario*), with $NS = 5$ services and $ND = 7$ services, $A_v = 33\%$ instead $A_vS = 38\%$.

In the *Generic scenario* in specific, the A_v and A_vS metrics obtained are very similar to each other (always $A_v \geq A_vS$). This situation obeys to one or more of the following considerations: (i) T_A^E and T_A^R are large compared to the size of time slots (minimum amount of time in the system, 10 minutes in the simulations) and (ii) in the final allocations there is not much difference between the T_A^R of all services (i.e., C_B^k) and (P_d^j) have similar values for all services). In the *Realistic scenario* scenario instead, A_v (Fig. 6.31a) and A_vS (Fig. 6.31b) values are different from each other, mainly for $T_A^E = 5$ [hours] and $T_A^E = 10$ hours, this due to the considerable difference of T_A^R for all services, in particular referred to S_7, whose demanded consumption (9 [A]) is much greater than the rest of services. In this case, the algorithm needs to allocate a greater number of drones to execute the service, or in other words to keep the service in active mode a high replacement rate is necessary (i.e., high transition between the *service execution state* and the *replacement state*).

On the other hand, in Fig. 6.31a (A_v) and Fig. 6.30b (A_vS) are shown the performace metrics achieved for the *Realistic scenario*. In this scenario, all cases ($T_A^E = 5$, 10 and 15 [hours]) consider $NS = 7$ services, and as a result the evaluation, the algorithm provides that the required ND to reach $A_v = 100\%$, which in all cases, is equal to $ND = 10$ drones. This value represents the minimum amount of resources (drones and batteries) that the system must use to face a reliable ($A_v = 100\%$) network services deployment. In this scenario, have been considered different T_A^E values, to evaluate the behavior of the strategy in both short-term and long-term real applications. In this regard, the scheduling algorithm performs a few number of allocation procedures when $T_A^E = 5$ hours and $T_A^E = 10$ hours (only S_5 needs to used different resources), instead a high transition between *service execution* and *state* is experienced when $T_A^E = 15$ hours.

6.3.6 Conclusions

In this section, an optimal drone scheduling algorithm is developed, which by leveraging 5G and NFV capabilities, is able perform an efficient energy-aware management of resources for network services provisioning. Through this strategy, it is possible to calculate the required number of drones for a certain degree of service, to be used in real scenarios. The scheduling strategy, based on two states, *service execution* and *replacement*, provides the information about the number of drones and their sequence of replacement to run services and reach a certain availability level, during a finite time interval. Thus, the proposed scheduling algorithm can be used as a useful tool in system sizing and missions planning tasks, in order to provide reliable and safe drone-based network services deployments.

The porposed algorithm can perform the optimal scheduling in both short and long-term applications, and it can be used as a resource/availability planner in a wide variety of real scenarios, such as: emergency scenarios, relief disaster services, search and rescue tasks, among others.

Simulations results validate the performance of the proposal and provide the metrics achieved, as well as the amount of resources needed for the execution of services in different scenarios. The results provided by the simulations can be used to know the level of availability for a certain number of services and available drones. Likewise, these results allow to know the number of drones needed to run services to guarantee 100% of availability level.

Finally, this section presents the evaluation of the proposal for scenarios up to $NS = 7$ services and $ND = 11$ drones, limited by the complexity of the algorithm. Although these limits can be very useful and quite adequate values in many practical scenarios and applications, it is necessary to develop additional strategies in order to be able to handle larger scenarios and in a faster running time. In this regard, the brute-force search solution developed is also useful as a baseline method to design heuristics or metaheuristics approaches.

6.4 Adaptive Energy Management in Multi-UAV Deployments for Multiple Regions

Nowadays, UAVs are frequently present in the civilian environment. However, proper implementations of different solutions based on these aircraft still face important challenges. This section deals with multi-UAV systems, forming aerial networks, mainly employed to provide Internet connectivity and a variety of network services to ground users in different regions. However, the mission duration (hours) is longer than the limited UAVs' battery life-time (minutes). This section introduces the UAV replacement procedure as a way to guarantee ground users' connectivity over time. This section also formulates the practical UAV replacements problem in moderately large multi-UAV swarms and proves it to be an NP-hard problem in which an optimal solution has exponential complexity. In this regard, the main objective of this section is to evaluate the suitability of heuristic approaches for different scenarios. This section presents betweenness centrality heuristic algorithm (BETA), a graph theory-based heuristic algorithm. BETA not only generates solutions close to the optimal (even with 99% similarity to the exact result) but also improves two ground-truth solutions, especially in low-resource scenarios.

6.4.1 Introduction

The unstoppable growth of the UAVs (commonly known as drones) ecosystem during these last years, has been proven to be just the beginning of a near-future global phenomenon. The US Federal Aviation Administration predicts [182] that UAVs providing commercial services will triple over the next five years, and will overtake consumer off-the-shelf UAVs by the year 2024. UAVs will grow eightfold over the next decade and will become the largest segment of the civilian market.

The utilization of multi-UAV systems, because of their rapid deployment, mobility, and flexibility, has recently attracted attention to support/extend the 5G in extraordinary situations (e.g., massified events, natural disasters, infrastructure failures). 5G will certainly bring faster uploading and downloading speeds in combination with a dramatic decrease of the network latency. However, in exceptional or emergency circumstances, the deployment of 5G terrestrial infrastructure may not be economically viable. In addition, the deployment times of these extraordinary on-demand 5G network services should meet the Key Performance Indicators (KPI) defined by the 5G-PPP [183], which states that new deployments must finish within 90 minutes. Accordingly, it is here where UAVs are expected to play a crucial role. If properly deployed and configured, UAV networks can provide fast-ubiquitous 5G access (which is also within the 5G KPIs) employing wireless communications solutions in a diversity of real-world scenarios.

5G UAV missions, e.g., to complement existing cellular networks in high-density environments, deliver network coverage in hard to reach rural areas (Remote Access Networks), or in IoT scenarios, may require the management of moderately large UAV fleets. UAVs mainly act as aerial communication platforms such as *(i)* aerial base stations (BS) (to support existing 5G infrastructure in high traffic demand) [184], or *(ii)* aerial WiFi APs

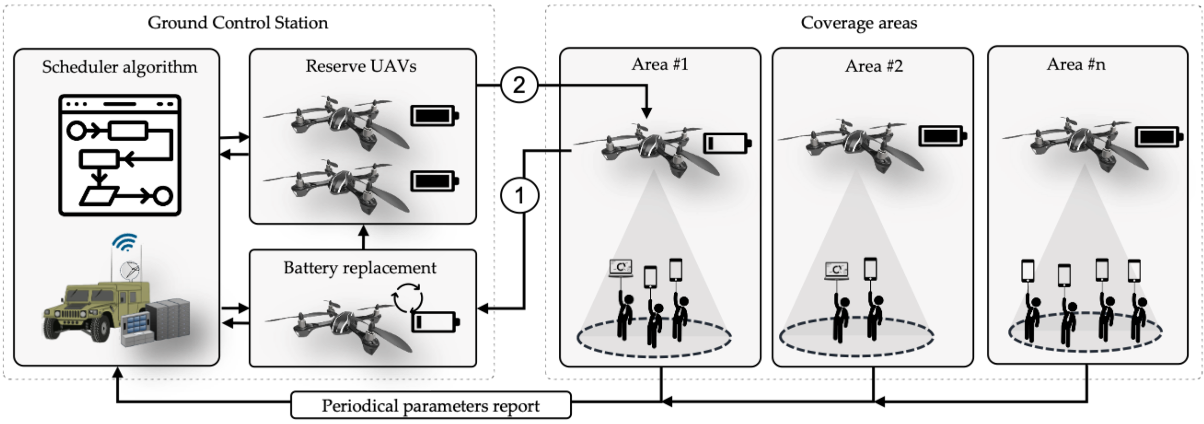


FIGURE 6.32: Typical UAV use case using the proposed methodology for covering multiple regions.

forming a FANET (to create new networks) [185]. So that, UAV research area aims to extend the 5G network (where it has no range) or support the existing 5G network (when it is not enough) using radio solutions as payload [186], e.g., 5G or LTE microcells, multi-hop solutions based on commodity WiFi. When compared to terrestrial antennas, aerial units may have some advantages since they can change their altitude with the possibility of avoiding obstacles, including no geographical restriction on the antenna location. However, these advantages turn into crucial design challenges, such as the optimal positioning, the limited flight time, or the optimal trajectories calculation and network planning [187].

In particular, multi-UAV environments may give rise to long-endurance missions that require uninterrupted service provisioning (performing UAV replacements) that are not achievable using a single UAV due to battery capacity constraints (at most around 20 minutes flight [188]). A UAV replacement means that a UAV that is waiting in the GCS becomes active and goes into the scenario to substitute one of the UAVs that is on service to provide its same functionality. This is only possible providing a fleet exceeding the number of UAVs that have to be active on service at the same time, i.e., there is a reasonable number of fresh UAVs for replacement.

Nevertheless, developing an appropriate replacement strategy of UAVs, is one of the critical hurdles that have not yet been properly addressed by the research community. The replacement strategy enables to optimize the cost in terms of required aerial infrastructure resources, while keeping the provided level of service. To guarantee that long term (beyond battery life) services can be deployed, some of the UAVs that are at the GCS must be able to successfully replace the UAVs that provide the actual service on the stage whenever necessary (for example, when a UAV has low battery or fails). However, the economic cost of oversizing the fleet is enormous, as these devices commonly have high prices. For this reason, it is necessary to develop a resource optimization mechanism in order to allow intelligent and autonomous UAV systems to be managed with the lowest possible number of UAVs.

Figure 6.32 illustrates a representative use case of UAVs delivering network coverage. As it can be appreciated, some UAVs provide connectivity to several end-users. A Controller entity, located in the GCS, is in charge of scheduling when UAV replacements will take place. Once the replacement procedure is started for a certain UAV, that UAV directly goes back to the GCS to change its battery 1, while another one comes back in its place 2. As soon as it has a charged battery installed, it is available again in the replacement pool for other UAVs to be changed when required. By following this methodology, an uninterrupted service may be provided. Reducing the UAV fleet while ensuring a reasonable quality of service is not a straightforward procedure (Further details about the methodology depicted in Fig. 6.32 can be found in Section 6.4.4, subsections 6.4.4.1 and 6.4.4.4).

6.4.2 Related Work and Background

Ubiquitous connectivity is one of the current challenges of 5G networks and beyond 5G [187]. UAVs have appeared as a promising solution to provide reliable and flexible wireless communication services for ground users in a wide variety of scenarios [184]. The usage of UAVs promises to provide cost-effective wireless connectivity for devices without infrastructure coverage. Concretely, UAVs are considered as flying BSs for coverage extension and capacity enhancement of the existing 5G cellular networks. In [189], authors explore the use of UAV-BSs to provide coverage during natural disasters. In this work [186], an evolved packet core (EPC) inside a UAV is introduced, to orchestrate the LTE RAN in the presence of multiple BSs. This EPC can also interoperate with commercial BSs as well as commodity user equipment. In [190], the authors provide an overview of UAV-aided networks, introducing the underlying architecture and wireless channel characteristics.

One of the most critical design challenges in multi-UAV systems is the achievement of the all-to-all communication between UAVs, which is necessary for cooperation and collaboration [191][192]. If every UAV is connected to existing network infrastructure such as a GCS, satellite network or base stations, swarm communications can be delivered via this infrastructure. This type of network scheme simplifies some problems that may be associated with UAVs ad hoc networks alternatives, like routing protocols or the distributed control of the network. However, it also brings as a consequence certain limitations such as the expensive equipment (long-range or satellite antennas) and obviously less flexibility since the deployment is fixed to an existing infrastructure. An alternative solution is the usage of FANETs. In this type of systems, UAVs have several roles, not only as functional devices to provide coverage, gathering sensor data, or video dissemination but also to be used as network relays to connect all UAVs through the UAV network itself. Commonly, only one (or a few) UAV (also known as backbone UAV) are required to be connected to the fixed infrastructure (GCS). The backbone UAV is generally equipped with two radios: *(i)* Low power radio (WiFi or Bluetooth, for instance) is used for communication between the UAVs and *(ii)* high power long-range radio to communicate with the GCS [193]. It is common to find quite a few examples of research works that use FANETs to support 5G networks [194]. For instance, [185] extends a 5G network slice for video monitoring with a FANET composed of small low-altitude UAVs with multi-access edge computing facilities to allow high-speed transmission.

Although the development of UAV networks is receiving significant attention by the research community, some challenges must be solved before their proper deployment and consolidation. One of them is their limited battery capacity since normally a UAV source power mainly depends on small batteries (we are considering in this section small rotary-wing UAVs and not big fixed-wing UAVs with fuel engines). Consequently, these SUAVs (Small UAVs) are hardware-constrained devices that can not be too heavy or carry heavy payloads. Besides, to the power consumption of the flight engines, it is essential to consider the additional energy required by onboarded computers, that may not be carrying their own external batteries and in case they were, extra weight would be added to the system. As a consequence, we find that the useful lifetime of a UAV system is undoubtedly limited by these restrictions. Different research works propose solutions to provide uninterrupted service on long endurance missions and overcome the reduced-battery challenge. For instance, [195] presents an algorithm to offer continuous structural inspection services using UAVs not only through simulation results, but also using an implementation. In [196], authors consider UAV replacement (among other possible alternatives, such as refueling [197] or recharging) to maintain total surveillance of an area perimeter. Additionally, some articles propose the automatic batteries replacement [198][199][180]. They offer a GCS capable of swapping UAV batteries without human interaction. Ground task automation not only reduces human interaction but also increases the multi-UAV system operation area, improving the coverage and enabling operation in hazardous environments. This trend makes us choose battery replacement as the preferred option in the solution proposed in this section. Battery price is considerably lower than the cost of a UAV, and the time to replace the battery is remarkably shorter than the time to recharge it. Moreover, thanks to these studies and their practical experimentation, we use these results as input for our scheduling algorithms to provide accuracy to the design of UAV replacement strategies. Diverse works attempt to solve the limited battery life problem which is inherent to current SUAVs by proposing diverse alternatives. In [200], it is considered that the UAVs land to provide service (if possible and secure operation). The work in [201] summarizes different techniques to prolong the UAV operation time from Battery dumping [202] to Photovoltaic arrays [203][204]. Some other additional techniques have been proposed like wireless charging using lasers is in [205].

The optimization field, to improve the restricted communication performance of UAV networks while using the minimum amount of physical resources, is also an actual discussion topic in state of the art. In [206], the effective use of flight-time constrained devices is investigated, maximizing the average data service to ground users following a fair resource allocation policy. The solution of the cooperative allocation problem proposed in [207] significantly improves the performance of several network parameters. In [208], the authors try to minimize the number of vehicle-mounted BSs required to guarantee wireless coverage for a group of distributed ground users. Similar work in [209] proposes a placement algorithm for vehicle-mounted BSs that maximizes the number of covered ground users using the minimum UAVs. In [210], authors investigate the UAV coverage problem and propose a multi-UAV coverage model based on energy-efficient communication. The work in [211][212] focuses on the application of a multi-layout multi-subpopulation genetic algorithm achieving significantly better performance results than the other meta-heuristic algorithms also considered to improve the coverage deployment of multi-UAV networks. An explicit definition of the minimum-energy paths between a predefined initial and final

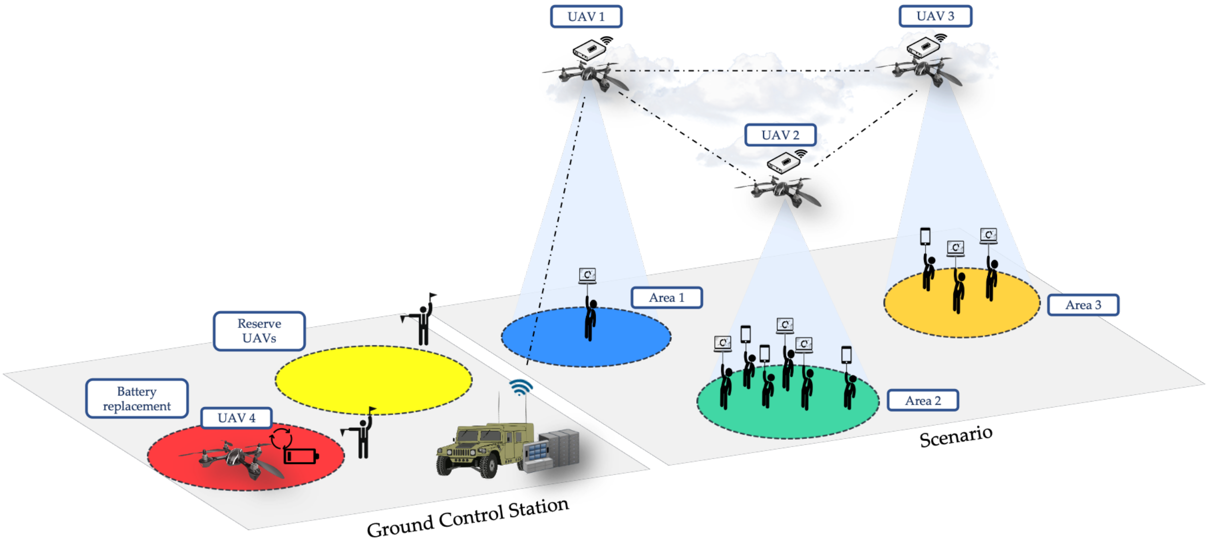


FIGURE 6.33: Multi-UAV system during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$).

configuration of a quadrotor by solving an optimal control problem concerning the angular accelerations of rotors is detailed in [213]. Their solution yielded minimum-energy and fixed-energy paths for the aerial vehicle.

6.4.3 Problem Statement

As it has just been mentioned, one of the main challenges in multi-UAV systems is to keep all the target geographic areas covered overtime by UAVs, since their battery lifetime is limited (minutes) as compared to the typical mission timelines (hours). In order to face this problem, our approach is to use a fleet with the number of UAVs that are required to cover the whole scenario, and then maintain extra UAVs in a backup pool to serve as replacement units (as it can be seen in Fig. 6.33). Once a replacement has been scheduled by the GCS, a fully recharged UAV enters the scenario while the replaced UAV goes back home to substitute its empty battery and be therefore ready to be changed by the next active UAV that requires a replacement. However, this procedure of identifying the minimum number of necessary (extra) UAVs and scheduling UAV replacements in the appropriate moment (to guarantee a minimum level of service availability) resembles a sophisticated approach and is the main problem that is treated in this section.

Figure 6.33 depicts the reference scenario considered in our analysis. In this scenario, different colours are used to represent different geographic areas, which encompass: the target areas where UAVs are intended to provide network coverage to end-users, the geographic location where UAVs are directed for a battery replacement, and the specific area where the backup pool of UAVs is kept for subsequent use. These colour patterns have been reproduced in Fig. 6.34 to classify not only what task the UAVs are doing at a certain moment but also to show which UAVs are covering the target areas at any given time. The following subsection faces the practical UAV replacement problem from an optimization viewpoint stating a simplified and manageable procedure, checking its

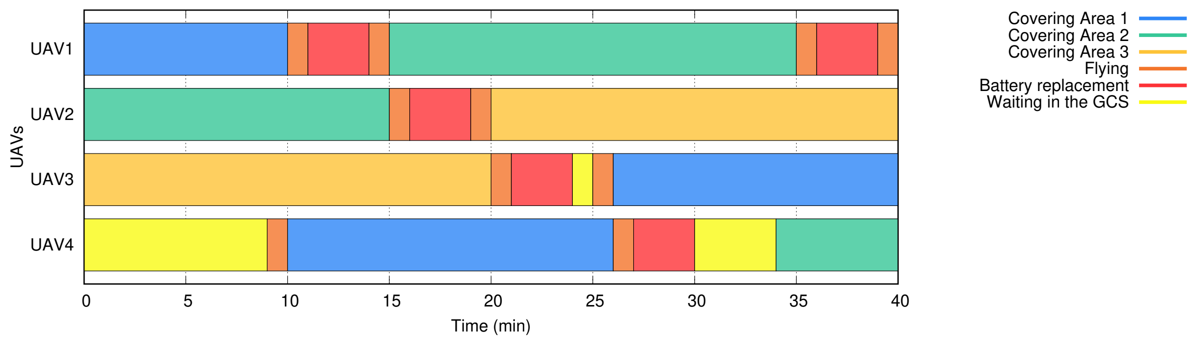


FIGURE 6.34: Multi-UAV system states during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$).

complexity, and solving it through different approaches (optimal brute force algorithm, heuristic algorithm).

6.4.3.1 Complexity Analysis

In this section, we prove that a simplified version of the proposed problem maps to an NP-hard problem (bin-packing problem [214] in this particular case) so that we are able to state its complexity. We denote a UAV using the index i and the target areas using the index j. Each UAV i has $C_{B_i}(t)$ battery level at instant t and the UAV might be in four different states: *(i)* battery replacement state (landed in the GCS), *(ii)* flying state (towards the GCS, or towards a region where it is intended to provide a network service), *(iii)* covering a region, or *(iv)* waiting in the reserve UAVs area to replace an active UAV. These four states can be appreciated in Fig. 6.34. This diagram represents a hypothetical scenario with three regions ($j = 3$) and 4 UAVs ($i = 4$), also indicating when the replacements take place to guarantee system availability over time. Note that a region/area indicates where a UAV has to fly. In the case (e.g., the number of users, high volume of traffic) two UAVs have to be geographically near, we consider two different areas. This problem statement must guarantee each region j is always covered by a UAV, i.e.:

$$\sum_i x_{i,j}(t) = 1, \quad \forall j, \forall t \tag{6.64}$$

with $x_{i,j}(t) = 1$ whenever UAV i covers region j. Moreover, at any time t, the battery level of UAV i has to stay above a safe threshold a_j (e.g., 20%) for each region, so as to ensure the flight back to the GCS:

$$\sum_j a_j x_{i,j}(t) \leq C_{B_i}(t) y_i(t), \quad \forall i, \ \forall t \tag{6.65}$$

here $y_i(t) = 1$ whenever UAV i is not in the GCS.

Additionally, battery levels keep on decreasing while UAV is covering a region. Otherwise, we consider its battery levels is set to 100% once it has returned to the GCS, and the operator has replaced the battery:

$$C_{B_i}(t+1) = \left(C_{B_i}(t) - c\right) y_i(t) + R_{T_d,T_r}\left(1 - y_i(t)\right), \quad \forall i,\ \forall t \tag{6.66}$$

with c the battery consumption, and R_{T_d,T_r} the average battery charge ratio during the time spent in returning to the GCS T_d, and the operator replacement task T_r. T_d remains constant in this simplified version of the problem, no matter how far a UAV i is from the region j it was covering, to the GCS. The main goal of this problem is to minimize the number of active UAVs over time:

$$\min \sum_{i,t} y_i(t) \tag{6.67}$$

This optimization problem with objective function (Eq. 6.67), and constraints (Eq. 6.64, Eq. 6.65), maps to the bin-packing problem. Notice that this simplified problem has as bins the drones and as items the areas. Battery levels $C_{B_i}(t)$ are just the bin capacities[9], and the battery threshold of each region j becomes the items' weights. Thus, constraint in Eq. 6.65 is just the bin-packing restriction that prevents exceeding bin capacities. Furthermore, constraint in Eq. 6.64 imposes that all items (our regions j) are fitted inside a bin.

Without considering Eq. 6.65, we already have an instance of the bin-packing problem. Since this makes some instances of our problem being NP-hard, our reduced problem automatically becomes NP-hard. Then, the next step is to generate a heuristic algorithm that will provide a sub-optimal solution. At the same time, it is required to develop a methodology that will enable the algorithm evaluation.

6.4.4 Methodology

This section describes the different elements depicted in Fig. 6.32 and explains the steps to be followed by the mission planner to provide uninterrupted network services. It first describes the parameters that UAVs must report to the GCS in order to serve as input for the scheduler algorithms. Then, it details the diverse assumptions taken for system modeling, that enable simulations to evaluate the preliminary proposals. Later, it presents the metrics to assess the performance of the proposed solutions. Finally, it describes the different strategies used in this section to schedule UAVs replacements.

[9] Note that having time-dependent variables correspond to have t repeated such variable multiple times, i.e., with $t = \{1, 2\}$, $y_i(t)$ is expressed as two different variables $y_{i,1}$ and $y_{i,2}$.

6.4.4.1 Reported Parameters

Current UAV systems regularly report to their control station their location (GPS coordinates if the UAV incorporates this type of navigation) and the remaining battery. However, this knowledge may not be enough to have a holistic view of the UAV network which enables the scheduler algorithm to satisfy the objective function (Eq. 6.67) of minimizing the number of required UAVs to provide guaranteed service availability. This is the list of parameters periodically reported by the UAVs to the GCS that enable the calculation of essential inputs for the scheduler algorithms that will be defined later to make the appropriate replacement scheduling:

- *GPS coordinates*: Longitude, latitude, and altitude enable the calculation of the distance between each UAV and the GCS. Consequently, taking into account the cruising speed of the UAV, it is possible to estimate the time, and also the required battery, needed to complete the replacement procedure.
- *Remaining battery*: The current value of the available battery, in combination with the historical battery values (last n values), allows calculating the average energy consumption. With these values, an approximation of the UAVs' lifetime can be determined.
- *Network neighbors*: The neighboring nodes enable us to generate a graph that represents the UAV network. With the GPS position and the theoretical wireless range, an overview of the network topology can be obtained. However, in certain circumstances, such as several packet collisions or high interferences, having a UAV nearby does not guarantee to have a proper communication channel established. Assuming that communications are bidirectional, for a network link to exist between two UAVs, both have to report each other to the controller, i.e., UAV_A reports UAV_B and UAV_B reports UAV_A. This functionality is deployed in the UAV payload equipment.
- *Number of connected users*: If a UAV acts as a BS or AP (it can also act as relay, video transmitter, telemetry/sensor transmitter), it must report the number of users that it is serving. This way, it can be better determined the impact that a failure (disconnection) of this particular UAV causes in the network.

6.4.4.2 Assumptions

Using a discrete time model and in order to provide a reasonable implementation of the UAV replacement strategies, it is required to make some assumptions (simplifications):

1. The UAVs in the fleet are all the same model. It also implies that all batteries have the same dimensions and therefore have the same capacity/duration. This approach is reasonable since handling UAVs that use the same battery model reduces the number of these in the GCS and also simplifies the battery exchange procedure.

2. As long as there is a topological path existing between two nodes, it is assumed that the network route is possible and it is configured, i.e., no time is needed to configure different routes when topology modifications happen. Unquestionably, the routing protocol used in the network may eventually affect the system but under normal circumstances, the convergence time is negligible [171].

3. The chosen path between two network nodes (UAVs or network users) is the shortest path based on the number of hops. A priori, this decision makes sense since taking the shortest path minimizes the delay (and actually reducing the delay is one of the main objectives of 5G). However, if the network has several users distributed heterogeneously, using different paths may be interesting to balance the network load and avoid packet collisions.

4. When a UAV is in flight, any non-flight related energy consumption (for instance due to wireless transmissions) is negligible [174]. Furthermore, UAVs usually incorporate two batteries [215]. The primary battery is in charge of supplying the flying engines while the secondary supplies the payload equipment. The secondary battery enables a static mode of operation and, in particular situations, UAVs may land to extend the life of the provided network service, stopping the flying engines while keeping the payload powered with its own battery [200]. Our laboratory experiments with the secondary battery (3.7 V and 3,800 mAh) result in more than 2 hours of duration, so the battery limiting the UAV operation is the primary one in any case. Moreover, the battery consumption model is linear and the same in all the UAVs (it does not depend on the flight conditions).

5. Because the price of batteries is remarkably lower than the cost of UAVs, it is assumed that the number of available batteries cells is huge. This way, the GCS never runs out of charged batteries. Batteries can also be recharged during the mission, and some of them could be reused.

6. UAV payloads have enough computing capacity, consequently not saturated under any conditions. In a previous work [215], experiments using Raspberry Pi 3B single board computers are carried out to prove their correct functioning.

Although all these assumptions may affect the results of the simulations, the primary purpose of this energy management approach in the UAV domain is not to achieve accurate results but to verify that it is worth using a replacement scheduler algorithm to manage moderately large UAV fleets. Once this hypothesis is demonstrated, the progressive replacement of each simplification opens interesting future work for the evaluation of more realistic results.

6.4.4.3 Performance metric

The average number of users connected (over time) is used as the metric to evaluate the performance of UAV replacement strategies. Each sampling period (e.g., 5 seconds in our simulations), this metric is examined in order to calculate the percentage of end users connected to the GCS, which is in charge of providing Internet connectivity (i.e.

the number of end users that through a path established across the UAVs network are actually connected to the GCS). The average value of all the partial results during the simulation time will be used as performance metric.

6.4.4.4 Scheduler algorithm proposals

The following subsection outlines the strategies that have been taken into account when performing the simulations. Obtaining the optimal solution and defining a heuristic algorithm is part of the optimization process. The optimal solution will not predictably serve for large scenarios (in a reasonable time), but it will validate the heuristic algorithm in small scenarios for its future application in real environments. A summary of the parameters that describe the proposed UAV replacement strategies is shown in Table 6.18.

- *Optimal algorithm:* To find the optimal UAV scheduling strategy that minimizes the number of UAVs used to cover a certain analysis region and a given number of users, a brute-force algorithm has been proposed. This algorithm is an evolution of the strategy developed by us and presented in [16], which has incorporated positioning information, number of users, and specific parameters related to the displacement between the GCS and the regions to be covered (i.e., landing time, take off time, and cruising speed of UAVs). In this regard, the proposed algorithm can be seen as an evolution of the approach addressed in [16]. The UAV scheduling strategy is explained in Fig. 6.35, and its operation can be summarized in the following three stages.
  - *Computation of parameters for UAV allocation:* This procedure consists of calculating the lifetime of each UAV (i.e., the battery lifetime to exclusively provide the service in the designated region) and its corresponding replacement time, considering the information about the locations of the GCS and the service regions (GPS coordinates) as well as the parameters v, T_o and T_l. In this step, a priority level or ranking is also assigned to each region according to the number of users that can be affected (disconnected) directly or indirectly if the UAV allocated to that region, and acting as an AP, suffers a failure. Thus, a higher priority level corresponds to a region that, if is not covered (no UAV allocated), it produces a higher number of disconnected users directly or indirectly (AP in that location with a link or links to other locations). This information is used in the process of allocation of UAVs to each region (next step), and ensures that fewer users are affected if $i < j$ at a given time.
  - *Optimal distribution of UAVs to cover the service regions:* Through a brute-force analysis, all possible combinations of available UAVs to cover the different service regions are explored. The best distribution (combination) of UAVs, which consists of those whose characteristics (battery duration) allow for the highest service availability time, is systematically selected.

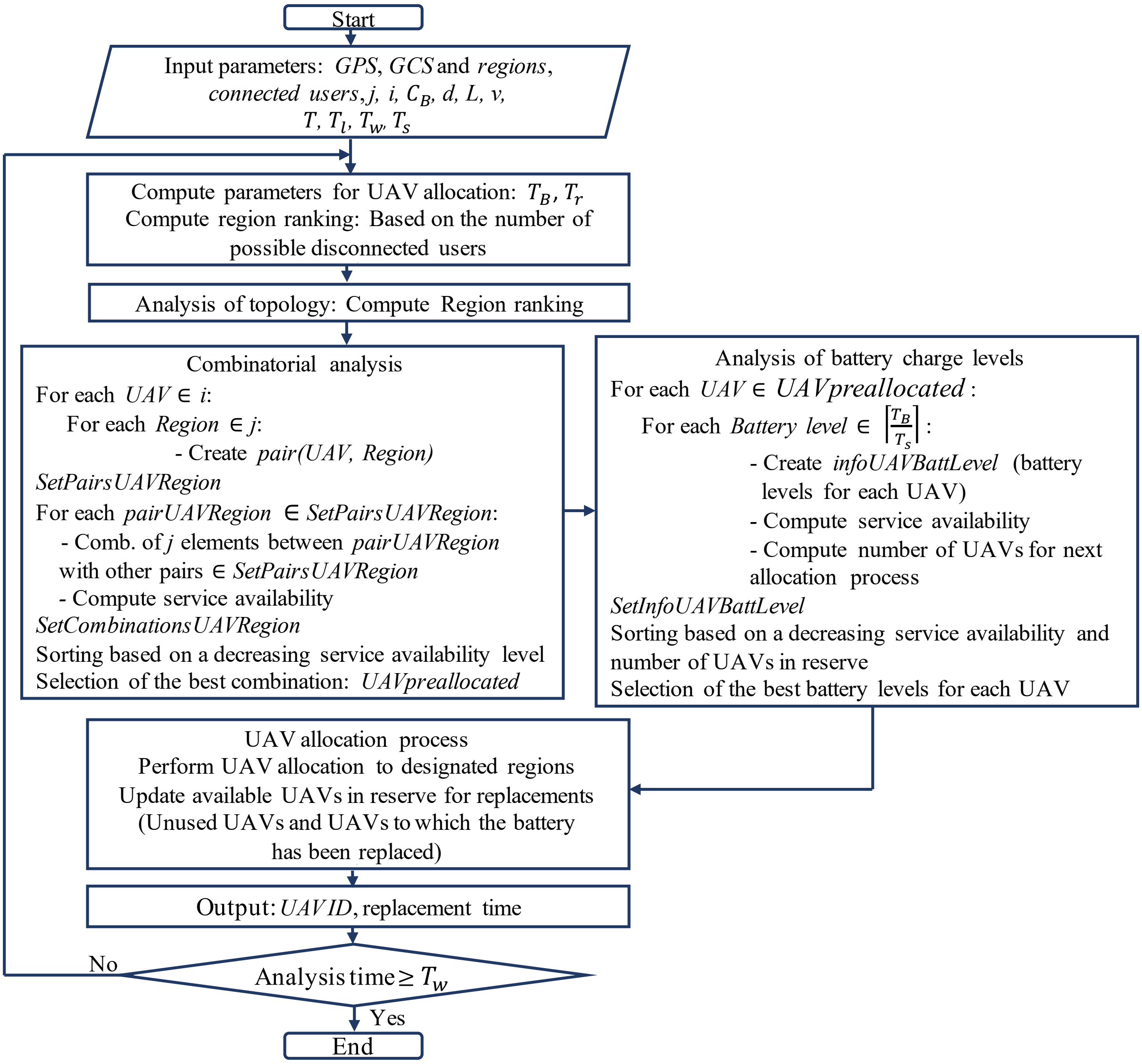
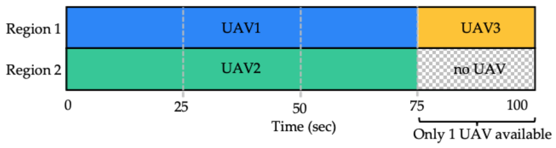
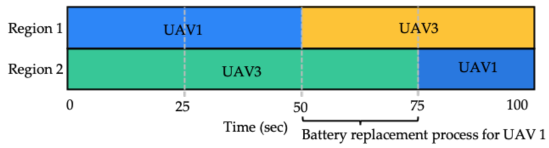


FIGURE 6.35: Optimal UAV battery replacement strategy.

– *Analysis of the percentage of battery charge to perform the replacement:* With the information from the previous step (UAV allocation per service region), the algorithm analyses the optimal charge level for each UAV in which the corresponding replacement must be performed. This procedure is carried out by means of an exhaustive exploration of each level of charge for every UAV, and seeks to guarantee the highest service availability time and an efficient use of the available resources (minimization of the number of UAVs for replacements). In a traditional approach, as shown in Fig. 6.36a, replacement is performed when the battery capacity reaches its minimum threshold ($T_B = 75$ seconds in the example). Although this procedure allows the full capacity of the battery to be used, the simultaneous discharge of several or all UAVs may cause a greater demand of resources (UAVs) for the subsequent allocations (in the worst case $i = j$) and an unavailability of one or several regions if there are no UAVs available for replacements. On the contrary, a desynchronization in the replacement time, as shown in Fig. 6.36b, allows not only a greater



(A) UAV replacement considering all battery consumption



(B) UAV replacement to offer the maximum service availability and the highest number of UAVs for the next allocation

FIGURE 6.36: Differences between the analysed scheduling procedures. Example for $j = 2$ and $i = 3$ (2 UAVs in services and 1 UAV for replacement).

availability of services but also minimization of the number of UAVs in the system. In the example presented in Fig. 6.36a, 4 UAVs are required (2 UAVs in services and 2 in the reserve) to guarantee a service availability equal to 100%, whereas in Fig. 6.36b, only 3 UAVs are necessary to reach the same availability level. Once the algorithm has determined the charge levels for replacement that allow the maximum service availability and the maximum number of UAVs available for the next allocation, these UAVs are allocated to their corresponding regions. The allocation process continues iteratively (i.e. execution of step two and step three) until reaching the maximum time horizon T_w, as shown in the example in Fig. 6.36b with $T_w = 100$ seconds.

The proposed optimal strategy is an offline exhaustive search mechanism whose complexity is given by Eq. 6.68.

$$f(i, j, T_B, T_s) = C(i \times j\,, j) + \left(\left\lceil \frac{T_B}{T_s} \right\rceil \right)^j \tag{6.68}$$

where, the first term represents the combinatorial analysis for the allocation of UAVs and the second term corresponds to the analysis of the charge levels for replacements. Both terms in Eq. 6.68 are non-polynomial, the first term is the dominant and, according to the Big-O classification [216], the order of growth of the algorithm is $O(C(ixj, j))$, i.e. non-polynomial. Based on preliminary tests we can report that if the UAVs have the same characteristics (i.e., equal battery capacity) the analysis in step three (exploration of the charge levels) is only necessary for the first allocation process, because desynchronization is maintained all other allocations, as shown in Fig. 6.38. While this mechanism can partially reduce the complexity of the algorithm, obtaining an optimal solution using an exhaustive search limits it to real-time applications and reveals the drawbacks in selecting the number of regions and UAVs (at most $i = 12$ and $j = 6$). In this regard, this strategy can be used in planning stages to estimate the number of UAVs needed for a mission, such as an emergency or rescue scenario. However, in these cases a suitable alternative is the strategy described in [16], because it is a more generic and less complex approach.

Therefore, the hardness of the problem analysed in Section 6.4.3.1 (NP-Hard) and the complexity of the optimal solution shown in Eq. 6.68 (exponential) demonstrate the need for less complex heuristic mechanisms that can be used in real-time implementations. These strategies are described in the following sections and represent the major contributions of this porposal.

- *BETA: BETweenness centrality heuristic Algorithm:* Heuristic algorithms are employed to solve optimization problems that are out of scope in reasonable times by optimal algorithms. In this particular case, it is also essential that this heuristic algorithm has a fast execution time because it must be run in real-time. BETA schedules the replacements based on the relevance of each participant within the network. To determine the relevance of an area in a network scenario, we apply graph theory fundamentals. Each area/UAV (an area is covered by an UAV) would correspond to the graph vertices (also called nodes), while the links among UAVs correspond to the graph edges (also called links or lines). One of the most well-known metrics to identify which are the most significant vertices in a graph is centrality, more specifically the betweenness centrality, which resembles the number of times a vertice acts as a connection along the shortest path between two other nodes. However, in the proposed multi UAV networks, nodes do not communicate with other nodes randomly, since they do it with those that have Internet connectivity to the public network (either the GCS or a 5G-enabled UAV), as this provides the ground users with Internet connectivity.

  To formulate this custom metric, we have divided the graph into two sub-graphs: *(i)* sub-graph which is composed by those UAVs that do not have Internet connectivity and *(ii)* sub-graph which is formed not only by the GCS, but also by the UAVs that may eventually have connectivity to the core network. Therefore, due to these specifications, the centrality metric has been calculated in the following way:

$$g'(v) = \sum_{\substack{s \neq v \\ s \in A \\ t \in B}} \left(\frac{\sigma_{st}(v)}{\sigma_{st}} U_s\right), v \in A \tag{6.69}$$

  being $\sigma_{st}(v)$ the number of shortest paths from UAV s to UAV t that traverse UAV v, and σ_{st} the total number of shortest paths from UAV s to UAV t. U_s is the number of users connected to UAV U_s. The amount of users is crucial since if there are no users connected to UAV U_s, there is no impact on the network. This statement (Eq. 6.69) (which is quite versatile) despite being designed for FANETs is also suitable for BS scenarios (where UAVs are directly connected to the core network).

  A ranking is computed using the $g'(v)$ metric as input. In case two UAVs have the same value $g'(v)$ the one closer to the GCS will be above the other in the ranking since this will minimize the total replacement procedure time, and the replaced UAV will be active sooner to perform another UAV replacement. Now it is possible to assume which scheduling strategy to follow. BETA attends the following strategy (it can be appreciated in Fig. 6.37: *(i)* if there is any topological change or the ground users move around the scenario, the algorithm must compute the ranking again; *(ii)* whenever there is a UAV available in the reserve, the algorithm schedules a replacement to the UAV with less remaining battery. However, this replacement takes place only if it does not affect UAVs that have a higher position in the ranking, i.e., that means that the remaining lifetime of the top UAVs is shorter than the time needed to make a UAV available again (after flying towards the GCS and battery replacement). In the case that this UAV replacement cannot be performed, the same analysis is repeated for the next UAV with the lower battery until the algorithm finds a UAV to make the replacement. For this algorithm to work correctly, it has to be executed periodically. In our case BETA runs every 5 seconds which coincides with the sampling period.

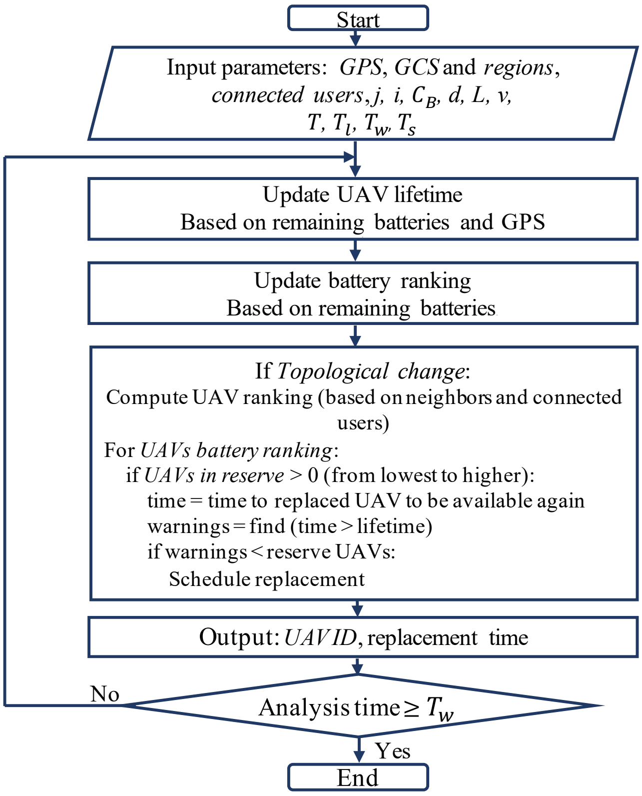


FIGURE 6.37: UAV replacement methodology BETA.

6.4.5 Simulation details and Results

In order to validate the proposed algorithms we have used different scenarios with different properties that will be discussed in this section. The following sections detail *(i)* the simulation parameters and the justification of their selection, *(ii)* the ground-truth solutions with which the BETA algorithm is also compared, *(iii)* the simulation setup, and finally *(iv)* the simulated scenarios in combination with achieved results.

6.4.5.1 Simulation parameters

This section details the parameters that have been taken into account to carry out the simulations and the selection criteria. This data, together with the related notation, can be appreciated in Table 6.18 and Table 6.19.

TABLE 6.18: System parameters in energy-aware multi-UAV deployments.

Parameter	**Notation**	**Units/Coments**
Number of regions	j	Integer number
Number of UAVs	i	Integer number
Location GCS	P_{GCS}	x,y coordinates
Location UAV$_i$	P_{UAV_i}	x,y coordinates
Number of users per region	u_j	Integer number
Total number of users	U	Integer number
Battery replacement time	T_r	Time units, *e.g*, seconds
Battery capacity	C_B	Electric current per time units, *e.g*, mAh
Device consumption	d	Electric current, *e.g*, mA
Link distance	L	Length units, *e.g*, meters
UAV cruising speed	v	Speed unitis, *e.g*, meters/seconds
Take-off time	T_o	Time units, *e.g*, seconds
Landing time	T_l	Time units, *e.g*, seconds
Simulation time	T_w	Time units, *e.g*, seconds
Sampling time	T_s	Time units, *e.g*, seconds

TABLE 6.19: Simulation parameters for algorithmic strategies in energy-aware multi-UAV deployments.

<table>
<tr><th rowspan="2">Scenario</th><th colspan="12">Parameters</th></tr>
<tr><th>j</th><th>i</th><th>U</th><th>T_r</th><th>C_B</th><th>d</th><th>L</th><th>v</th><th>T_o</th><th>T_l</th><th>T_w</th><th>T_s</th></tr>
<tr><td>I</td><td>6</td><td>6-12</td><td>300</td><td rowspan="4">180 s</td><td rowspan="4">2700 mAh</td><td rowspan="4">5670 mA</td><td rowspan="4">70 m</td><td rowspan="4">5 m/s</td><td rowspan="4">60 s</td><td rowspan="4">60 s</td><td rowspan="4">3600 s</td><td rowspan="4">5 s</td></tr>
<tr><td>II</td><td>25</td><td>25-50</td><td>250</td></tr>
<tr><td>III</td><td>25</td><td>25-50</td><td>300</td></tr>
<tr><td>IV</td><td>50</td><td>50-100</td><td>500</td></tr>
</table>

The time needed to perform a battery replacement is based on [195]. The battery capacity is based on the Parrot Bebop 2 specifications [188] (It is chosen because we have performed several tests using this model, and it is the selected unit in the technical validations we have worked previously [215][217][218], since it has demonstrated that it is able to carry a single board computer onboard like a Raspberry Pi for a reasonable time without problems and a reasonable cost). To calculate the device consumption, we have assumed that the UAV flies for 20 minutes (also specified in the technical characteristics). For WiFi range and although the standards state that the range is quite large, in practice, we have found that the WiFi range is relatively short for an acceptable received signal level [219]. The cruising speed has been calculated based on its maximum speed (also on technical specifications). Meanwhile, takeoff and landing times have also been calculated by our own measurements since we have not found accurate information. The simulations are iterated assuming a fixed number of areas to be served (and obviously one UAV per area) and then increasing the number of replacement UAVs (starting by 0 and increasing until the number of UAVs in reserve equals the number of UAVs in the scenario, which would mean doubling the size of the fleet).

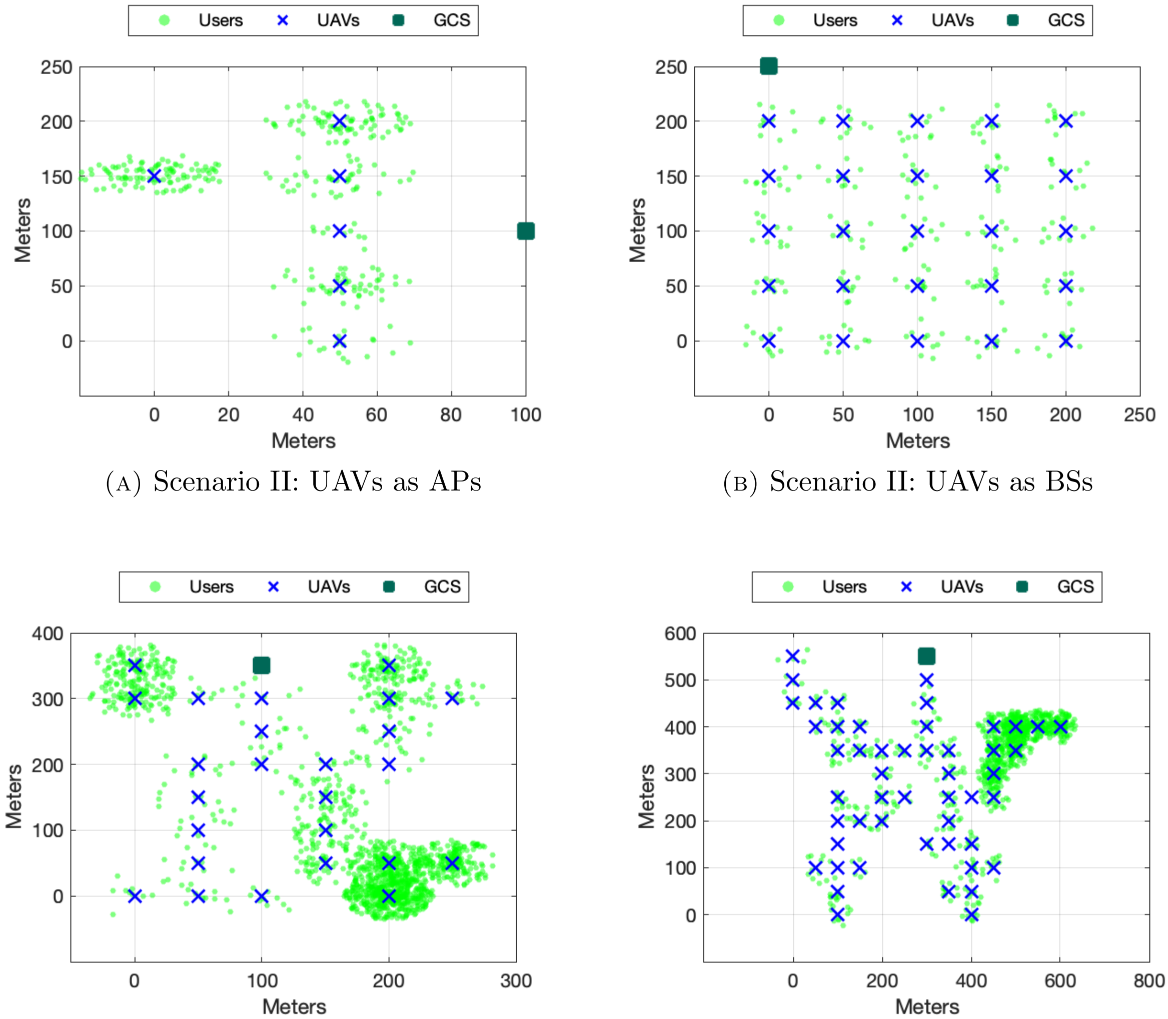
6.4.5.2 Ground-truth solutions

To provide context to the BETA and optimal algorithms performance, they will both be compared with two alternative solutions (with smaller complexity). The primary purpose of this proposal is not to measure how far the heuristic solution is from the optimal but to highlight that the use of this type of solutions is worthwhile and under which conditions and in which scenarios. In order to do that the four scheduling techniques will be compared (BETA, optimal, baseline and simple scheduling) and different conclusions will be obtained

- *Baseline:* This is the simplest strategy. UAVs are assumed to periodically send their current battery level and GPS position, however, no further calculations are made from the GCS. When an active UAV reaches a minimum battery threshold, i.e., only the required battery to return to the GCS plus a safety threshold, e.g., 20%, a replacement is scheduled (if UAVs are available), i.e., the drained UAV flights to the GCS, and at that moment (when the drained UAV starts flying to the GCS), a fresh UAV takes off and flies to the uncovered target area to provide the service. If no fresh UAVs are available, there will have no service in that area until a UAV is ready to go and cover it again. The lack of intelligence in this baseline solution prevents from reaching 100% service provisioning in any case because even with infinite UAVs to serve as fresh replacements there will always be a gap without network service corresponding to the time that passes since the drained UAV leaves the stage towards the GCS until the moment the new UAV enters the stage and starts operating.
- *Simple scheduling:* This strategy is inspired by [195]. UAVs are also assumed to send their current battery level and GPS periodically. However, in this case, the controller is required to estimate a battery threshold (based on battery reports and other parameters such as the UAV speed and the takeoff and landing times) that includes not only the minimum battery needed to return to GCS but also includes the time needed for the fresh UAV (in case there are available units) to reach the target area. That way the new UAV will start serving the area just after the old one leaves and predictably, if there are enough UAVs in reserve, all the areas can be covered for the whole mission time (or at least a high percentage of the time). In case the active UAV reaches the threshold and there is no fresh UAV to perform the replacement, the UAV can still continue providing service until it reaches the battery level needed to reach the GCS enlarging the service time.

6.4.5.3 Simulation setup

A Matlab (Matlab R2017b) event-based simulator achieves all the results. To calculate the BETA, simple scheduling, and baseline solutions in all the scenarios, a computer equipped with a 2.6 GHz Intel Core i5 processor and 8GB RAM was used. Meanwhile, the optimal algorithm has been run on a computer equipped with a 3.33 GHz x 12 cores Intel Core i7 Extreme processor and 24GB RAM. If the reader is interested in reproducing the experiment, the code is available in this repository [220].



(A) Scenario II: UAVs as APs

(B) Scenario II: UAVs as BSs

(C) Scenario III: UAVs as APs

(D) Scenario III: UAVs as BSs

FIGURE 6.38: Proposed scenarios for algorithm performance evaluation in multi-UAV deployments.

6.4.5.4 Validation scenarios

- *Scenario I: Proof of concept:* To start the analysis, we have defined a basic scenario (it can be appreciated in Fig. 6.38a) as a proof of concept. In this stage, there are a total of 6 coverage areas and 300 ground users heterogeneously distributed. In a scene with these reduced dimensions, it is possible (in terms of reasonable computation time) to run the algorithm that provides the optimal solution so it will be possible to compare all the alternatives. Fig. 6.39a depicts the average connected users for the four algorithms when the UAVs act as a FANET which means that UAVs onboard commodity WiFi equipment and use the created UAV WiFi adhoc network itself to connect to the GCS (which in turn provides the Internet connectivity). Therefore if one of the UAVs that is geographically closer to the GCS (and hence connecting part of the topology to the GCS) runs out of battery and there is no possible UAV replacement, some parts of the network may get disconnected even though the rest of UAVs may be successfully covering other target areas. Following this logic, whenever there is a failure in the backbone UAV, the system gets completely divided. On the other hand, Fig. 6.39b depicts the average connected users for the four algorithms when UAVs act as BSs (they are directly connected to the public network without the need of a hop by hop network like a FANET). These scenarios are usually employed in massified events where the existing cellular network is operating correctly, but may be insufficient. As expected, these results are better than the FANETs results since each UAV is only responsible for its own end users. However these on-boarded BS solutions are usually more expensive and it is not always viable (when the infrastructure does not exist or is temporarily damaged for instance).

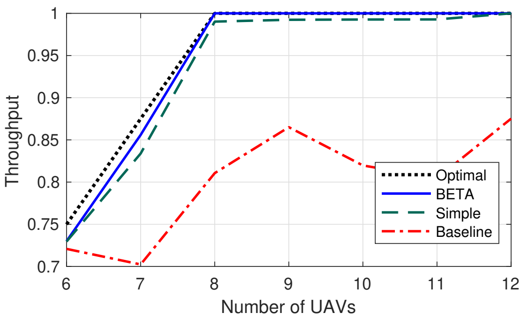
Figure 6.39a shows that both BETA and the simple scheduling strategies perform similarly and are close to the optimal solution. To reach 100% of connected users with the simple scheduling approach, it is required to double the UAV fleet (12 UAVs) but in any case in reduced scenarios, the simple scheduling solution is enough to provide an adequate service. On the other hand, the baseline algorithm provides erratic and unintuitive results considering that the performance decreases as the fleet increases. This phenomenon happens because although the time that UAVs are covering the target areas is higher, the network is disconnected for longer, i.e., having more UAVs does not guarantee overall connectivity if the backbone UAV is not working. If there are no reserve UAVs in reserve (the fleet size is equal to the number of target areas), the return and battery replacement process (of all the UAVs in the scenario) is almost synchronized (and operate simultaneously). However, if there are some UAVs in reserve, this process may be unsynchronized. For this reason, the baseline results decrease and, in consequence, are worse and inconstant.

The results in Fig. 6.39b (UAVs acting as BSs) are better as we commented and again BETA and simple scheduling strategies are close to the optimal solution. The baseline solution performance improves in this case, as the size of the fleet increases. All the strategies in fact stabilize with a fleet of 8 UAVs, two in reserve (fleet 25% oversize), and both BETA and simple scheduling achieve acceptable values.

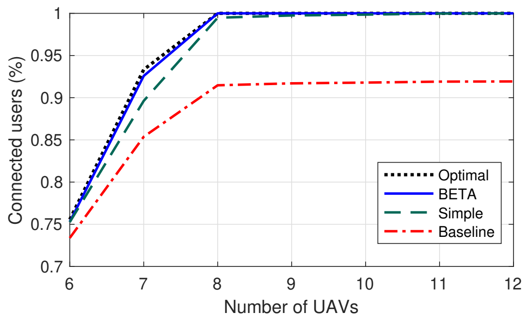
In scenarios with reduced dimensions this 25% of fleet oversize (having two UAVs in reserve) seems quite reasonable. However, in a scenario with numerous areas, e.g., 25 areas, 50 areas, this oversize may imply a rather expensive operation. It is then important to validate the solutions in much bigger scenarios and see the performance of the algorithms there.

- *Scenario II: Grid:* Figure 6.38b shows a scenario with 25 coverage areas and 250 ground users homogeneously distributed, i.e., ten ground users per area. We have selected a grid topology which is fail-tolerant since there are multiple alternative paths to reach the GCS from each area. Moreover, all the areas have the same number of users, which makes the difference in the UAV ranking insignificant in the FANET scenario and almost nonexistent in the BSs scenario.

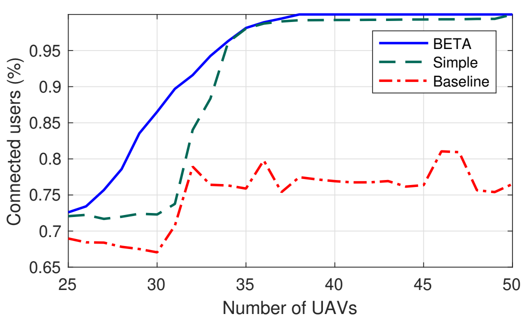
  Figure 6.39c shows that the performance of the heuristic strategy is better than the simple scheduling solution (when the fleet is formed by 30 UAVs, 5 UAVs in



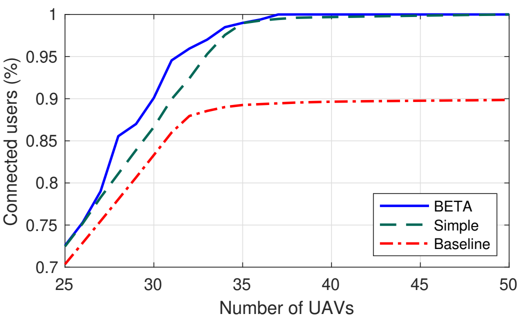
(A) Scenario I: UAVs as APs



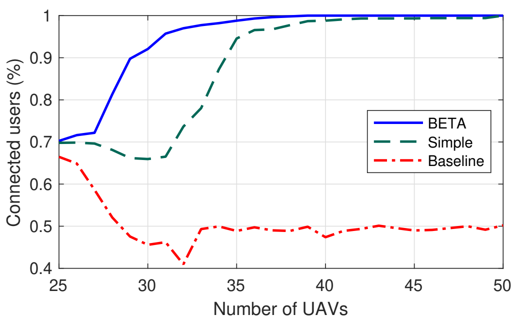
(B) Scenario I: UAVs as BSs



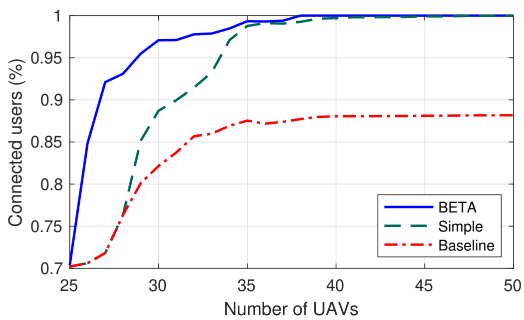
(C) Scenario II: UAVs as APs



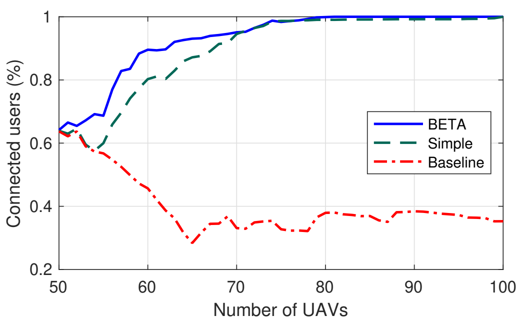
(D) Scenario II: UAVs as BSs



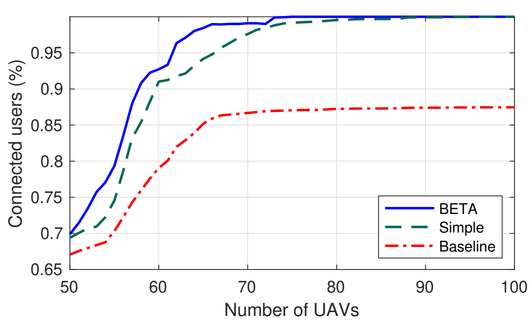
(E) Scenario III: UAVs as APs



(F) Scenario III: UAVs as BSs



(G) Scenario IV: UAVs as APs



(H) Scenario IV: UAVs as BSs

FIGURE 6.39: Average number of users connected in different scenarios increasing the fleet size.

reserve, the results improve by more than 10%). Both strategies reach acceptable levels from 35 UAVs fleet. The heuristic solution reaches 100% of users connected with a 38 UAVs fleet while the simple scheduling solution, as in the first scenario, needs to double the fleet size to reach 100% of connected users. On the other hand, the baseline solution has a similar behavior to the previous scenario. This outcome highlights that if no strategy (however simple) is used to schedule the UAV replacements, the results can be harmful, and even over-dimensioning the resources does not guarantee favorable results.

Figure 6.39d presents the results of UAVs acting as BSs. In this case, the heuristic algorithm behaves similarly to the simple scheduling solution. The heuristic algorithm schedules the UAV replacements based on $g'(v)$ metric (Eq. 6.69), which is determined using graph theory. In this scenario, the nodes representing the UAV network have the same $g'(v)$ since they are all directly connected to the infrastructure and provide connectivity to the same ground users; therefore, all UAVs connect the same number of users to the network. For this reason, scheduling the UAV replacements using the heuristic strategy has no advantage other than that they are performed as soon as there is an available fresh UAV. This phenomenon reveals that the heuristic solution makes the difference in scenarios where UAVs have different relevance within the network.

- *Scenario III and Scenario IV: Tree:* Finally, we have designed two tree-type scenarios with the users distributed very heterogeneously. This type of scheme makes some UAVs much more relevant, and scheduling replacements effectively seems to have a substantial impact on the final performance. The first scenario has 25 coverage areas and 300 ground users. The second scenario has 50 coverage areas, 500 ground users. The areas and user distribution can be appreciated in Fig. 6.38c and Fig. 6.38d.

  Figure 6.39e reveals that the difference between the BETA solution and the simple scheduling solution is significant in these scenarios. For a 30 UAVs fleet (5 UAVs in reserve), we achieve a 20% improvement by using the heuristic scheduler, which is an important variation when providing a network service. The heuristic strategy obtains 100% of users connected from 36 UAVs. As in previous scenarios, the baseline solution produces insufficient results. It is interesting to observe that as the UAV fleet increases (which have high economic cost), the users are not connected longer.

  Similarly, when UAVs act as BSs, Fig. 6.39f, we obtain better results using the heuristic strategy. This variation is because, in this scenario, the ground users are heterogeneously distributed, and consequently, UAVs have different $g'(v)$ since they connect diverse numbers of ground users, which implies that performing the correct replacement has more impact.

  The conclusions of scenario IV are similar to the ones of scenario III, although the performance (Fig. 6.39g and Figure 6.39h) is worse because of the greater complexity of the UAV network topology and the greater failure possibility.

6.4.5.5 Comparison of the UAV replacement strategies

Once the results have been obtained (Fig. 6.39) and discussed, in this section we will perform a comparison of the results by computing: *(i)* for scenario I, the distance from the optimal solution (Opt) to the suboptimal or approximate solutions ($SubOpt$), in order to verify the quality of the results, and *(ii)* for the succeeding scenarios, the difference between the simple scheduling and baseline approaches against the Heuristic strategy. To this end, the criterion of approximation ratio (ρ) has been used (see Section 5.4.0.3) and adapted to the features of this proposal. Thus, ρ in this particular case is defined as:

$$\rho = \frac{1}{i} \sum_{i} \frac{SubOpt_i}{Opt_i} \tag{6.70}$$

where $SubOpti$ and $Opti$ are the results for all the variation of UAVs in reserve (from zero to fleet size) for the optimal and suboptimal strategies, respectively. For a better understanding of the calculation of this parameter, Eq. 6.71 presents an example for the simple scheduling approach of Scenario I when UAVs act as APs (Fig. 6.39a).

$$\rho = \frac{1}{7} \times \left(\frac{72.95}{74.99} + \frac{83.38}{87.51} + \frac{99.02}{100} + \frac{99.24}{100} + \frac{99.27}{100} + \frac{99.28}{100} + \frac{100}{100} \right) = 0.98 \tag{6.71}$$

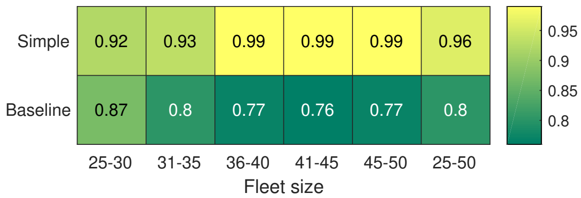
The result of Eq. 6.71 shows that the suboptimal solution (simple scheduling approach) is similar to the optimal solution in a factor equal to 0.98 (98% similarity between solutions). The rest of the ρ factors for Scenario I (Fig. 6.39a and Fig. 6.39b) are summarized in Fig. 6.40, while the ρ values for other scenarios are presented in Fig. 6.41. The comparison between the optimal solution and the approximate solutions in Scenario I, based on ρ factor, reveals that all the proposed strategies produce not only near-optimal solutions, but also a stable performance (i.e., high-quality feasible solutions). In all cases, as illustrated in Fig. 6.39a and Fig. 6.39b and then corroborated in Fig. 6.40, the approximate algorithms (BETA, simple scheduling and baseline) generate solutions very close to the optimal, even with 99% similarity to the exact result (1% of error), which is achieved by the BETA approach. Then, this strategy is used as a baseline to evaluate the performance of the other strategies (simple scheduling and baseline) for Scenario II, Scenario III, and Scenario IV. In summary, the average number of connected users allows us to appreciate where one algorithm improves another, while the distance to the optimal solution computed in this section allows us to quantify this variation. To provide a higher level of detail, ρ has been analyzed by ranges depending on the fleet size (all the analysis has been carried out by increasing the number of UAVs gradually). In addition, this value has also been computed at a global level. The main reason is to analyse in which areas the solution improves and quantify it. Since from a particular value, the solutions provide similar results, computing this metric at a global level makes it difficult to recognize the areas of improvement.

It can be seen in Fig. 6.41 that the central area of improvement is in the first two ranges, i.e., from 25 to 35 UAVs. This result is positive because, as expected, if there

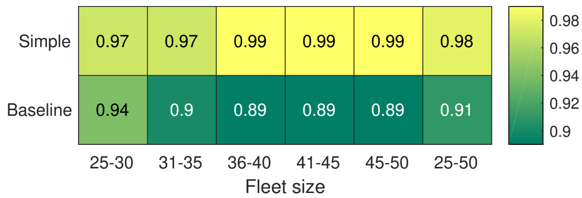
(A) Scenario I: UAVs as APs

(B) Scenario I: UAVs as BSs

FIGURE 6.40: Approximation ratio ρ: optimal strategy vs. heuristic strategies for Scenario I

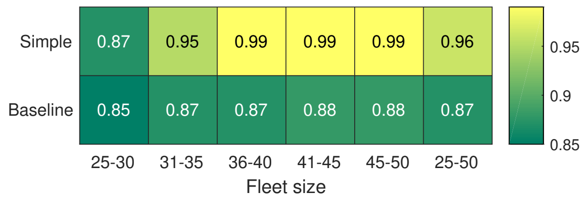


(A) Scenario II: UAVs as APs



(B) Scenario II: UAVs as BSs

(C) Scenario III: UAVs as APs



(D) Scenario III: UAVs as BSs

(E) Scenario IV: UAVs as APs

(F) Scenario IV: UAVs as BSs

FIGURE 6.41: Approximation ratio ρ: BETA vs. other suboptimal strategies.

is a reasonable amount of UAVs (with their corresponding cost), a typical solution can perform adequately. However, in scenarios with limited resources using the heuristic strategy improves in all cases the simple scheduling solution.

Analysing the above metrics, we can conclude that using strategies to make replacements is worthwhile. However, the heuristic strategy designed in this proposal is considerably aggressive since it schedules a replacement whenever UAVs are available. This strategy can result in the number of replacements skyrocketing over time, as well as the number of batteries to be used, which would bring a high economic cost. It should be noted that the price of a battery is much lower than that of a UAV but in any case it is not negligible.

Figure 6.42 displays the number of UAV replacements as a function of the number of UAVs forming the fleet and the mission time for scenario III using the BETA strategy.

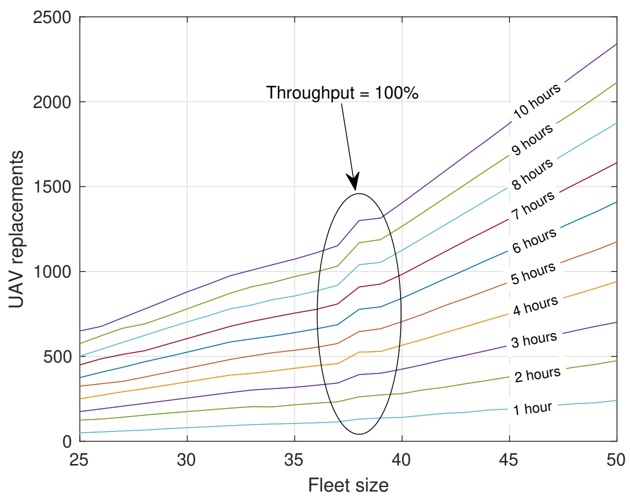


FIGURE 6.42: Number of UAV replacements using BETA in scenario III.

Approximately 1200 replacements are needed to provide a service of 10 hours and obtain 100% of users connected. Furthermore, due to the aggressive nature of BETA, the replacement grows exponentially after reaching a fleet size that guarantees 100% of connected users. In addition, it should be mentioned that the main advantage of small UAVs is that deployments are generally done very quickly and very flexibly, but as we have seen it is both economically and logistically difficult, to achieve reasonable solutions when service time largely exceeds battery lifetime. Other alternatives should be used in these cases like bigger UAVs with more battery capacity (or even using fuel), land the UAVs on the ground to improve their autonomy, or even deploying of a fixed infrastructure if service is expected to be maintained for a long.

6.4.6 Conclusions and Future work

This section states the practical UAV replacement problem, where a multi-UAV network is expected to provide long-endurance network services (in the order of hours) using constrained devices with limited autonomy (in the order of minutes). It is verified that the optimal UAV scheduling to minimize the number of UAVs for replacements while providing a guaranteed service availability, is NP-hard and that its optimal solution has exponential complexity. In this regard, some heuristics approaches have been analysed and evaluated.

Secondly, this section details a methodology, including the simulation environment and the parametrization required to perform a preliminary evaluation of these heuristic strategies. The simulator code is available in [220] to reproduce the experiment and evaluate upcoming future strategies.

The section also introduces the BETA (Betweenness centrality algorithm), a heuristic replacement strategy that performs the replacements as soon as possible based on the relevance of each UAV within the network. BETA is presented as an example in order to verify if it is worthwhile using a heuristic replacement approach or not. BETA is capable of running in real-time with a 99% similarity with the optimal solution in some simple scenarios (scenario I). In heterogeneous scenarios, BETA improves the basic solutions, achieving the most significant improvement in instances where the scenarios are heterogeneous, and the resources are limited. Furthermore, we conclude that it is far better to have a replacement strategy (no matter how simple it is) than having no strategy at all. BETA is compared with the optimal algorithm in order to evaluate the distance whenever possible and with other alternatives in some other scenarios and it has been possible to see that in some situations the advantages are not so relevant as in other ones.

This section opens several lines of future research, such as to be able to provide priority of replacements for UAVs serving users in emergency/disaster scenarios. The application of replacement strategies in disaster scenarios includes an uncertainty degree caused by several factors caused by moving UAVs while they are operating (that may change the topology), or extreme conditions that may force the engines and battery consumption. Some other futures lines include the combination of FANETs and BSs in the same scenario, to test UAVs models with different battery capacity, to model the energy consumption according to more realistic consumption patterns based on experimentation, or to schedule UAV replacements without making them through the GCS.

6.5 Application for Optimal Spectrum Allocation in Flex-Grid Optical Networks

This section presents an NFV-Enabled optimal spectrum allocation solution applicable to fixed-grid and flex-grid wavelength division multiplexing (WDM) optical networks. By exploiting the frequency shifting capability of individual lightpaths, the proposed approach aims to recalculate the initial spectrum allocation to eliminate the unusable gaps between occupied frequency slots. The available spectrum can then be used for future connections, which enables the maximization of spectrum utilization. In this regard, this section describes the NFV-enabled scheme for carrying out the optimal spectrum allocation. Then, the optimization problem is formulated as an ILP model. The problem is solved optimally using an exact brute-force search-based algorithmic strategy (OptFs). Given the NP-hard nature of the optimization problem and the non-polynomial complexity of OptFs, a suboptimal, faster heuristic strategy (FastFs), based on a pre-partition method, is proposed. Simulation results validate the performance of the proposed optimal spectrum allocation approach, and the exact and heuristic solutions, compared to the non-application of strategies, demonstrate improvements in spectrum utilization and better spectral efficiency while offering a dynamic network operation and higher data rates.

6.5.1 Introduction

The exponential traffic growth in optical communications has triggered the evolution from fixed-grid WDM networks with channels of fixed bandwidth to flex-grid systems with channels of variable size that can adapt the bandwidth utilization to the established demands [221]. Fixed-grid optical networks need to accommodate transmissions inside fixed channels of 50 GHz. Depending on the bandwidth occupancy, this channel space can be either insufficient for processing high data rates (e.g., 400 Gbps or 1 Tbps), or it can be underused if the demands require a space smaller than (or is not an exact multiple of) the channel size. Then, these operating conditions can produce degraded network performance and a wasting spectrum (a resource that could be used for transmitting additional lightpaths) [221]. These issues are solved by flex-grid technology.

Anchored to 193.1 THz and specified in the recommendation ITU-T G.694.1 flex-grid systems addresses two main aspects: (i) finer wavelength granularity, with a channel granularity equal to 12.5 GHz and a frequency slot granularity equal to 6.25 GHz, and (ii) the ability to group adjacent frequency slots to form arbitrary sized channels ranging from 12.5 GHz to 100 GHz and wider (integer multiples of 100 GHz) [222]. The potential of flex-grid to set up the channel spacing on a link dynamically according to the specific requirements of the network enables improving spectral efficiency and utilization, and network capacity (higher data rates for traffic demands) [223]. Besides, it is found that efficient spectrum utilization is essential to achieve cost and energy efficiency [224].

Despite their advantages, flex-grid optical networks impose different challenges that emerge due to the dynamic network operation and traffic demands. For instance, constantly arrival and departures of optical connections (demanding a dynamic number of consecutive frequency slots) as the result of the operation of add/drop multiplexers and the activation/deactivation of transponders generate that the available spectrum reaches a fragmentation state. Specifically, the spectrum fragmentation refers to the generation of unusable gaps between adjacent channels that lead that the spectrum is insufficient for certain connections or simply wasted as it cannot be used. In this regard, different schemes have been proposed to defragment the optical spectrum making empty and adjacent spectrum slots available for future connection requests [225]. Most of the defragmentation approaches are focused on the configuration of the lightpaths through dynamic routing and spectrum allocation algorithms and avoid the use of the migration process, because it may result in inevitable connection disruptions. Unlike existing solutions, the proposal presented in this section by leveraging the frequency shifting capability of individual lightpaths seeks the optimal spectrum allocation that enables the maximization of transmitted channels and the optimal spectrum utilization. The proposed solution assumes that the frequency shifting is not an issue and can be solved by future optical network developments and current software-based solutions such as NFV technology.

6.5.2 Optimal Spectrum Management Proposal

6.5.2.1 Proposal Description

In this section, we address the optimal spectrum allocation issue in flex-grid networks by proposing a solution that leveraging the shifting capability on the central frequency of transmitting channels can rearrange the bandwidth of the lightpaths and eliminate the gaps between adjacent transmissions to optimize the available spectrum utilization. The proposal assumes that modifying the central frequency of channels within a finite window (i.e., towards higher of lower frequency slots) is not a constraint, and it can be envisioned as a feasible solution in future optical communications deployments. In this regard, current optical deployments such as multiple-line rate, bandwidth variable, and software-defined transponders, together with wavelength selective switches, and sophisticated technologies such as NFV and SDN can be seen as potential enablers towards this evolution. The optimal spectrum allocation in our proposal is carried out through algorithmic strategies deployed at the NFV domain (as shown in the example in Fig. 6.43) and in each node in the network. The objective of these strategies, using frequency shifting if needed, is to compute the optimal channel distribution in the frequency grid to optimize the use of available spectrum while maximizing the number of transmitted lightpaths (i.e., minimizing the rejection of services if spectrum contiguity constraint is not fulfilled).

Technically the allocation process aims to concentrate the information (i.e., the bandwidth of the channels to be transmitted) using the least amount of spectrum as possible, so that frequency slots between consecutive channels that would be wasted if they are not fit the requirements of transmitting lightpaths can be used for future communications. In summary, the proposed optimal spectrum allocation can produce improved spectrum utilization. It allows more dynamic traffic behavior able to scale the used bandwidth constantly to maximize the number of transmitted channels and can be applied to fixed and flex-grid communications. Besides, it has been demonstrated that optimal spectrum allocation leads to energy efficiency and cost reduction [226].

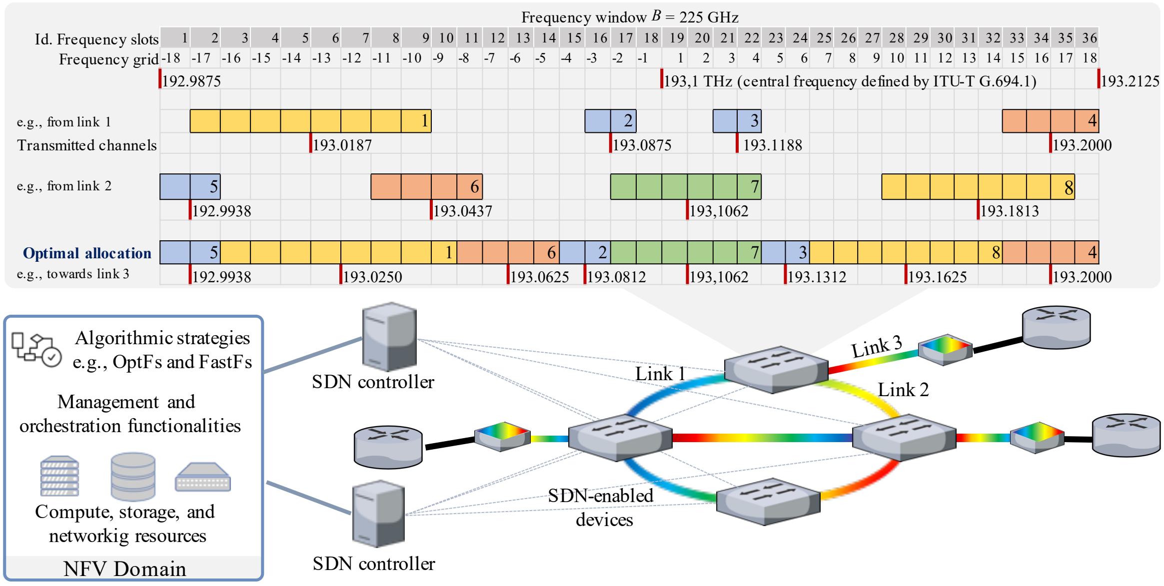


FIGURE 6.43: NFV-Enabled scheme for the optimal spectrum allocation in flex-grid optical networks.

6.5.2.2 NFV-Enabled Scheme for the Optimal Spectrum Allocation

Traditionally, the spectrum reallocation process has been avoided due to possible connection disruptions and the complexity that can be involved. In our proposal, we assume that the disruption is not an issue and can be solved through current technologies and software-based management schemes such as NFV and SDN. Regarding the complexity, the optimal spectrum allocation considering frequency shifting capabilities falls in the general category of resource-constrained knapsack problems, which have been proven to be a computationally demanding mechanism [110]. In this context, the requirements for sophisticated computational resources (e.g., memory and processing power) needed for the execution of spectrum allocation algorithmic strategies, especially for services that demand very-low end-to-end delay, is met by NFV technology. The NFV domain in the proposal offers software-based reconfigurable behavior, the provision of on-demand resources depending on the application scenario, and the management and orchestration entities to coordinate all actions/instructions between all components in the network. NFV, complemented with SDN, can offer a suitable environment for deploying sophisticated management schemes [9], in this case, focused on the optimal spectrum allocation. Besides, the existing literature has demonstrated that SDN is compatible with flex-gird technology [227], and it is a feasible solution that can be deployed on network devices (SDN-enabled) and telecommunications infrastructures (e.g., data centers or point of presence), where the core optical equipment reside.

Fig. 6.43 shows the schematic representation of the proposed ecosystem for managing optimal spectrum allocation. The example in Fig. 6.43 for a specific node shows that the proposed solution leads to the optimal spectrum utilization and the maximization in service processing. Services that under normal conditions may be rejected due to the lack of available spectrum continuity and the overlapping with other existing channels with our proposal can be potentially processed. The operation of the NFV ecosystem system complemented by SDN is summarized in three steps: (i) each node (SDN-enabled) informs the spectrum distribution status to NFV domain to trigger the spectrum allocation process, (ii) the algorithmic strategies (optimal or heuristics) are executed at core infrastructure, with the premise of guaranteeing the execution of as many services as possible, and (iii) the optimal spectrum allocation result is communicated to the network devices to perform the spectrum allocation to services that can be executed.

6.5.2.3 ILP Formulation of the Optimal Spectrum Allocation

In the spectrum allocation proposal each individual transmission, lightpath, or channel i is characterized by: (i) a central frequency (F_c^i), (ii) a finite bandwidth (b_i), (iii) a backward frequency shifting (F_{bw}^i), (iv) and a forward frequency shifting (F_{bw}^i). With the information of each lightpath, the aim of the optimal spectrum allocation approach is the optimal spectrum utilization, which is achieved through maximizing the number of channels that can be allocated within a finite frequency horizon B considering the possible frequency shifting of the transmitted lightpaths if needed. Mathematically, this objective is modeled as an objective function and is conditioned by several constraints, as indicated below.

$$OF : maximize \left\{ \sum_{i=1}^{N} \sum_{j \in M_i} b_{ij} x_{ij} \right\} \tag{6.72}$$

$$C1 : \sum_{i=1}^{N} \sum_{j \in M_i} b_{ij} x_{ij} \leq B \tag{6.73}$$

$$C1.1 : \sum_{j \in M_i} x_{ij} = 1, i \in \{1, \ldots, N\}, x_{ij} \in \{0, 1\}, j \in M_i \tag{6.74}$$

$$C2 : B \geq 0 \tag{6.75}$$

$$C3 : F_c^i \geq 0 \tag{6.76}$$

$$C4 : \{F_c^i - F_{bw}^i\} > 0 \tag{6.77}$$

$$C5 : F_c^i = 193.1 + a + 0.00625 \tag{6.78}$$

$$C6 : b_i = 12.5 \times b \tag{6.79}$$

$$C7 : F_c^i \geq \left\{ \frac{b_i}{2} + F_c^{<} i - 1 + \frac{b_{i-1}}{2} \right\} \tag{6.80}$$

- *Objective function:* OF describes the objective function to be maximized. The application of frequency shifting on central frequency of all N channels produce N mutually disjoint classes $M_1, \ldots, M_N$ of channels (later on, detonated as variations of channels, see Section 6.5.2.6) to be allocated within a bandwidth B (i.e., within all available wavelength slots). Each channel (variation) $j \in M_i$ has a bandwidth b_{ij}, and the problem is to maximize the number of channels that can be allocated within B, while, the use of this resource is optimized, respecting that exactly one channel from each class can be chosen. In summary, Eq. 6.72 finds the optimization of acceptance ratio.

- *Capacity constraint:* $C1$ determines the maximum capacity in the system, in which the decision variable $C1.1$ takes on value 1 if and only if channel j is chosen in class M_i.
- *Domain constraints:* Considering that the domain and frequency constraints are the same for the original channels (i) as for the variations (j), we simplify the index ij by i in these constraints if applicable. In this regard, $C2$, $C3$, and $C4$ guarantee non-zero and non-negative frequencies of available spectrum and center frequencies of channels.
- *Frequency constraints:* $C5$ ensures the nominal central frequency (in THz) allowed for flex-grid networks, in which a is a positive or negative integer including 0, and 0.00625 (in THz) corresponds to the frequency slot granularity. $C6$ in turn guarantees the channel bandwidth (in GHz) granularity, in which b is a positive integer. $C5$ and $C6$ are defined in ITU-T G.694.1 [222]. The original frequency distribution of lightpaths can admit channel overlapping. However, after the application of frequency shifting two adjacent channels should not overlap. This condition is ensured by $C7$ as long as i and $i-1$ are contiguous channels.

6.5.2.4 Complexity Analysis of the ILP

The process of allocating different channels, considering frequency shifting capabilities, to efficiently use a finite bandwidth B is analogous to the objective of the multiple-choice 1/0 Knapsack Problem of selecting exactly one item out of a set of items partitioned into classes (variations of channels due to frequency shifting application in our proposal) without overloading the knapsack. Then, the literature has proven that the complexity attached to this kind of problem is NP-Hard [110].

6.5.2.5 Performance Metrics for Spectrum Allocation Strategies

To solve the ILP model, we propose two spectrum allocation strategies that are described in Section 6.5.2.6. To evaluate the effectiveness of these strategies, four metrics are defined, which are described below.

1. *Acceptance Ratio (AR):* This metric measures the total number of processed channels with respect to N.
2. *Spectrum Utilization (U_B):* This metric measures the amount of spectrum used by the allocated or processed channels with respect to B.

3. *Number of frequency shiftings performed (Num_{Fs}):* This metric measures the total number of processed channels to which a non-zero frequency shifting has been applied.

4. *Cumulative value of frequency shiftings performed ($Total_{Fs}$):* This metric measures the total value to the frequency shifting performed by the processed channels.

6.5.2.6 Algorithmic Strategies: OptFs and FastFs

This section presents two algorithmic strategies to perform efficient spectrum allocation. These strategies work in an offline approach, produce exact/optimal and suboptimal solutions, and are denoted as OptTs and FastTs, respectively. The OptTs strategy bases its operation on an exhaustive brute-force search method that performs the combinatorial analysis for all N channels considering the application of all shifting values (backward and forward) to which the central frequency of channels can be subject. The optimal strategy has a non-polynomial growth of complexity that depends on the value of N and the maximum frequency shifting interval $[F_{bw}^i, F_{fw}^i]$ selected, as shown in Eq. 6.82. The FastTs strategy is based on a prepartitioning methodology that iteratively analyzes the subset of channels instead of the total set N (i.e., a divide-and-conquer procedure), which produces a reduction of complexity and running time while offers high-quality results compared to the optimal. The algorithmic strategies are explained in Fig. 6.44 (the blocks shaded blocks correspond exclusively to the heuristic), and the main steps carried out are summarized below.

- *Variations of channels ($VarCh$):* A variation of a channel is the product of the application of a specific discrete frequency shifting value (backward or forward) to the F_c^i of a channel i. The algorithmic strategies analyze all possible variations within the interval $[F_{bw}^i, \ldots, 0, \ldots, F_{fw}^i]$ for all N channels. Considering that, for simplicity $max\{F_{bw}^i\} = max\{F_{fw}^i\} = Fs$, the analysis of all channels within the frequency shifting interval produces a total number of variations equal to $AllVarCh$, as shown in Eq. 6.81.

$$AllVarCh = 2 \times N \times Fs + N \tag{6.81}$$

- *Prepartition phase:* The set of N channels is divided into $numPart$ partitions of equal or similar size that are processed iteratively. OptTs can be seen as a particular case of FastTs with $numPart = 1$.

- *Computation of combinations per partition:* For each partition, the algorithm performs the combinatorial analysis for all N channels considering all possible of variations of channels. The set of N different variations of channels is denoted as combinations of variations of channels ($CombCh$). The total number of combinations is equal to $AllCombCh$, as shown in Eq. 6.82 and Eq. 6.82 for $OptTs$ and $FastTs$, respectively.

$$AllCombCh = (2 \times Fs + 1)^N \tag{6.82}$$

$$AllCombCh = \sum_{i=1}^{numPart} (2 \times Fs + 1)^{lengthPart_k} \tag{6.83}$$

- *Computation of metrics and cost function:* For each $CombCh$ the strategies first compute the metrics AR and $NumFs$ to later build a total cost function $Cost_{comb}$ which is composed of the cost functions $Cost_{AR}$ and $Cost_{Numb_{Fs}}$. The $Cost_{comb}$ evaluates the impact on channel rejection and shifting within the combination. This value increases as the channel rejection increase (decrease of AR) and as the channels are shifted for being processed.

- *Optimal spectrum allocation:* The $CombCh$ with the minimum $Cost_{comb}$ represents the optimal allocation of services. This combination is obtained directly for OptFs, and it is building up with the merge of partial solution for FastTs. Finally, the algorithmic strategies compute the performance metrics in Section 6.5.2.5.

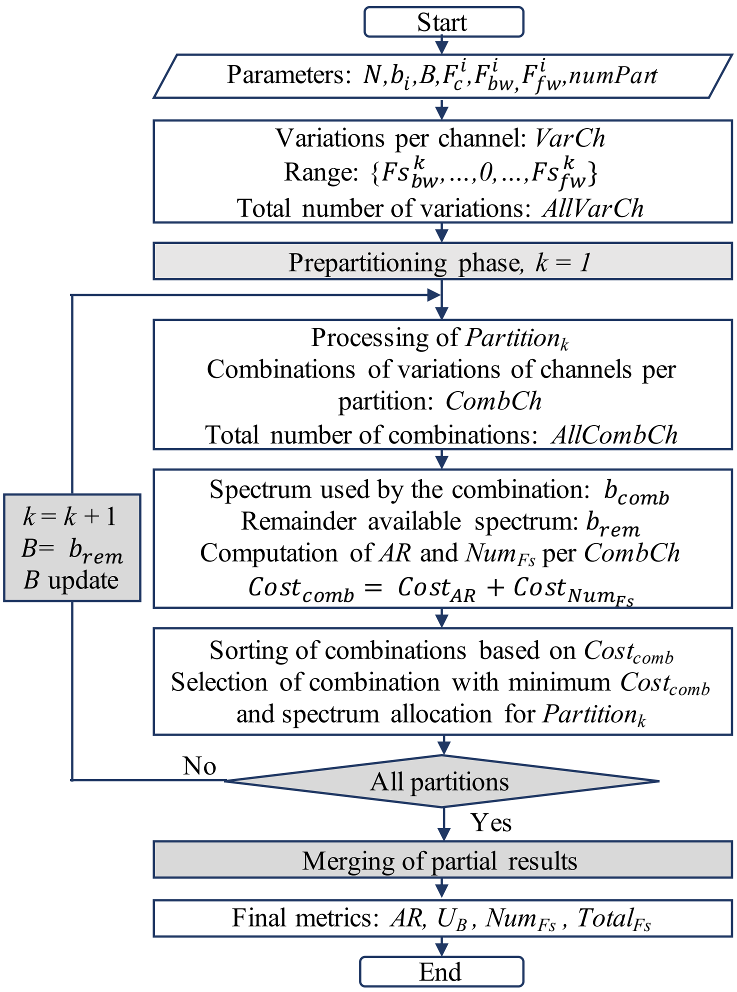


Figure 6.44: Flowchart of spectrum allocation strategies OptFs and FastFs.

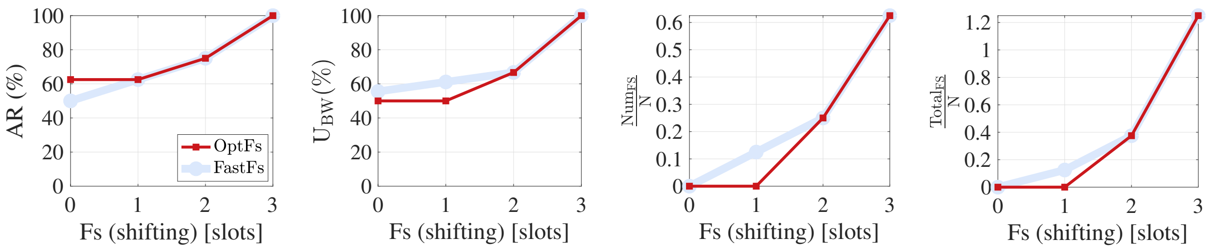
6.5.3 Evaluation

6.5.3.1 Simulation Setting and Scenarios Description

For the simulation of the spectrum allocation strategies, a discrete frequency simulator has been implemented using Matlab (R2017b) running on a computer with the features described in Section 5.4.0.1. For the evaluation of OptFs and FastFs, the analysis for a single node has been considered in two scenarios that are detailed in Table 6.20. The simulations leverage parallel processing, and up to four cores have been used for OptFs and FastTs in the scenario I and up to six cores for FastTs in scenario II. The total running time in the scenario I exceeds one hundred minutes for OptTs, and it is less than two seconds for OptTs. In scenario II, the running time for FastTs is above 4 hours, considering 50 iterations to ensure the stability of results. The simulation results are given in terms of performance metrics in Section 6.5.2.5, they are compared to the traditional scenario in which no frequency shifting is applied (i.e., $F^i_{bw} = F^i_{fw} = 0$), besides for FastTs a confidence interval of 95% has been considered, as shown in Fig. 6.45 and Fig. 6.46. If the reader is interested in reproducing the experiment, the code is available in this repository [228].

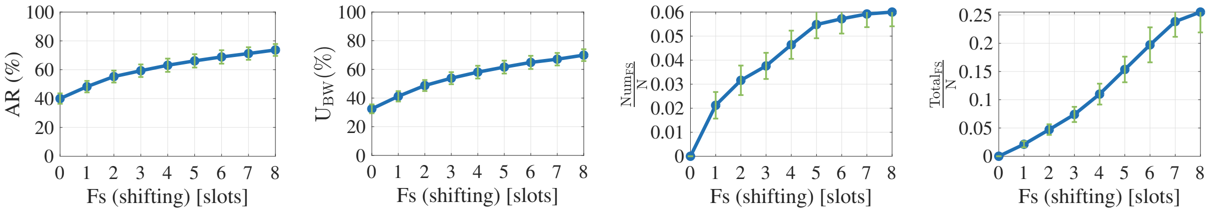
Table 6.20: Simulation parameters.

Scenario	N	b_i **[GHz]**	B **[GHz]**	F^i_c **[THz]**	$max\{F^i_{bw}\} = max\{F^i_{fw}\}$
I Example	8	As shown in Fig. 6.43.	$B = \sum_{i=1}^{N} b_i$	As shown in Fig. 6.43 and centered around 193.1 THz (ITU-T G.694.1)	3 frequency slots for all i (1 slot = 6.25 GHz, ITU-T G.694.1)
II Random Scenario	50	Random value of set {12.5, 25, 37.5, 50}	$B = \sum_{i=1}^{N} b_i$	Random value uniformly distributed within B and centered around 193.1 THz (ITU-T G.694.1)	8 frequency slots for all i (1 slot = 6.25 GHz, ITU-T G.694.1)



(a) AR OptFs and FastFs. (b) U_B OptFs and FastFs. (c) Num_{Fs} OptFs and FastFs. (d) $Total_{Fs}$ OptFs and FastFs.

Figure 6.45: Performance evaluation of spectrum allocation strategies for the proposed example. Parameters: According to scenario I in Table 6.20 and with $numPart = 2$ partitions for FastTs.



(A) AR FASTFS, $N = 50$. (B) U_B FASTFS, $N = 50$. (C) Num_{Fs} FASTFS, $N = 50$. (D) $Total_{Fs}$ FASTFS, $N = 50$..

FIGURE 6.46: Performance evaluation of FASTTS for a random scenario, $N = 50$ wavelengths. Parameters: According to scenario II in Table 6.20 and with $numPart = 12$ partitions.

6.5.3.2 Numerical Results

Simulation results demonstrate that the proposed allocation strategies lead to the maximization of AR and the optimization of U_B at the cost of an increase of the computational complexity due to incremental use of frequency shifting, as shown in Fig. 6.45 and Fig. 6.46. Results in Fig. 6.45 also validate that FASTTS produces high-quality solutions close to the optimal and with reduced complexity and related running time (FASTTS is 3100x faster than OPTFS), which enable it to be used for the analysis of scenarios with larger values of N and Fs, as shown in Fig. 6.46. Based on experimental results, we report that the algorithmic strategies have limits in terms of the accuracy of produced solutions and running time. For instance, for OPTTS values of $N > 10$ and/or $Fs > 4$ produce a huge number of combinations to be explored, as indicated in Eq. 6.82, which results in excessive demand for computational resources and running time (e.g., around of three weeks for a single simulation). On the other hand, we have verified that FASTTS has a good performance as long as the minimum length of partitions is greater or equal than 4 channels, Fs is lower or equal than 8 slots, and the maximum N is lower or equal than 50 channels. Otherwise, FASTTS produce very small increases in AR and U_B. Thus, there exists a need for developing more sophisticated heuristics for larger scenarios (e.g., thousands of lighpaths) and with very lower running time (especially for real-time application). In this regard, our proposal provides the metrics and upper bounds for performance comparison and evaluation of future developments.

6.5.4 Conclusions

The proposed solution for the optimal spectrum allocation is valid for fix-grid and flex-grid optical networks; it follows the parameters in recommendation ITU-T G. 694.1 and leverages the frequency shifting capabilities of the lightpaths to eliminate the unusable gaps between adjacent channels, which lead to the maximization of AR and optimal spectrum utilization. The proposed solution includes the ILP formulation related to the optimal spectrum allocation process, two algorithmic strategies–an exact optimal OPTFS and a heuristic FASTFS–for solving the ILP, and an NFV-enabled scheme in which the strategies have the computational resources necessary for their execution. It is verified that the optimal spectrum allocation considering frequency shifting capabilities is NP-Hard and that its optimal solution OptFs has an exponential complexity that depends on the value of N and Fs, as shown in Eq. 6.82. In this regard, FastFs covers the requirements for larger values of N and Fs, but it is conditioned to a maximum of $N = 50$ channels and $Fs = 8$ slots, and a minimum length per partition $numPart = 4$ channels. Thus, future work could be focused on developing faster and more scalable approaches based on metaheuristics or machine learning techniques and taking as a baseline the metrics and the upper bounds presented in this work. The evaluation of OptFs and FastFs allows us to verify their operation in deterministic and random scenarios. Simulation results in terms of performance metrics demonstrate the effectiveness of using frequency shifting capabilities in the spectrum allocation and utilization and the possible implications related to the increase of computational complexity.

Chapter 7

Conclusions and future work

7.1 Conclusions

The ever-growing worldwide energy demand, the CO2 emissions generated due to the production and use of energy, climate change, and the depletion of natural resources have become critical concerns that require new solutions for energy management and consumption. In order to ensure ICT and energy sustainability, measures, including the use of renewable energy sources, the deployment of adaptive energy consumption schemes, and consumer participation, are currently envisioned as feasible and de-carbonized alternatives. In this regard, this thesis proposes an adaptive energy management solution that, by leveraging the massive connectivity offered by current IoT technologies, promotes the participation of energy consumers and prioritizes the primary use of energy from green sources.

The proposed solution in this thesis starts by presenting a long-term sustainable DR architecture that, based on NFV and SDN, enables the adaptive energy consumption of IoT infrastructures. As part of the NFV/SDN architecture, we have identified the stakeholders that composed the adaptive energy provisioning and consumption ecosystem, and we have also introduced a novel consumption model conditioned on availability in which the consumers are an active part of the management process. Several management mechanisms are proposed to efficiently use the energy mainly from renewable sources, such as prioritizing the energy supply and service scheduling using time-shifting capabilities, quality service degradation, and service rejection. Also, we have analyzed the complexity related to the adaptive energy management process, and we have identified the potential enablers and operational features to present an appropriate architectural framework.

Supported by the proposed NFV/SDN architecture, we have developed the mathematical model related to adaptive energy management conditioned to availability. To do this, we have characterized the ES and the ECs based on the energy capacity and consumption parameters, respectively. Meanwhile, the modeling for adaptive energy consumption has been based on an ILP formulation whose objective is the minimization of residual power, which in the context of the proposal is the energy that in normal conditions (i.e., without the use of management mechanisms such as the use of time-shifting) is wasted by the ES if not used. To solve the ILP model, we have analyzed some algorithmic strategies, and we have decided to start with an exact or optimal method. To this end, we have developed a brute-force-based algorithmic strategy defined as defined as OpTsCost that delivers the optimal solutions in terms of performances metrics P_{RES}, AR, and P_{LACK}. With the optimal strategy, some scenarios have been evaluated whose results have been used as a baseline for later evaluating the quality of results delivered by heuristic near-optimal algorithmic strategies.

Based on the information of the energy management model and the proposed optimal solution, we have identified that the energy management problem (using management mechanisms such as the time-shifting capability on service execution) falls into the categorization of a multidimensional-multiple choice knapsack problem proven to be NP-Hard. Moreover, the evaluation of the optimal strategy has demonstrated that this algorithmic solution presents an exponential complexity that depends on the values of the entire set of services analyzed and the maximum values or intervals of the management mechanisms chosen. In this context, we have verified that the use of optimal algorithmic solutions is constrained to scenarios with a reduced number of services (e.g., up to ten services) and small values of management mechanisms (e.g., time-shifting limited up to four-time slots). To overcome this limitation and according to the literature reviewed, we have analyzed some heuristics strategies to solve the multidimensional-multiple choice knapsack problem. In the end, we have implemented and evaluated three heuristics: GreedyTs (based on a greedy approach), GATs (based on a genetic-algorithm-based solution), and DPTs (based on a dynamic programming approach).

The evaluation of the proposed heuristics strategies through extensive simulations considering different generation and consumption profiles in diverse scenarios has demonstrated that these approaches are less complex and produce high-quality solutions than the optimal algorithmic strategy. The solutions delivered from heuristics are validated based on the performance metrics results and the values of the approximation ratio taking as a baseline the optimal results. The performance of the heuristics has also demonstrated consumption of a fraction of computational capacity (in terms of CPU and RAM usage) and a reduced running time, that is among two and seven orders of magnitude faster, in comparison with the OptTsCost.

In order to scale up the applicability of heuristics strategies to IoT-enabled scenarios with thousands and hundreds of thousands of energy demands, we have incorporated a prepartitioning method to each developed algorithm (optimal and heuristics), which has resulted in eight different algorithmic strategies. The prepartitioning scheme is inspired by a divide-and-conquer approach and aims to process iteratively a subset of services instead of the all energy demands in a single step. With this feature Incorporated into the algorithmic strategies, we have verified a remarkable reduction of complexity in terms of running time and use of computation capacity. Moreover, the results of these prepartitioned versions are very similar to the obtained with the original versions (i.e., without prepartitioning), which were validated according to the performance metrics and the approximation ratio in the different simulated scenarios.

According to the simulation results, we verify that the proposed algorithmic strategies offer the best possible services scheduling, leading to efficient energy utilization and increased service processing. Moreover, when deployed into ICT infrastructures such as NFV or fog computing, these scheduling strategies can offer adaptive energy consumption in IoT-enable environments, with the capability of prioritizing the use of green energy sources.

In the last part of this thesis, we have presented different application scenarios that validate both the proposed architecture applicability and the algorithmic strategies developed. We are aware that around the field of adaptive energy management, a large number of application domains may arise due to the importance of the Internet and the efficient use of energy in today's society. However, for practicality, we have addressed only a few case studies.

In the first place, we have considered a HEMS because our proposal allows consumer participation through interaction with IoT devices and given the importance of residential sector consumption in the global energy matrix. In this scenario, we have verified that adaptive management in a HEMS allows the efficient management of the supply from the ES and the self-generated energy (e.g., from solar panels) in both offline and online approaches. Second, we have analyzed the application of adaptive management in a 5G network slicing scenario. We have exploited the energy prioritization feature described in our proposal (at network slice level) and the different management mechanisms to adapt consumption to availability. The results in this second scenario validate that our proposal effectively addresses the lack of consumption management in mobile networks and the specific context of network slicing. Subsequently, we present two scenarios in which the adaptive management concepts are applied to the environment of systems enabled with UAVs, in which the adaptive management in an online approach is carried out on the energy obtained from the batteries, guaranteeing an effective use not only of energy but also of resources (drones) to deploy FANETs. The results in these UAV environments have allowed us to verify the applicability of the proposal in environments beyond simulation because the parameters and values used in the algorithms correspond to measurements obtained from real deployments using devices (e.g., Raspberry Pi platforms and drones) and emulators. Last but not least, we have realized that the developed algorithms can be applied in a scope different to the energy; as a use case, we present a scenario in which the strategy OptTsCost and the corresponding prepartitioned version have been adapted to produce the optimal spectrum allocation in DWDM networks using flex-grid technology.

7.2 Future Work

Due to the heterogeneity of the topics presented in this thesis, this section describes future work in the different areas addressed.

7.2.1 Future Work Related to the Architecture for Adaptive Energy Consumption

This thesis presents an NFV/SDN-enabled architecture for efficient adaptive management of renewable and non-renewable energy, identifying the stakeholders involved, their interaction, and the corresponding mathematical characterization. Moreover, we have described the energy consumption model, the bi-directional communication that must be established to ensure adaptive management, the complexity of the proposal, and the technology enablers to deploy the architecture. The next step in our proposal should focus on deploying a proof of concept. In this regard, we can mention that in preliminary tests with Raspberry Pi platforms, we have verified the viability of our adaptive management solution in the context of a HEMS. A subsequent step may focus on making a prototype based on commercial NFV/SDN solutions. The different steps of the proposed algorithmic solutions can be implemented through VNFs and containers and managed by management entities such as the NFVO. We can mention that at this point, we have carried out preliminary tests, and we have identified the OSM, OpenStack, or Kubernetes projects as suitable environments for the deployment of our adaptive management strategies.

At a later stage, we hope that our proposal will motivate research in the field of the use of advanced network technologies for energy resource management and contribute to the evolution of the IoE. In this sense, future work may be focused on the actual deployment of our adaptive management solution in the facilities of energy distribution and generation companies and seen as an evolution of smart grids. At the moment, in the energy sector, there are ICT infrastructures for monitoring and management. In the process of natural evolution, the adaptation of advanced technologies such as SDN, NFV, fog computing, and IoT frameworks is imminent, which would result in an ideal environment for the deployment of the proposal presented in this thesis.

7.2.2 Future Work Related to the Energy Management Models and Algorithmic strategies

Future work can address a variety of aspects related to the energy management model or the development of more sophisticated strategies. Possible improvements for the energy model include: (i) the incorporation of a parameter that enables the variation of consumption over time, so that a service can increase or decrease consumption on availability; (ii) the possible processing of partial services, in this case, the proposed model can be based a fractionational knapsack problem; and (iii) the incorporation of a parameter that represents the possible storage of energy (i.e., the use of battery units in the model), so that in energy surplus the energy can be stored and used when needed.

Concerning future algorithmic solutions, a first step might be the development of a hybrid strategy in which, depending on the size of the scenario, the proposed algorithm selects the appropriate strategy (e.g., use of GATs for small-scenarios, DPTs for large-scale scenarios, and use of GreedyTsPart for very large-scale IoT environments). The use of machine learning techniques can be explored for adaptive energy management. For instance, supervised learning can be applied to implement a prediction method of produced energy resource or consumption, which can be used as an input to the proposed energy model and improve its performance. The clustering of consumption patterns based on unsupervised learning can be applied to the energy model to guarantee the energy supply for CS. Moreover, a complete adaptive energy management model could be established using a reinforcement learning approach, as inspired by the multi-armed bandits problem.

Bibliography

[1] International Energy Agency and Fatih Birol. *World energy outlook 2013*. International Energy Agency Paris, 2013.

[2] J.M. Carrasco, L.G. Franquelo, J.T. Bialasiewicz, E. Galvan, R.C.P. Guisado, Ma.A.M. Prats, J.I. Leon, and N. Moreno-Alfonso. Power-Electronic Systems for the Grid Integration of Renewable Energy Sources: A Survey. *Industrial Electronics, IEEE Transactions on*, 53(4):1002–1016, 2006. ISSN 0278-0046. doi: 10.1109/TIE.2006.878356. URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1667898.

[3] Alex Q Huang, Mariesa L Crow, Gerald Thomas Heydt, Jim P Zheng, and Steiner J Dale. The future renewable electric energy delivery and management (freedm) system: the energy internet. *Proceedings of the IEEE*, 99(1):133–148, 2010.

[4] Google. Google environmental report, 2018. URL https://cloud.google.com/sustainability.

[5] Greenpeace. Clicking clean: Who is winning the race to build a green internet? 2017.

[6] The New York Times. Apple becomes a green energy supplier, with itself as customer, Aug 2016. URL https://nyti.ms/33ZKqMe.

[7] Vinay Chamola and Biplab Sikdar. Solar powered cellular base stations: current scenario, issues and proposed solutions. *IEEE Communications magazine*, 54(5): 108–114, 2016.

[8] Jose Medina, Nelson Muller, and Ilya Roytelman. Demand response and distribution grid operations: Opportunities and challenges. *IEEE Transactions on Smart Grid*, 1(2):193–198, 2010. ISSN 19493053. doi: 10.1109/TSG.2010.2050156.

[9] Christian Tipantuña and Xavier Hesselbach. Nfv/sdn enabled architecture for efficient adaptive management of renewable and non-renewable energy. *IEEE Open Journal of the Communications Society*, 1:357–380, 2020.

[10] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1, 2018.

[11] Robert Basmadjian, Juan Felipe Botero, Giovanni Giuliani, Xavier Hesselbach, Sonja Klingert, and Hermann De. Making Data Centres Fit for Demand Response

: Introducing GreenSDA and GreenSLA Contracts. *IEEE Transactions on Smart Grid*, 3053(c), 2017. doi: 10.1109/TSG.2016.2632526.

[12] Bahram Shakerighadi, Amjad Anvari-Moghaddam, Juan C Vasquez, and Josep M Guerrero. Internet of things for modern energy systems: State-of-the-art, challenges, and open issues. *Energies*, 11(5):1252, 2018.

[13] Thillainathan Logenthiran, Dipti Srinivasan, and Tan Zong Shun. Demand side management in smart grid using heuristic optimization. *IEEE transactions on smart grid*, 3(3):1244–1252, 2012.

[14] Christian Tipantuña and Xavier Hesselbach. Demand-Response Power Management Strategy Using Time Shifting Capabilities. In ACM, editor, *Proceedings of the Ninth International Conference on Future Energy Systems*, pages 480–485. ACM, 2018. ISBN 9781450357678. URL https://dl.acm.org/citation.cfm?id=3213519.

[15] Thomas Strasser, Filip Andrén, Johannes Kathan, Carlo Cecati, Concettina Buccella, Pierluigi Siano, Paulo Leitao, Gulnara Zhabelova, Valeriy Vyatkin, Pavel Vrba, et al. A review of architectures and concepts for intelligence in future electric energy systems. *IEEE Transactions on Industrial Electronics*, 62(4):2424–2438, 2014.

[16] Christian Tipantuña, Xavier Hesselbach, Victor Sánchez-Aguero, Francisco Valera, Ivan Vidal, and Borja Nogales. An nfv-based energy scheduling algorithm for a 5g enabled fleet of programmable unmanned aerial vehicles. *Wireless Communications and Mobile Computing*, 2019, 2019.

[17] Victor Sanchez-Aguero, Francisco Valera, Ivan Vidal, Christian Tipantuña, and Xavier Hesselbach. Energy-aware management in multi-uav deployments: Modelling and strategies. *Sensors*, 20(10):2791, 2020.

[18] Christian Tipantuña and Xavier Hesselbach. Adaptive energy management in 5g network slicing: Requirements, architecture, and strategies. *Energies*, 13(15):3984, 2020.

[19] Christian Tipantuña and Xavier Hesselbach. Nfv-enabled efficient renewable and non-renewable energy management: Requirements and algorithms. *Future Internet*, 12(10):171, 2020.

[20] Christian Tipantuña and Xavier Hesselbach. Iot-enabled proposal for adaptive self-powered renewable energy management in home systems. *IEEE Access*, 9: 64808–64827, 2021. doi: 10.1109/ACCESS.2021.3073638.

[21] Christian Tipantuña, Xavier Hesselbach, and Walter Unger. Heuristic strategies for nfv-enabled renewable and non-renewable energy management in the future iot world. *IEEE Access*, 2021.

[22] Christian Tipantuña and Paúl Yanchapaxi. Network Functions Virtualization : An Overview and Open-Source Projects. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pages 1 – 6, Salinas, Ecuador, 2017. ISBN 9781538638941. doi: 10.1109/ETCM.2017.8247541.

[23] Christian Tipantuña, Carla Parra, Jorge Carvajal, and Ricardo Llugsi. Sdn: Layers, architecture and perspectives of a key technology for the future internet. *XII Jornadas en Ingeniería Eléctrica y Electrónica, FIEE, Escuela Politécnica Nacional, Quito, Ecuador*, 2018.

[24] Christian Tipantuña and Xavier Hesselbach. Nfv-enabled optimal spectrum allocation in flex-grid optical networks. In *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, pages 1–7, 2020. doi: 10.1109/ICTON51198.2020.9203329.

[25] Visual Networking Index, Cisco Vni, Cisco Vni, and V N I Forecast Highlights. Cisco Visual Networking Index: Forecast and Methodology, 2016–2021. pages 2016–2021, 2017.

[26] Guangyi Liu and Dajie Jiang. 5G: Vision and Requirements for Mobile Communication System towards Year 2020. *Chinese Journal of Engineering*, 2016(March), 2016.

[27] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. Next Generation 5G Wireless Networks : A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*, 18(3):1617–1655, 2016.

[28] Frank Heart, Alex McKenzie, John McQuillian, and David Walden. Arpanet completion report. *BBN Report. Bolt, Beranek and Newman Inc.(BBN). Also published in an edited version as BBN Report*, 4799:58–63, 1978.

[29] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The Road to SDN : An Intellectual History of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 44:87–98, 2014.

[30] Diego Kreutz and F Ramos. Software-Defined Networking: A Comprehensive Survey. *arXiv preprint arXiv: ...*, page 49, 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2371999. URL http://arxiv.org/abs/1406.0440.

[31] Andrew S Tanenbaum et al. Computer networks, 4-th edition. *ed: Prentice Hall*, 2003.

[32] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69, 2008. ISSN 01464833. doi: 10.1145/1355734.1355746. URL http://portal.acm.org/citation.cfm?doid=1355734.1355746.

[33] Scott Shenker. The future of networking, and the past of protocols, 2011. URL https://www.youtube.com/watch?v=YHeyuD89n1Y.

[34] Zuhran Khan Khattak, Muhammad Awais, and Adnan Iqbal. Performance evaluation of OpenDaylight SDN controller. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, 2015-April:671–676, 2014. ISSN 15219097. doi: 10.1109/PADSW.2014.7097868.

[35] Open Networking Foundation. Software-Defined Networking : The New Norm for Networks. 2012. URL https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

[36] Internet Research Task Force (IRTF). Software-Defined Networking (SDN): Layers and Architecture Terminology. Technical report, Internet Research Task Force, 2015. URL https://tools.ietf.org/html/rfc7426.

[37] Huynh Tu Dang, Han Wang, Theo Jepsen, Gordon Brebner, Changhoon Kim, Jennifer Rexford, Robert Soulé, and Hakim Weatherspoon. Whippersnapper: A p4 language benchmark suite. In *Proceedings of the Symposium on SDN Research*, pages 95–101, 2017.

[38] Zimmerman Michael, David Allan, Shirazipour, Marc Cohn, Nabil Damouny, Christos Kolias, Jeff Maguire, Serge Manning, Dave McDysan, Evelyne Roch, and Meral Shirazipour. OpenFlow-enabled SDN and Network Functions Virtualization. *Open Network Foundation*, pages 1–12, 2014.

[39] Faqir Zarrar Yousaf, Michael Bredel, Sibylle Schaller, and Fabian Schneider. Nfv and sdn-key technology enablers for 5g networks. *IEEE Journal on Selected Areas in Communications*, 2017.

[40] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review*, 43(4):3–14, 2013.

[41] Nikos Bizanis and Fernando A Kuipers. Sdn and virtualization solutions for the internet of things: A survey. *IEEE Access*, 4:5591–5606, 2016.

[42] Nils Dorsch, Fabian Kurtz, Hanno Georg, Christian Hägerling, and Christian Wietfeld. Software-defined networking for smart grid communications: Applications, challenges and advantages. In *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, pages 422–427. IEEE, 2014.

[43] ETSI. Network Functions Virtualisation,An Introduction, Benefits, Enablers, Challenges & Call for Action. *ETSI white paper*, (1):1–16, 2012. doi: 10.1.1.402.8583. URL https://portal.etsi.org/nfv/nfv{_}white{_}paper.pdf.

[44] ETSI. NFV-Architectural Framework. *Etsi*, 1:1–21, 2013. doi: DGS/NFV-0011.

[45] Diego R. López – Telefónica I+D. What Software Networks Are, 2016.

[46] ETSI. GS NFV 001 - V1.1.1 - Network Functions Virtualisation (NFV); Use Cases. 1:1–50, 2013.

[47] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual Network Embedding : A Survey. *IEEE Communications Surveys and Tutorials*, 15(4):1888–1906, 2013.

[48] Rashid Mijumbi, Joan Serrat, Juan-luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. Network Function Virtualization : State-of-the- art and Research Challenges Network Function Virtualization : State-of-the-art and Research Challenges. 18(SEPTEMBER):236–262, 2015. ISSN 1553-877X. doi: 10.1109/COMST.2015.2477041.

[49] Bob Lantz, Brandon Heller, and N McKeown. A network in a laptop: rapid prototyping for software-defined networks. *... Workshop on Hot Topics in Networks*, pages 1–6, 2010. ISSN 1450304095. doi: 10.1145/1868447.1868466. URL http://dl.acm.org/citation.cfm?id=1868466.

[50] OpenStack Foundation. OpenStack Open Source Cloud Computing Software, 2012. URL https://www.openstack.org.

[51] Diego Lopez. The Dataplane Ready Open Source NFV MANO Stack. (March), 2015. URL https://www.ietf.org/proceedings/92/slides/slides-92-nfvrg-7.pdf.

[52] ETSI. Open Source MANO (OSM), 2016. URL https://osm.etsi.org.

[53] The Linux Foundation. Open Platform for NFV (OPNFV), 2014.

[54] Robert Basmadjian, Gergo Lovasz, Michael Beck, Hermann De Meer, Xavier Hesselbach-Serra, Juan Felipe Botero, Sonja Klingert, Maria Perez Ortega, Juan Carlos Lopez, Andries Stam, Rick Van Krevelen, and Marco Di Girolamo. A generic architecture for demand response: The ALL4Green approach. *Proceedings - 2013 IEEE 3rd International Conference on Cloud and Green Computing, CGC 2013 and 2013 IEEE 3rd International Conference on Social Computing and Its Applications, SCA 2013*, pages 464–471, 2013. doi: 10.1109/CGC.2013.79.

[55] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10):770–791, 2013. ISSN 01665316. doi: 10.1016/j.peva.2013.08.014. URL http://dx.doi.org/10.1016/j.peva.2013.08.014.

[56] Linquan Zhang, Shaolei Ren, Chuan Wu, and Zongpeng Li. A Truthful Incentive Mechanism for Emergency Demand Response in Geo-Distributed Colocation Data Centers. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 1(4):1–7, 2016.

[57] G Ghatikar, Venkata Ganti, and Nance Matson. Demand Response Opportunities and Enabling Technologies for Data Centers : Findings from Field Studies. *Lawrence Berkeley National Laboratory*, (August), 2012.

[58] Glenn A Jones and Kevin J Warner. The 21st century population-energy-climate nexus. *Energy Policy*, 93:206–212, 2016.

[59] Eduardo Redondo-Iglesias, Pascal Venet, and Serge Pelissier. Efficiency degradation model of lithium-ion batteries for electric vehicles. *IEEE Transactions on Industry Applications*, 55(2):1932–1940, 2018.

[60] Tuo Shi, Siyao Cheng, Jianzhong Li, and Zhipeng Cai. Constructing connected dominating sets in battery-free networks. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[61] Gael Loubet, Alexandru Takacs, and Daniela Dragomirescu. Implementation of a battery-free wireless sensor for cyber-physical systems dedicated to structural health monitoring applications. *IEEE access*, 7:24679–24690, 2019.

[62] Biyao Huang, Xiaomin Bai, Zhenyu Zhou, Quansheng Cui, Daohua Zhu, and Ruwei Hu. Energy informatics: Fundamentals and standardization. *ICT Express*, 3(2): 76–80, 2017.

[63] LH Tsoukalas and R Gao. From smart grids to an energy internet: Assumptions, architectures and requirements. In *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pages 94–98. IEEE, 2008.

[64] Fadi AlFaris, Adel Juaidi, and Francisco Manzano-Agugliaro. Intelligent homes' technologies to optimize the energy performance for the net zero energy home. *Energy and Buildings*, 153:262–274, 2017.

[65] Nicholas Good, Keith A Ellis, and Pierluigi Mancarella. Review and classification of barriers and enablers of demand response in the smart grid. *Renewable and Sustainable Energy Reviews*, 72:57–72, 2017.

[66] Adam Cahn, Juan Hoyos, Matthew Hulse, and Eric Keller. Software-Defined Energy Communication Networks : From Substation Automation to Future Smart Grids. In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 558–563, 2013. ISBN 9781479915262.

[67] Min Wei, Seung Ho Hong, and Musharraf Alam. An iot-based energy-management platform for industrial facilities. *Applied energy*, 164:607–619, 2016.

[68] Hung Khanh Nguyen, Ju Bin Song, and Zhu Han. Demand side management to reduce peak-to-average ratio using game theory in smart grid. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 91–96. IEEE, 2012.

[69] Zhuang Zhao, Won Cheol Lee, Yoan Shin, and Kyung-Bin Song. An optimal power scheduling method for demand response in home energy management system. *IEEE transactions on smart grid*, 4(3):1391–1400, 2013.

[70] Phani Chavali, Peng Yang, and Arye Nehorai. A distributed algorithm of appliance scheduling for home energy management system. *IEEE Transactions on Smart Grid*, 5(1):282–290, 2014.

[71] Saqib Kazmi, Nadeem Javaid, Muhammad Junaid Mughal, Mariam Akbar, Syed Hassan Ahmed, and Nabil Alrajeh. Towards optimization of metaheuristic algorithms for iot enabled smart homes targeting balanced demand and supply of energy. *IEEE Access*, 7:24267–24281, 2017.

[72] Antonio J Conejo, Juan M Morales, and Luis Baringo. Real-time demand response model. *IEEE Transactions on Smart Grid*, 1(3):236–242, 2010.

[73] Bilal Hussain, Qadeer Ul Hasan, Nadeem Javaid, Mohsen Guizani, Ahmad Almogren, and Atif Alamri. An innovative heuristic algorithm for iot-enabled smart homes for developing countries. *IEEE Access*, 6:15550–15575, 2018.

[74] Yu Wang, Shiwen Mao, and R Mark Nelms. Online algorithm for optimal real-time energy distribution in the smart grid. *IEEE Transactions on Emerging Topics in Computing*, 1(1):10–21, 2013.

[75] Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J Ramos-Munoz, Javier Lorca, and Jesus Folgueira. Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges. *IEEE Communications Magazine*, 55(5):80–87, 2017.

[76] Zuo Xiang, Frank Gabriel, Elena Urbano, Giang T Nguyen, Martin Reisslein, and Frank HP Fitzek. Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working. *IEEE Journal on Selected Areas in Communications*, 37(5):1098–1116, 2019.

[77] Nicolas Huin, Andrea Tomassilli, Frédéric Giroire, and Brigitte Jaumard. Energy-efficient service function chain provisioning. *Journal of Optical Communications and Networking*, 10(3):114–124, 2018.

[78] Vincenzo Eramo, Mostafa Ammar, and Francesco Giacinto Lavacca. Migration energy aware reconfigurations of virtual network function instances in nfv architectures. *IEEE Access*, 5:4927–4938, 2017.

[79] Apichart Wantamanee, Sorawee Watarakitpaisarn, Giuseppe Carella, Chaodit Aswakul, and Thomas Magedanz. Virtualising machine to machine (m2m) application using open baton as nfv-compliant framework for building energy management system. In *2016 11th International Conference on Computer Science & Education (ICCSE)*, pages 199–204. IEEE, 2016.

[80] Xi Li, Raihana Ferdous, Carla Fabiana Chiasserini, Claudio Ettore Casetti, Francesca Moscatelli, Giada Landi, Ramon Casellas, Kei Sakaguchi, Shahzoob Bilal Chundrigar, Ricard Vilalta, et al. Novel resource and energy management for 5g integrated backhaul/fronthaul (5g-crosshaul). In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 778–784. IEEE, 2017.

[81] Ahmed N Al-Quzweeni, Ahmed Q Lawey, Taisir EH Elgorashi, and Jaafar MH Elmirghani. Optimized energy aware 5g network function virtualization. *IEEE Access*, 7:44939–44958, 2019.

[82] Robert Basmadjian, Juan Felipe Botero, Giovanni Giuliani, Xavier Hesselbach, Sonja Klingert, and Hermann De Meer. Making data centers fit for demand response: Introducing greensda and greensla contracts. *IEEE transactions on smart grid*, 9(4):3453–3464, 2016.

[83] Michael Niedermeier and Hermann De Meer. Constructing dependable smart grid networks using network functions virtualization. *Journal of Network and Systems Management*, 24(3):449–469, 2016.

[84] Helen C Leligou, Theodore Zahariadis, Lambros Sarakis, Eleftherios Tsampasis, Artemis Voulkidis, and Terpsichori E Velivassaki. Smart grid: a demanding use case for 5g technologies. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 215–220. IEEE, 2018.

[85] Delong Yang, Sachula Meng, Zhihui Wang, Sai Wu, Xuan Li, Xinzhi Cao, Xiaoyun Liu, Xue Wang, and Yuanbin Chen. E2e delay optimization for smart grids mission-critical slices in core networks. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2019.

[86] Yonghoon Choi, Muhammad Reza Kahar Aziz, Kangwook Cho, Deokjai Choi, et al. Latency-optimal network intelligence services in sdn/nfv-based energy internet cyberinfrastructure. *IEEE Access*, 8:4485–4499, 2019.

[87] Yu Wang, Shiwen Mao, and R Mark Nelms. Distributed online algorithm for optimal real-time energy distribution in the smart grid. *IEEE Internet of Things Journal*, 1(1):70–80, 2014.

[88] Cemal Keles, Asim Kaygusuz, and Baris Baykant Alagoz. Multi-source energy mixing by time rate multiple PWM for microgrids. *4th International Istanbul Smart Grid Congress and Fair, ICSG 2016*, 2016. doi: 10.1109/SGCF.2016.7492416.

[89] Mansoor Shafi, Andreas F Molisch, Peter J Smith, Thomas Haustein, Peiying Zhu, Prasan De Silva, Fredrik Tufvesson, Anass Benjebbour, and Gerhard Wunder. 5g: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE journal on selected areas in communications*, 35(6):1201–1221, 2017.

[90] ETSI Industry Specification Group NFV. Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. Technical report, ETSI, 2015. URL `https://www.etsi.org/deliver/etsi{_}gs/NFV-EVE/001{_}099/005/01.01.01{_}60/gs{_}NFV-EVE005v010101p.pdf`.

[91] S Johnson Michael, R Garey and David. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Free. Co., San Fr, 1st editio edition, 1979. ISBN 978-0716710455.

[92] 5G PPP Architecture-Working Group. View on 5g architecture. *White Paper 5GPPP, July*, 2016.

[93] OpenADR Allience. OpenADR 2.0 Profile Specification B Profile. Technical report, 2013. URL `http://www.openadr.org/specification`.

[94] Hakki C Cankaya. Software defined networking and virtualization for smart grid. *Transportation and Power Grid in Smart Cities: Communication Networks and Services*, pages 171–190, 2018.

[95] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog Computing and Its Role in the Internet of Things Characterization of Fog Computing. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–15, 2012.

[96] Richard Cziva and Dimitrios P. Pezaros. Container Network Functions: Bringing NFV to the Network Edge. *IEEE Communications Magazine*, 55(6):24–31, 2017. ISSN 01636804. doi: 10.1109/MCOM.2017.1601039.

[97] M Series. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. *Recommendation ITU M.2083–0*, 2015.

[98] Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer. A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials*, 18(1): 623–654, 2015.

[99] Shankar Lal, Tarik Taleb, and Ashutosh Dutta. Nfv: Security threats and best practices. *IEEE Communications Magazine*, 55(8):211–217, 2017.

[100] Anthony R Metke and Randy L Ekl. Security technology for smart grid networks. *IEEE Transactions on Smart Grid*, 1(1):99–107, 2010.

[101] Ming Zhu, Jiannong Cao, Zhiping Cai, Zongjian He, and Ming Xu. Providing flexible services for heterogeneous vehicles: an nfv-based approach. *IEEE network*, 30(3):64–71, 2016.

[102] Hans Kellerer, Ulrich Pferschy, and David Pisinger. Multidimensional knapsack problems. In *Knapsack problems*, pages 235–283. Springer, 2004.

[103] David Pisinger. Algorithms for knapsack problems. 1995.

[104] Pablo Pavon Marino. *Optimization of computer networks: modeling and algorithms: a hands-on approach*. John Wiley & Sons, 2016.

[105] Johann Dréo, Alain Pétrowski, Patrick Siarry, and Eric Taillard. *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media, 2006.

[106] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[107] Edward YH Lin and MC Chen. A dynamic programming approach to the multiple-choice multi-period knapsack problem and the recursive apl2 code. *Journal of Information and Optimization Sciences*, 31(2):289–303, 2010.

[108] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research*, 123(2):394–407, 2000.

[109] Dimitris Bertsimas and Ramazan Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48(4): 550–565, 2002.

[110] Hans Kellerer, Ulrich Pferschy, and David Pisinger. The multiple-choice knapsack problem. In *Knapsack Problems*, pages 317–347. Springer, 2004.

[111] Benjamin BM Shao and H Raghav Rao. A parallel hypercube algorithm for discrete resource allocation problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(1):233–242, 2005.

[112] Christian Tipantuña and Xavier Hesselbach. Heuristic strategies for nfv-enabled renewable and non-renewable energy management in the future iot world repository, Aug 2020. URL https://github.com/ChristianTipantuna/NFV-EnabledHeuristicStrategiesIoT.

[113] Emmanuel Mudaheranwa, Ye-Obong Udoakah, and Liana Cipcigan. Rwanda's energy profile and potential renewable energy resources mapping toward sustainable development goals. In *2019 IEEE PES/IAS PowerAfrica*, pages 533–538. IEEE, 2019.

[114] Yue Min Ding, Seung Ho Hong, and Xiao Hui Li. A demand response energy management scheme for industrial facilities in smart grid. *IEEE Transactions on Industrial Informatics*, 10(4):2257–2269, 2014.

[115] Onur Ayan and Belgin Turkay. Energy management algorithm for peak demand reduction. In *2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pages 1–4. IEEE, 2018.

[116] Ian Richardson, Murray Thomson, David Infield, and Conor Clifford. Domestic electricity use: A high-resolution energy demand model. *Energy and buildings*, 42 (10):1878–1887, 2010.

[117] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[118] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1, 2018.

[119] Junyon Kim. Hems (home energy management system) base on the iot smart home. *Contemporary Engineering Sciences*, 9(1):21–28, 2016.

[120] MT Iqbal. A feasibility study of a zero energy home in newfoundland. *Renewable energy*, 29(2):277–289, 2004.

[121] Kun Wang, Huining Li, Sabita Maharjan, Yan Zhang, and Song Guo. Green energy scheduling for demand side management in the smart grid. *IEEE Transactions on Green Communications and Networking*, 2(2):596–611, 2018.

[122] Petra Mesarić and Slavko Krajcar. Home demand side management integrated with electric vehicles and renewable energy sources. *Energy and Buildings*, 108:1–9, 2015.

[123] Usman Zafar, Sertac Bayhan, and Antonio Sanfilippo. Home energy management system concepts, configurations, and technologies for the smart grid. *IEEE access*, 8:119271–119286, 2020.

[124] Aravind Kailas, Valentina Cecchi, and Arindam Mukherjee. A survey of communications and networking technologies for energy management in buildings and home automation. *Journal of Computer Networks and Communications*, 2012, 2012.

[125] Young-Sung Son, Topi Pulkkinen, Kyeong-Deok Moon, and Chaekyu Kim. Home energy management system based on power line communication. *IEEE Transactions on Consumer Electronics*, 56(3):1380–1386, 2010.

[126] Habib Elkhorchani and Khaled Grayaa. Novel home energy management system using wireless communication technologies for carbon emission reduction within a smart grid. *Journal of Cleaner Production*, 135:950–962, 2016.

[127] Dusit Niyato, Lu Xiao, and Ping Wang. Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine*, 49(4):53–59, 2011.

[128] Venkatesh Reddy, Mahbub Rabbani, Mohammad T Arif, and Aman MT Oo. Iot for energy efficiency and demand management. In *2019 29th Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6. IEEE, 2019.

[129] Abdul-Rahman Al-Ali, Imran A Zualkernan, Mohammed Rashid, Ragini Gupta, and Mazin Alikarar. A smart home energy management system using iot and big data analytics approach. *IEEE Transactions on Consumer Electronics*, 63(4):426–434, 2017.

[130] Enxin Yao, Pedram Samadi, Vincent WS Wong, and Robert Schober. Residential demand side management under high penetration of rooftop photovoltaic units. *IEEE Transactions on Smart Grid*, 7(3):1597–1608, 2015.

[131] AutoSolar. Solar kit for three-phase self-consumption - 7000 w, Dec 2020. URL https://autosolar.es/kits-solares-conexion-red/kit-autoconsumo-trifasico-7000w-36000whdia-growatt.

[132] Nadeem Javaid, Ihsan Ullah, Mariam Akbar, Zafar Iqbal, Farman Ali Khan, Nabil Alrajeh, and Mohamad Souheil Alabed. An intelligent load management system with renewable energy integration for smart homes. *IEEE access*, 5:13587–13600, 2017.

[133] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61:30–40, 2016.

[134] Xinyu Fang, Ning Wang, and Thomas Aaron Gulliver. A plc channel model for home area networks. *Energies*, 11(12):3344, 2018.

[135] Md Zahurul Huq and Syed Islam. Home area network technology assessment for demand response in smart grid environment. In *2010 20th Australasian Universities Power Engineering Conference*, pages 1–6. IEEE, 2010.

[136] Tiago DP Mendes, Radu Godina, Eduardo MG Rodrigues, João CO Matias, and João PS Catalão. Smart home communication technologies and applications: Wireless protocol assessment for home area network resources. *Energies*, 8(7):7279–7311, 2015.

[137] Vinod Namboodiri, Visvakumar Aravinthan, Surya Narayan Mohapatra, Babak Karimi, and Ward Jewell. Toward a secure wireless-based home area network for metering in smart grids. *IEEE Systems Journal*, 8(2):509–520, 2013.

[138] Leehter Yao, Chien-Chi Lai, and Wei Hong Lim. Home energy management system based on photovoltaic system. In *2015 IEEE international conference on data science and data intensive systems*, pages 644–650. IEEE, 2015.

[139] John Day. *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, 2008. ISBN 978-0132252423.

[140] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. Performance evaluation of mqtt and coap via a common middleware. In *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, pages 1–6. IEEE, 2014.

[141] Eben Upton and Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[142] 3GPP. System architecture for the 5G System (5GS). Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP), 03 2020. URL https://www.3gpp.org/specifications/specification-numbering. Version 16.4.0.

[143] 3GPP. Service requirements for the 5G system. Technical Specification (TS) 22.261, 3rd Generation Partnership Project (3GPP), 03 2020. URL https://www.3gpp.org/specifications/specification-numbering. Version 17.2.0.

[144] 3GPP. Management and orchestration; Concepts, use cases and requirements. Technical Specification (TS) 28.530, 3rd Generation Partnership Project (3GPP), 12 2019. URL https://www.3gpp.org/specifications/specification-numbering. Version 16.1.0.

[145] 3GPP. Management and orchestration;Provisioning. Technical Specification (TS) 28.531, 3rd Generation Partnership Project (3GPP), 12 2019. URL https://www.3gpp.org/specifications/specification-numbering. Version 16.4.0.

[146] ETSI. Report on Network Slicing Support with ETSI NFV Architecture Framework. Technical Report (TR) GR NFV-EVE 012, European Telecommunications Standards Institute (ETSI), 12 2017. URL https://www.etsi.org/standards. Version 3.1.1.

[147] 3GPP. Study on management and orchestration of network slicing for next generation network. Technical Report (TR) 28.801, 3rd Generation Partnership Project (3GPP), 01 2018. URL https://www.3gpp.org/specifications/specification-numbering. Version 15.1.0.

[148] 3GPP. Study on tenancy concept in 5G networks and network slicing management. Technical Report (TR) 28.804, 3rd Generation Partnership Project (3GPP), 10 2019. URL https://www.3gpp.org/specifications/specification-numbering. Version 16.0.1.

[149] 3GPP. Energy efficiency of 5G. Technical Specification (TS) 28.310, 3rd Generation Partnership Project (3GPP), 03 2020. URL https://www.3gpp.org/specifications/specification-numbering. Version 1.3.0.

[150] 3GPP. Study on system and functional aspects of energy efficiency in 5G networks. Technical Report (TR) 32.972, 3rd Generation Partnership Project (3GPP), 09 2019. URL https://www.3gpp.org/specifications/specification-numbering. Version 16.1.0.

[151] ITU. Sustainable power-feeding solutions for 5G networks. Recommendation ITU-T L.1210, International Telecommunication Union (ITU), 12 2019. URL https://www.itu.int/rec/T-REC-L/en. L.1210.

[152] GSMA. *Energy Efficiency: An Overview*. GSM Association, 05 2019 (accessed March 1, 2020). https://www.gsma.com/futurenetworks/wiki/energy-efficiency-2/.

[153] ITU. Study on methods and metrics to evaluate energy efficiency for future 5G systems. Recommendation ITU-T L.1310, International Telecommunication Union (ITU), 11 2017. URL https://www.itu.int/rec/T-REC-L/en. L.1310 Supplement 36.

[154] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober. An overview of sustainable green 5g networks. *IEEE Wireless Communications*, 24(4):72–80, 2017.

[155] ITU. Standardization terms and trends in energy efficiency. Recommendation ITU-T L.1315, International Telecommunication Union (ITU), 05 2017. URL https://www.itu.int/rec/T-REC-L/en. L.1315.

[156] ITU. Assessment of mobile network energy efficiency. Recommendation ITU-T L.1331, International Telecommunication Union (ITU), 04 2017. URL https://www.itu.int/rec/T-REC-L/en. L.1331.

[157] ITU. Energy control for the software-defined networking architecture. Recommendation ITU-T L.1360, International Telecommunication Union (ITU), 12 2016. URL https://www.itu.int/rec/T-REC-L/en. L.1360.

[158] ITU. Measurement method for energy efficiency of network functions virtualization. Recommendation ITU-T L.1361, International Telecommunication Union (ITU), 11 2018. URL https://www.itu.int/rec/T-REC-L/en. L.1361.

[159] Stefano Buzzi, I Chih-Lin, Thierry E Klein, H Vincent Poor, Chenyang Yang, and Alessio Zappone. A survey of energy-efficient techniques for 5g networks and challenges ahead. *IEEE Journal on Selected Areas in Communications*, 34(4):697–709, 2016.

[160] Daniela Renga and Michela Meo. From self-sustainable green mobile networks to enhanced interaction with the smart grid. In *2018 30th International Teletraffic Congress (ITC 30)*, volume 1, pages 129–134. IEEE, 2018.

[161] Osamah Ibrahiem Abdullaziz, Marco Capitani, Claudio Ettore Casetti, Carla Fabiana Chiasserini, Shahzoob Bilal Chundrigar, Giada Landi, Xi Li, Francesca Moscatelli, Kei Sakaguchi, and Samer T Talat. Energy monitoring and management in 5g integrated fronthaul and backhaul. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–6. IEEE, 2017.

[162] Yixue Hao, Daxin Tian, Giancarlo Fortino, Jing Zhang, and Iztok Humar. Network slicing technology in a 5g wearable network. *IEEE Communications Standards Magazine*, 2(1):66–71, 2018.

[163] Gabriel da Silva, Dyego Oliveira, Rafael L Gomes, Luiz F Bittencourt, and Edmundo RM Madeira. Reliable network slices based on elastic network resource demand. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2020.

[164] Matteo Pozza, Patrick K Nicholson, Diego F Lugones, Ashwin Rao, Hannu Flinck, and Sasu Tarkoma. On reconfiguring 5g network slices. *IEEE Journal on Selected Areas in Communications*, 38(7):1542–1554, 2020.

[165] Menglan Jiang, Massimo Condoluci, and Toktam Mahmoodi. Network slicing management & prioritization in 5g mobile systems. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6. VDE, 2016.

[166] Yong Xiao and Marwan Krunz. Dynamic network slicing for scalable fog computing systems with energy harvesting. *IEEE Journal on Selected Areas in Communications*, 36(12):2640–2654, 2018.

[167] GSMA. *Generic Network Slice Template NG.116*. GSM Association, 05 2019. Version 1.0.

[168] Hazim Shakhatreh, Ahmad Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned aerial vehicles: A survey on civil applications and key research challenges. *arXiv preprint arXiv:1805.00881*, 2018.

[169] Borja Nogales, Victor Sanchez-Aguero, Ivan Vidal, Francisco Valera, and Jaime Garcia-Reinoso. A nfv system to support configurable and automated multi-uav service deployments. In *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, pages 39–44. ACM, 2018.

[170] Kyle JS White, Ewen Denney, Matt D Knudson, Angelos K Mamerides, and Dimitrios P Pezaros. A programmable sdn+ nfv-based architecture for uav telemetry monitoring. In *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, pages 522–527. IEEE, 2017.

[171] Borja Nogales, Victor Sanchez-Aguero, Ivan Vidal, and Francisco Valera. Adaptable and automated small uav deployments via virtualization. *Sensors*, 18(12):4116, 2018.

[172] Mohamed Sara, Imad Jawhar, and Mohamed Nader. A softwarization architecture for uavs and wsns as part of the cloud environment. In *Cloud Engineering Workshop (IC2EW), 2016 IEEE International Conference on*, pages 13–18. IEEE, 2016.

[173] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE Communications Surveys & Tutorials*, 18(2):1123–1152, 2016.

[174] Yong Zeng and Rui Zhang. Energy-efficient uav communication with trajectory optimization. *IEEE Trans. Wireless Commun*, 16(6):3747–3760, 2017.

[175] Fabian Kaup, Philip Gottschling, and David Hausheer. Powerpi: Measuring and modeling the power consumption of the raspberry pi. In *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*, pages 236–243. IEEE, 2014.

[176] Aaron Carroll, Gernot Heiser, et al. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14, pages 21–21. Boston, MA, 2010.

[177] Pablo Serrano, Andres Garcia-Saavedra, Giuseppe Bianchi, Albert Banchs, and Arturo Azcorra. Per-frame energy consumption in 802.11 devices and its implication on modeling and design. *IEEE/ACM Transactions on Networking (ToN)*, 23(4): 1243–1256, 2015.

[178] Steven Dunbar, Freeman Wenzl, Christopher Hack, Ronny Hafeza, Hasan Esfeer, Francois Defay, Sebastien Prothin, Damienne Bajon, and Zoya Popovic. Wireless far-field charging of a micro-uav. In *Wireless Power Transfer Conference (WPTC), 2015 IEEE*, pages 1–4. IEEE, 2015.

[179] Mingxi Wang. Systems and methods for uav battery exchange, May 24 2016. US Patent 9,346,560.

[180] Koji AO Suzuki, Paulo Kemper Filho, and James R Morrison. Automatic battery replacement system for uavs: Analysis and design. *Journal of Intelligent & Robotic Systems*, 65(1-4):563–586, 2012.

[181] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.

[182] Federal Aviation Administration. *FAA Aerospace Forecast*, 2019 (accessed March 25, 2020). https://www.faa.gov/.

[183] 5GPPP. *Key Performance Indicators*, 2019 (accessed March 25, 2020). https://5g-ppp.eu/kpis/.

[184] Samira Hayat, Evşen Yanmaz, and Raheeb Muzaffar. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, 18(4):2624–2661, 2016.

[185] Christian Grasso and Giovanni Schembra. A fleet of mec uavs to extend a 5g network slice for video monitoring with low-latency constraints. *Journal of Sensor and Actuator Networks*, 8(1):3, 2019.

[186] Mehrdad Moradi, Karthikeyan Sundaresan, Eugene Chai, Sampath Rangarajan, and Z Morley Mao. Skycore: Moving core to the edge for untethered and reliable uav-based lte networks. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 35–49, 2018.

[187] Bin Li, Zesong Fei, and Yan Zhang. Uav communications for 5g and beyond: Recent advances and future trends. *IEEE Internet of Things Journal*, 6(2):2241–2263, 2018.

[188] Parrot. *Parrot Bebop 2*, 2019 (accessed March 25, 2020). https://www.parrot.com/es/drones/parrot-bebop-2.

[189] Arvind Merwaday and Ismail Guvenc. Uav assisted heterogeneous networks for public safety communications. In *2015 IEEE wireless communications and networking conference workshops (WCNCW)*, pages 329–334. IEEE, 2015.

[190] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, 2016.

[191] Ilker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying ad-hoc networks (fanets): A survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.

[192] Jesús Sánchez-García, JM García-Campos, Mario Arzamendia, D Gutierrez Reina, SL Toral, and D Gregor. A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications. *Computer Communications*, 119:43–65, 2018.

[193] Muhammad Asghar Khan, Ijaz Mansoor Qureshi, and Fahimullah Khanzada. A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (fanet). *Drones*, 3(1):16, 2019.

[194] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE Communications Surveys & Tutorials*, 18(2):1123–1152, 2015.

[195] Milan Erdelj, Osamah Saif, Enrico Natalizio, and Isabelle Fantoni. Uavs that fly forever: Uninterrupted structural inspection through automatic uav replacement. *Ad Hoc Networks*, 94:101612, 2019.

[196] Oleg Burdakov, Jonas Kvarnström, and Patrick Doherty. Optimal scheduling for replacing perimeter guarding unmanned aerial vehicles. *Annals of Operations Research*, 249(1-2):163–174, 2017.

[197] Mario Luca Fravolini, Antonio Ficola, Giampiero Campa, Marcello Rosario Napolitano, and Brad Seanor. Modeling and control issues for autonomous aerial refueling for uavs using a probe–drogue refueling system. *Aerospace science and technology*, 8(7):611–618, 2004.

[198] Katsuya Fujii, Keita Higuchi, and Jun Rekimoto. Endless flyer: a continuous flying drone with automatic battery replacement. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, pages 216–223. IEEE, 2013.

[199] N Kemal Ure, Girish Chowdhary, Tuna Toksoz, Jonathan P How, Matthew A Vavrina, and John Vian. An automated battery management system to enable persistent missions with multiple aerial vehicles. *IEEE/ASME transactions on mechatronics*, 20(1):275–286, 2014.

[200] Victor Sanchez-Aguero, Francisco Valera, Borja Nogales, Luis F Gonzalez, and Ivan Vidal. Venue: Virtualized environment for multi-uav network emulation. *IEEE Access*, 7:154659–154671, 2019.

[201] Maxim Lu, Mehdi Bagheri, Alex P James, and Toan Phung. Wireless charging techniques for uavs: A review, reconceptualization, and extension. *IEEE Access*, 6: 29865–29884, 2018.

[202] Tan Chang and Hu Yu. Improving electric powered uavs' endurance by incorporating battery dumping concept. *Procedia Engineering*, 99:168–179, 2015.

[203] Thomas J Nugent Jr and Jordin T Kare. Laser power beaming for defense and security applications. In *Unmanned systems technology XIII*, volume 8045, page 804514. International Society for Optics and Photonics, 2011.

[204] Kate J Duncan. Laser based power transmission: Component selection and laser hazard analysis. In *2016 IEEE PELS Workshop on Emerging Technologies: Wireless Power Transfer (WoW)*, pages 100–103. IEEE, 2016.

[205] Boris Galkin, Jacek Kibilda, and Luiz A DaSilva. Uavs as mobile infrastructure: Addressing battery lifetime. *IEEE Communications Magazine*, 57(6):132–137, 2019.

[206] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization. *IEEE Transactions on Wireless Communications*, 16 (12):8052–8066, 2017.

[207] Vishal Sharma, Kathiravan Srinivasan, Han-Chieh Chao, Kai-Lung Hua, and Wen-Huang Cheng. Intelligent deployment of uavs in 5g heterogeneous communication environment for improved coverage. *Journal of Network and Computer Applications*, 85:94–105, 2017.

[208] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607, 2016.

[209] Mohamed Alzenad, Amr El-Keyi, Faraj Lagum, and Halim Yanikomeroglu. 3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters*, 6(4):434–437, 2017.

[210] Lang Ruan, Jinlong Wang, Jin Chen, Yitao Xu, Yang Yang, Han Jiang, Yuli Zhang, and Yuhua Xu. Energy-efficient multi-uav coverage deployment in uav networks: A game-theoretic framework. *China Communications*, 15(10):194–209, 2018.

[211] DG Reina, Hissam Tawfik, and SL Toral. Multi-subpopulation evolutionary algorithms for coverage deployment of uav-networks. *Ad Hoc Networks*, 68:16–32, 2018.

[212] DG Reina, Tracy Camp, Aarti Munjal, and SL Toral. Evolutionary deployment and local search-based movements of 0th responders in disaster scenarios. *Future Generation Computer Systems*, 88:61–78, 2018.

[213] Fabio Morbidi, Roel Cano, and David Lara. Minimum-energy path generation for a quadrotor uav. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1492–1498. IEEE, 2016.

[214] Silvano Martello and Daniele Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management science*, 44(3):388–399, 1998.

[215] Borja Nogales, Ivan Vidal, Victor Sanchez-Aguero, Francisco Valera, Luis F Gonzalez, and Arturo Azcorra. Automated deployment of an internet protocol telephony service on unmanned aerial vehicles using network functions virtualization. *JoVE (Journal of Visualized Experiments)*, (153):e60425, 2019.

[216] Michael R Garey and David S Johnson. A guide to the theory of np-completeness. *Computers and Intractability*, pages 37–79, 1990.

[217] Telemadrid. *La Universidad Carlos III junto al Instituto Imdea Networks desarrolla un dron antincendios*, 2020 (accessed March 25, 2020). https://bit.ly/2RsdPME.

[218] RTVE. *Zoom Net - 5GRange*, 2020 (accessed March 25, 2020). https://www.rtve.es/m/alacarta/videos/zoom-net/zoom-net-5g-dive-entrevista-shou-zi-chew-dreams/5526638/?media=tve.

[219] Victor Sanchez-Aguero, Borja Nogales, Francisco Valera, and Ivan Vidal. Investigating the deployability of voip services over wireless interconnected micro aerial vehicles. *Internet Technology Letters*, 1(5):e40, 2018.

[220] Victor Sanchez-Aguero, Christian Tipantuña, and Francisco Valera. *Energy-aware management strategies in Multi-UAV fleets repository*, 2020 (accessed March 25, 2020). https://github.com/vsaguero/energyUAV.

[221] Andrew Lord, Yu Rong Zhou, Rich Jensen, Annalisa Morea, and Marc Ruiz. Evolution from wavelength-switched to flex-grid optical networks. *Elastic Optical Networks*, pages 7–30, 2016.

[222] ITU Recommendation. Spectral grids for wdm applications: Dwdm frequency grid. *ITU-T G.694.1*, 694.1, 2006.

[223] Paul Wright, Andrew Lord, and Luis Velasco. The network capacity benefits of flexgrid. In *2013 17th International Conference on Optical Networking Design and Modeling (ONDM)*, pages 7–12. IEEE, 2013.

[224] E Palkopoulou, M Angelou, D Klonidis, K Christodoulopoulos, A Klekamp, F Buchali, E Varvarigos, and I Tomkos. Quantifying spectrum, cost, and energy efficiency in fixed-grid and flex-grid networks. *Journal of Optical Communications and Networking*, 4(11):B42–B51, 2012.

[225] Arian Ghallaj, Ronald Romero Reyes, Matthias Ermel, and Thomas Bauschert. Optimizing spectrum allocation in flex-grid optical networks. In *Photonic Networks; 18. ITG-Symposium*, pages 1–8. VDE, 2017.

[226] Marianna Angelou, Kostas Christodoulopoulos, D Klonidis, A Klekamp, Fred Buchali, E Varvarigos, and Ioannis Tomkos. Spectrum, cost and energy efficiency in fixed-grid and flex-grid networks. In *National Fiber Optic Engineers Conference*, pages NM3F–4. Optical Society of America, 2012.

[227] Neda Cvijetic, Akihiro Tanaka, Philip N Ji, Karthik Sethuraman, Shuji Murakami, and Ting Wang. Sdn and openflow for dynamic flex-grid optical access and aggregation networks. *Journal of Lightwave Technology*, 32(4):864–870, 2013.

[228] Christian Tipantuña and Xavier Hesselbach. Spectrum allocation in flex-grid optical networks repository, May 2020. URL https://github.com/ChristianTipantuna/Flex-gridOpticalNetwork.