



Article

NFV-Enabled Efficient Renewable and Non-Renewable Energy Management: Requirements and Algorithms

Christian Tipantuña^{1,2,*} and Xavier Hesselbach¹

¹ Department of Network Engineering, Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, E-08034 Barcelona, Spain; xavier.hesselbach@upc.edu

² Department of Electronics, Telecommunications and Information Networks, Escuela Politécnica Nacional, Ladrón de Guevara, E11-253 Quito, Ecuador

* Correspondence: christian.jose.tipantuna@upc.edu

Received: 8 September 2020; Accepted: 7 October 2020; Published: 14 October 2020



Abstract: The increasing worldwide energy demand, the CO₂ emissions generated due to the production and use of energy, climate change, and the depletion of natural resources are important concerns that require new solutions for energy generation and management. In order to ensure 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 alternatives. Accordingly, this paper presents the requirements and algorithmic solutions for efficient management of energy consumption, which aims to optimize the use of available energy, whether or not it is 100% renewable, by minimizing the waste of energy. The proposal works within a Demand-Response environment, uses Network Functions Virtualization as an enabling technology, and leverages the massive connectivity of the Internet of Things provided by modern communications technologies. The energy consumption optimization problem is formulated as an Integer Linear Program. It is optimally solved while using a brute-force search strategy, defined as OPTTs, to detect all concerns that are related to the problem. Given the \mathcal{NP} -hard nature of the problem and the non-polynomial complexity of OPTTs, some heuristic solutions are analyzed. Subsequently, a heuristic strategy, described as FASTTs based on a pre-partitioning method, is implemented. The simulation results validate our proposed energy management solution. Exact and heuristic strategies, when deployed in the Network Functions Virtualization domain, demonstrate improvements in the way that energy is consumed, thereby offering an increase in service processing. The evaluation results also show that FASTTs produces high-quality solutions that are close to those of OPTTs while executing $230\times$ – $5000\times$ faster.

Keywords: energy efficiency; energy management; demand response; green energy; NFV; pre-partitioning; service scheduling

1. Introduction

In recent years, the increasing energy consumption and CO₂ emissions caused by the deployment of new services and the proliferation of the Internet of Things (IoT) concept are becoming critical concerns driving the adoption of new solutions for energy production and management [1]. According to [2], the development of the Information and Communication Technology (ICT) industry could use up to 50% of the world's electricity in 2030 and up to 23% of CO₂ emissions. Consequently, the ICT and energy sectors have encouraged the use of renewable energy sources and deployment of adaptive energy management schemes with active customer-side participation. The purpose of these actions is to ensure sustainability and the long-term development of human society.

Renewable energy sources, such as solar and wind, are envisioned as an environmentally friendly and sustainable alternative to meet current and future energy demands [3]. However, the sporadic nature of green energy may cause instability when integrated into conventional energy sources [3]. This issue, along with the inefficient use of the generated energy (from renewable and non-renewable sources), have led to the development of Demand Response (DR) schemes, in which the consumption is adapted to generation dynamically using requests and actions exchanged between the Energy supplier (ES) and Energy Consumers (ECs) [4].

An essential component for deploying DR schemes is the ICT infrastructure that is needed for both the data exchange (i.e., notifications and instructions) between the ES and the ECs and the execution of management mechanisms (e.g., workload scheduling) [5]. In this regard, existing ICT solutions, such as cloud computing infrastructures and IoT technologies, can be used as key enablers for delivering feasible smart and programable management approaches for adaptive consumption of available supply. Specifically, the Network Functions Virtualization (NFV) technology can be used to deploy management mechanisms (carried out through algorithmic solutions) as software components, named Virtual Network Functions (VNFs), which can be used or modified on-demand [6], according to the dynamic requirements of the energy demands. The VNFs executed on Data Centers (DCs) (i.e., on cloud computing infrastructures) can dispose of a scalable computational capacity (mainly in terms of processing power and memory) needed to carry out adaptive energy management, as shown in [5].

Likewise, the massive connectivity 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 to participate in the energy management process [7]. It is expected that the number of devices connected to the Internet will increase considerably in the coming years (e.g., Cisco estimates that there will be 28.5 billion networked devices by 2022 [8]). Thus, eventually, all energy consumers (i.e., devices, appliances, or, in general, IoT infrastructures) will be able to be managed, for example, in terms of their consumption. This condition facilitates a true customer-side, the complete interaction between the ECs and the ES, the improvement in control, monitoring and management tasks, as well as the applicability to small scenarios, such as Home Energy Management Systems (HEMS) or environments as big as cities or countries. In this regard, an adaptive energy management proposal that is enabled by IoT and NFV technologies is a feasible, promising proposal that represents the evolution of smart grids, and it is aligned with the vision of future energy systems currently referred to as the Internet of Energy (IoE) [9].

This paper analyzes the adaptive energy management conditioned on the available supply of renewable and non-renewable sources from the point of view of communications systems. The proposal considers the manageability of the ECs, through massive connectivity and IoT technologies, and the dynamic and programmable behavior of management strategies that are deployed in an NFV domain and operating on scalable cloud computing infrastructures. This paper starts with an overview of the components of an NFV-enabled DR ecosystem. Subsequently, it presents the operating requirements and management mechanisms for adaptive energy management conditioned to the availability and with application to small and large-scale IoT scenarios. The proposal also includes the mathematical characterization of the ES and the ECs, as well as the adaptive consumption model using an Integer Linear Programming (ILP) formulation. The ILP model allows for us to identify the complexity of the problem and the possible algorithmic strategies to carry out adaptive energy management. In this context, this paper presents an algorithmic strategy, defined as OPTTs, which is based on a brute-force search mechanism and solves the ILP optimally. Given that OPTTs present a non-linear complexity, some heuristic strategies are analyzed in the scope of the problem. Subsequently, a fast heuristic strategy identified as FASTTs that is based on a pre-partitioning method is implemented. The optimal exact and heuristic strategies are evaluated through extensive simulations. The results are compared with a baseline case, in which no management solution is applied. The simulation results in different scenarios validate the performance of the proposed adaptive energy management

solution and demonstrate that the algorithmic strategies OPTTs and FASTTs, when deployed at the NFV domain, produces improvements in power consumption and demand processing. The major contributions of this paper are summarized, as follows:

- The description of an NFV-enabled energy management ecosystem, the requirements, and the mechanisms to perform adaptive energy management of available supply, whether or not it is renewable.
- The ILP formulation of the adaptive energy management model.
- An exact algorithmic strategy, defined as OPTTs, which based on a brute-force search combinatorial analysis solves the ILP optimally.
- The discussion of possible algorithms to tackle the hardness of the ILP and the computational complexity of OPTTs.
- A heuristic solution defined as FASTTs that based on a prepartitioning method produces solutions with a reduced running time (as compared to the optimal) and is applicable to large-scale environments.
- The evaluation of the strategies OPTTs and FASTTs through extensive simulations in order to confirm the improvements in energy consumption and demand processing.

Tables 1 and 2, respectively, present a list of the acronyms and most relevant notations used throughout the paper. The rest of the paper is organized, as follows. Section 2 discusses the related work. Section 3 introduces the problem of efficient energy consumption, the requirements, and the management mechanisms for the adaptive consumption of available energy. Section 4 presents the ILP formulation related to the energy management model. Section 5 describes OPTTs for solving the ILP problem optimally. A discussion of possible heuristic algorithms is addressed in Section 6. The heuristic solution, FASTTs, is presented in Section 7. Section 8 presents an overview of the deployment of the algorithmic strategies in the NFV domain. The evaluation results are discussed in Section 9. Finally, the conclusions and future work are drawn in Section 10.

Table 1. List of acronyms and corresponding definitions.

Acronym	Definition	Acronym	Definition
CS	Critical Services	IoE	Internet of Energy
DC	Data Center	IoT	Internet of Things
DR	Demand Response	MANO	Management and Orchestration
EC	Energy Consumer	NCS	Non-Critical Services
EM	Energy Manager	NFV	Network Functions Virtualization
ES	Energy Supplier	NFVO	NFV Orchestrator
HEMS	Home Energy Management System	SDN	Software Defined Network
ICT	Information and Communications Technologies	SFC	Service Function Chain
ILP	Integer Linear Programming	VNF	Virtual Network Function

Table 2. List of notations and corresponding definitions.

Notation	Definition	Notation	Definition
l	Priority identifier, $l \in \{1, \dots, L\}$	T_d^k	Lifetime of the service k
L	Number of priority levels	T_{init}^k	Starting time of the service k
m	Time interval where $P_{ES} > 0$	$T_{init}^{P_{ES}}$	Starting time of P_{ES}
N	Number of services	T_s^k	General or maximum time-shifting value
OPTTs	Optimal or exact algorithmic strategy	$T_{s_{bw}}^k$	Backward time-shifting of service k
$P_d^{k,l}$	Power demanded by service k with priority l	$T_{s_{fw}}^k$	Forward time-shifting of service k
P_D	Aggregated power demanded	U^k	Priority of service k
P_{ES}	Available power in the system	W	Maximum time horizon in the system
P_{NR}	Power from Non-Renewable sources	w_R	Weight associated to renewable energy
P_R	Power from Renewable sources	$\sigma_{P_{RES}}$	Standard deviation of P_{RES}
S_k	Service identifier, $k \in \{1, \dots, N\}$	σ_{T_s}	Standard deviation of T_s^k

2. Related Work

This section reviews the related work. Section 2.1 describes the ICT participation in energy management systems, including NFV-based approaches. Section 2.2 presents the key features of our proposal.

2.1. ICT-Based Energy Management Systems

In recent decades, the ICT infrastructure—especially IoT technology—has been identified as a key factor in the design and deployment of future energy systems because it enables the implementation of advanced energy monitoring, consumption, and management schemes [10]. 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 [7]. 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 [7], 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.

Given that IoT and DCs infrastructures have been identified as potential enablers for the deployment of DR programs, several studies have been proposed in this area [11]. For example, in [12], Wei et al. proposed an IoT-based common information model and a communication framework to deploy a DR energy management system for industrial consumers. The proposal mostly analyzes the operation from the facility side (i.e., from the ECs-side), and the experimental results demonstrate that the interoperability between entities (ES and ECs) not only results in improvements in energy efficiency, but also a reduction in the energy cost on the consumer side. Meanwhile, in [13], the authors present a HEMS that includes sensors, local energy production (from photovoltaic panels), and a central hub for monitoring energy consumption and executing the DR strategies for controlling loads. The results show that the integration of IoT technologies and renewable energy in the housing sector optimizes energy performance and it is a sustainable practice for reducing carbon emissions. On the other hand, DCs infrastructures, due to characteristics, such as advanced monitoring capabilities, significant influence on the operation of power grids, constant growth, and highly automated computational resources, have been considered to be suitable candidates for implementing DR programs that can cover neighborhoods or entire cities [11]. Subsequently, DCs in DR schemes enabled by communications standards, such as the OpenADR Communication Specification [14] or the NFV technology, can act as an Energy Manager (EM) and coordinate the voluntary change in consumption patterns from ECs. This action is performed by executing management mechanisms (e.g., workload scheduling) to adapt consumption to generation over time [15].

Regarding NFV, this technology has been used in energy management for several specific applications. In power grids, for example, Niedermeier et al. [16], proposed an NFV-enabled virtual advanced metering infrastructure network to transmit energy-related information about power consumption and production, with the aim of improving energy management. In mobile communications, the use of NFV to improve energy management and achieve energy efficiency has also been explored. In [17], the authors use the GWATT tool [18] to estimate the energy savings for three NFV use cases: Virtualized Evolved Packet Core, Virtualized Customer Premises Equipment, and Virtualized Radio Access Network. Meanwhile, other works [19,20] are focused on optimizing energy consumption and minimizing the energy footprint of the mobile network infrastructure (i.e., the transport networks and network functions) through the virtualization and management capabilities that are offered by NFV. Another interesting application field that is gaining momentum is drone-based communications, in this regard, Tipantuña et al., in [21], presents an NFV-based energy-aware management scheduling approach for a drone fleet with 5G connectivity capabilities that aims to guarantee a desired level of service availability. All of these works demonstrate the applicability of NFV in deploying energy efficient systems.

2.2. Features of Our Proposal

Our proposal is aligned with the requirements of future energy systems [10], because it considers: (i) the use of ICT infrastructures (NFV and IoT technologies) in energy management processes [22]; (ii) the use of renewable resources in energy provisioning [22]; and, (iii) adaptive consumption (consumption adjusted to the generation) with customer-side participation [23]. Most research works address energy efficiency by minimizing consumption or encouraging energy savings, whereas our proposal seeks the adaptive energy management of available supply, whether or not it is renewable. Technically, the adaptive consumption procedure is achieved by executing management mechanisms that aim to optimize the use of the available power or minimizing the waste of power at all times (both points of view completely equivalent).

In addition, energy management in our proposal is not solely approached from the supplier side, as described in [16], or solely from the customer side, as discussed in [12], nor is it exclusively focused on a specific infrastructure (e.g., network resource), as shown in [19]. On the contrary, our proposed solution presents the mathematical models of the ES and the ECs, and the formulation of the adaptive consumption model. The complexity that is related to adaptive energy consumption is also analyzed, and the performance metrics and algorithmic strategies are described in detail to motivate future work in this field.

3. Adaptive Energy Managements: Requirements, Components, and Mechanisms

Section 3.1 introduces the problem of inefficient use of available energy and an overview of the NFV-enabled ecosystem for energy management. Section 3.2 describes the requirements for adaptive energy consumption. Section 3.3 presents the characterization of provisioning and consumption, as well as the management mechanisms that enable the optimal use of available supply.

3.1. Problem Statement and Overview of the Energy Management Ecosystem

Because energy production is finite and not always synchronized with consumption, suppliers frequently face periods of energy scarcity or abundance. This operation condition produces that the energy demands cannot be processed due to a lack of supply or that generated energy may be wasted if it cannot be consumed or stored. Subsequently, inefficient energy distribution and use can cause partial or total energy shortages during specific periods (peak load hours) or in certain places, and an increase in tariffs for consumers. Therefore, the proper management of available energy at all times is essential. Accordingly, we propose an IoT and NFV-enabled adaptive energy management solution that is focused on the efficient (optimal) consumption of available power, whether or not it is renewable. The adaptive consumption procedure is carried out using management mechanisms (e.g., workload scheduling using time-shifting capabilities) implemented through algorithmic solutions and deployed on an NFV-enabled cloud computing infrastructure, as shown in Figure 1. In this context, if needed, Ref. [5] presents a detailed description of an NFV-enabled architecture for efficient energy management.

In summary, as illustrated in Figure 1, an IoT and NFV-enabled ecosystem for adaptive energy management is composed of three main components: (i) an ES, which provides power to the ecosystem; (ii) an NFV-Enabled EM, which manages services (energy demands from ECs) and energy resources (from the ES), and it is responsible for executing management algorithmic strategies that enable the efficient/optimal use of available energy; and, (iii) the ECs, who demand energy and modify their energy consumption (advance or delay demand execution) according the instructions that are received from the EM. The requirements and management mechanisms for adaptive energy consumption are presented below.

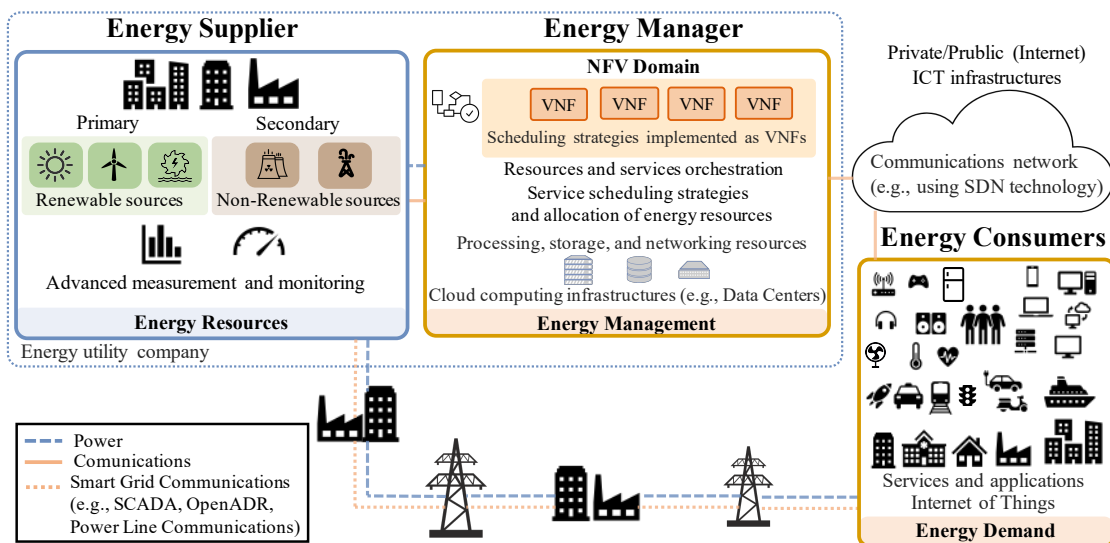


Figure 1. Overview of an Network Functions Virtualization (NFV)-enabled adaptive energy management ecosystem.

3.2. Requirements for Adaptive Energy Management

- Adaptive consumption of available energy:** for optimal use of available energy, the consumption pattern must be adapted to the existing energy resources at all times, whether or not they are deterministic (this is the case when the energy comes from renewable sources). In order to carry out this procedure, the ECs must collaborate with the ES and be willing to shift the execution of services (i.e., advance or delay consumption) according to availability. In this way, the ES can stimulate consumption or propose the deferment of services during periods of surplus or shortage, respectively. The cooperation between ES and ECs is defined by means of contracts in which technical (service parameters) and economic (pricing schemes) aspects are established. Technically, the consumption adaptation procedure is executed through management mechanisms, such as those described in Section 3.3.3.
- Customer-side participation in the energy management process:** in an adaptive energy management system, the ECs must interact with the ES. The level of interaction between these stakeholders can be fully automated or may require end-user participation, depending on the characteristics of each service and application environment (e.g., residential, industrial, or public infrastructures scenarios). In summary, each time an EC requests energy for service execution, a handshake process must be established with the ES (technically, this process is performed by the EM; see Figure 1), in which information on energy demands and available resources is exchanged. With the collected information, the ES (technically, the EM) must perform the corresponding calculations in order to inform the ECs of the consumption conditions (i.e., the services to which energy can be allocated and their respective execution time). This procedure is described in more detail in Section 3.3.3. Regarding the functional requirement, in order for ECs to participate in the energy management ecosystem, they must have automated IoT capabilities to allow communication (with the ES) and manageability (activation/deactivation of consumption). Currently, a large number of devices are manufactured with embedded communication systems. In addition, if needed, affordable platforms (e.g., Arduino, Raspberry, or ESP32 platforms) can be integrated into virtually any device in order to provide connectivity and automation capabilities. In this context, different protocols, interfaces, and IoT solutions can be used for different purposes and applications. However, a discussion of specific solutions for these purposes is beyond the scope of this paper.

- *ICT infrastructure for the establishment of communication between components:* the communications infrastructure is a fundamental component of an adaptive energy management system [10]. This infrastructure must enable the reliable and secure exchange of information and instructions related to energy management (i.e., information on service parameters, notifications, and consumption indications). Different communication technologies can be used to interconnect the components (ES-EM-ECs), as shown in Figure 1. However, the most suitable option of an underlying network to communicate the different stakeholders is the Software Defined Networking (SDN) technology due to complementarity with the NFV realm [24]. Regarding the ownership, operation, and maintenance aspects of the ICT infrastructure, different options can be analyzed. In the first instance, the ICT infrastructure could be leased by energy suppliers and distributors, but it is expected that, with the massive connectivity and IoT and IoE trends, the energy sector will invest and deploy sophisticated communications networks. The current smart grids could be used as a baseline infrastructure for this purpose.
- *ICT infrastructure for performing management strategies:* the ecosystem for adaptive energy management requires an entity that coordinates all actions between the ES and the ECs. In the scheme presented in Figure 1 these functionalities are performed by the EM; this component is normally part of the ES and is in charge of performing all calculations and executing all strategies (algorithms) that are associated with energy management from ES to ECs. In this regard, Ref. [5] demonstrates that NFV technology deployed on cloud computing infrastructures is an ideal solution to meet the scalability, flexibility, and reconfigurability requirements of adaptive energy consumption. An NFV-enabled EM can grow proportionally (increase in computing, storage, and networking resources) according to the varied requirements of the ECs, and it can dispose of the Management and Orchestration (MANO) entities [6], so that all the components of the generation and consumption ecosystem work in a coordinated manner. Thanks to NFV technology, the manageability of the ECs through the underlying network (e.g., SDN) is separated (abstracted) from the functionality (i.e., management strategies), a feature that enables energy management in different scenarios (e.g., HEMS, companies, or smart cities).
- *Primary use of renewable energy sources and control of their contribution:* although this requirement is not indispensable in optimizing the consumption of the available supply, our proposal encourages the primary use of green energy and the transition to 100% renewable energy systems, as an emissions-aware and sustainable solution. In this regard, the proposed energy management mechanisms, such as workload scheduling through time-shifting capabilities, prioritization in energy supply, or rejection of energy demands if necessary, enable the optimal use of the sporadic energy capacity from green sources such as solar or wind. Furthermore, given that the change towards zero-emission energy systems is gradual and it requires the co-existence of both renewable and non-renewable sources, controlling the contribution of each source according to its availability and need is an important parameter in the operation of the ES.

3.3. Components of the NFV-Enabled Ecosystem and Proposed Energy Management Mechanisms

3.3.1. Energy Supplier

In the ecosystem for energy management, the ES is equipped with advanced control and monitoring systems, performs the energy-mixing process (Process to obtain energy for direct use, combining different primary energy sources [25]), and it may be composed of several suppliers or sub-suppliers. For analytical simplicity, in the proposal, the ES is regarded as a single entity and it is characterized by its power production capacity. The power level delivered by the ES, or available power in the system, defined as P_{ES} has an initial time $T_{init}^{P_{ES}}$ and a duration equal to m . The P_{ES} is *de facto* the power received at the point of consumption, regardless of losses, and it is a combination of the power

obtained from renewable sources (P_R), such as hydroelectric, solar, and wind, and from non-renewable sources (P_{NR}), such as coal, natural, and gas, as indicated in Equations (1)–(3), respectively.

$$P_{ES} = P_R + P_{NR} \tag{1}$$

$$P_R = P_{Es} \times w_R \tag{2}$$

$$P_{NR} = P_{Es} \times (1 - w_R) \tag{3}$$

where the weight $w_R \in [0, 1]$, controls the contribution of the renewable energy in the total generated power P_{Es} . Thus, the $w_R = 1$ if the provisioning is exclusively from green sources.

3.3.2. Energy Consumers

The ECs are the IoT infrastructures that demand energy to execute services (i.e., workloads, jobs, or applications). They have processing, networking (e.g., SDN compatible), and control (automation systems to activate or deactivate the consumption) capabilities. The ES manages the ECs through the EM. In the proposed energy management model, the ECs are characterized by their energy consumption. Subsequently, an energy demand referred to also as a service is characterized by: (i) an identifier k , with $k \in \{1, \dots, N\}$; (ii) a starting time (T_{init}^k); (iii) a power demanded (P_d^k); and, (iv) a finite duration or lifetime (T_d^k). For simplicity, the management model assumes that P_d^k is invariant during T_d^k . Besides, the calculations for energy use optimization consider the joint action of all N services, which demand a total power demanded equal to P_D , a value that is equivalent to the sum of each service.

Given that the execution of some services is more important than that of others in a realistic environment, the energy management model also considers the prioritization in the power supply by including a priority-level identifier l , with $l \in \{1, \dots, L\}$, to each service. These priority levels can be defined in the contractual terms between the EC and the ES [15], and with this information, the EM can identify the services that cannot be processed if $P_{ES} < P_D$. The ecosystem (EM) prioritizes the processing of the services with the highest priorities (i.e., the demands are processed in descending order, with $l = 1$ being the highest priority level). The P_D considering the prioritization level of demands can be expressed as:

$$\forall k \in N, \forall l \in L : P_D = \sum_{k=1}^N \sum_{l=1}^L P_d^{k,l} \tag{4}$$

The services, according to their priority level, are categorized as Critical Services (CS) and Non-Critical Services (NCS). The CS (i.e., services with $l = 1$) cannot be shiftable in time, their execution is mandatory, and consequently, the system must always guarantee their energy provisioning. Some examples of CS are services (devices) used for disasters, emergencies, surgical interventions and human life support, and road safety for passengers and drivers. Instead, NCS (i.e., $l \in \{2, \dots, L\}$) can be shifted in time, their execution is critical, and they may even not be executed if $P_{ES} < P_D$. Some examples of NCS include entertainment systems, home air conditioning systems, services that are associated with cleaning (washing machines and dryers), and non-essential home automation system. The power demanded by the CS, the NCS, and the P_D is presented in Equations (5)–(7), respectively.

$$P_d^{k,cs} = \sum_{k=1}^N P_d^{k,1} \tag{5}$$

$$P_d^{k,ncs} = \sum_{k=1}^N \sum_{l=2}^L P_d^{k,l} \tag{6}$$

$$P_D = P_d^{k,cs} + P_d^{k,ncs} \tag{7}$$

3.3.3. NFV-Enabled Energy Manager and Mechanisms for Adaptive Consumption

The energy adaptation process is continuously carried out every time that an EC requests energy consumption. Thus, before using energy, the ECs notify the EM of their demands. The EM performs the calculations (execution of algorithmic strategies) for the efficient (optimal) utilization of available energy. As a result of this procedure, the EM knows the service(s) that can be executed and their execution time (a service can be advanced, delayed in time, or rejected). Subsequently, the consumption instructions are sent to the ECs, and the energy allocation is performed for the service(s) that can be executed. Depending on the amount of P_{ES} , the ES through the EM can promote an increase or decrease in consumption through the anticipated or delayed execution of services. Figure 2 shows a summary of the adaptation process between components.

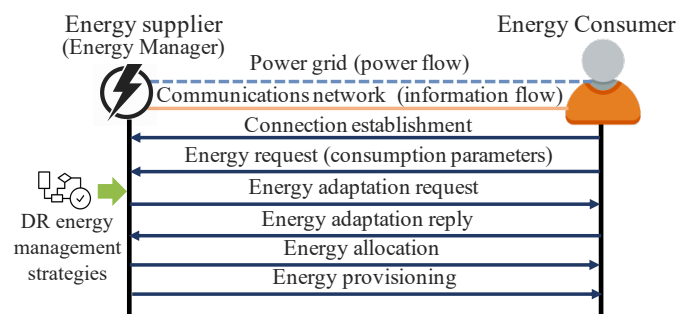


Figure 2. Energy adaptation process (handshake between the Energy Supplier (ES) and the Energy Consumer (EC)). Adapted from [5].

The strategies that are considered by the EM to optimize the use of P_{ES} are described below and they are used in the model that is presented in Section 4 and in the algorithmic strategies described in Sections 5 and 7.

- *Adaptation of consumption to availability by exploiting time-shifting capabilities of services:* the adaptation of the consumption pattern to the availability, is achieved with the use of temporary displacement on the service. Thus, a service k (S_k) can be affected by a *time-shifting* (T_s^k) forward (i.e., $T_{s_{fw}}^k$) or backward (i.e., $T_{s_{bw}}^k$) within a finite interval of time horizon W . These parameters are independent for each service S_k , and the management strategies (algorithms as shown in Section 5) deployed in the NFV domain determine the efficient (optimal) scheduling of services to optimize the use of P_{ES} .
- *Prioritization in energy distribution and rejection of demands:* if $P_{ES} < P_D$, the the EM allocates the energy resources to the services according to the category to which they belong, i.e., the supply is prioritized for the execution of CS ($l = 1$). The remaining energy is allocated to the NCS in decreasing order according to their priority level (from $l = 2$ down to $l = L$). Thus, services with a lower priority level are more likely to be unprocessed or rejected. If all services have an equal level of priority, the service scheduling algorithms aim to use energy as efficiently as possible and, second, to try to meet as many demands (services) as possible. This procedure is explained in detail in the algorithm in Figure 3. Other alternatives that could be analyzed in future work, if $P_{ES} < P_D$, are the degradation of the quality of services (e.g., decrease in display brightness of a device), and the use of energy stored in batteries during energy surplus periods. A service S_k is completely defined by the parameters that are listed in Table 3, and an example is illustrated in Figure 4.

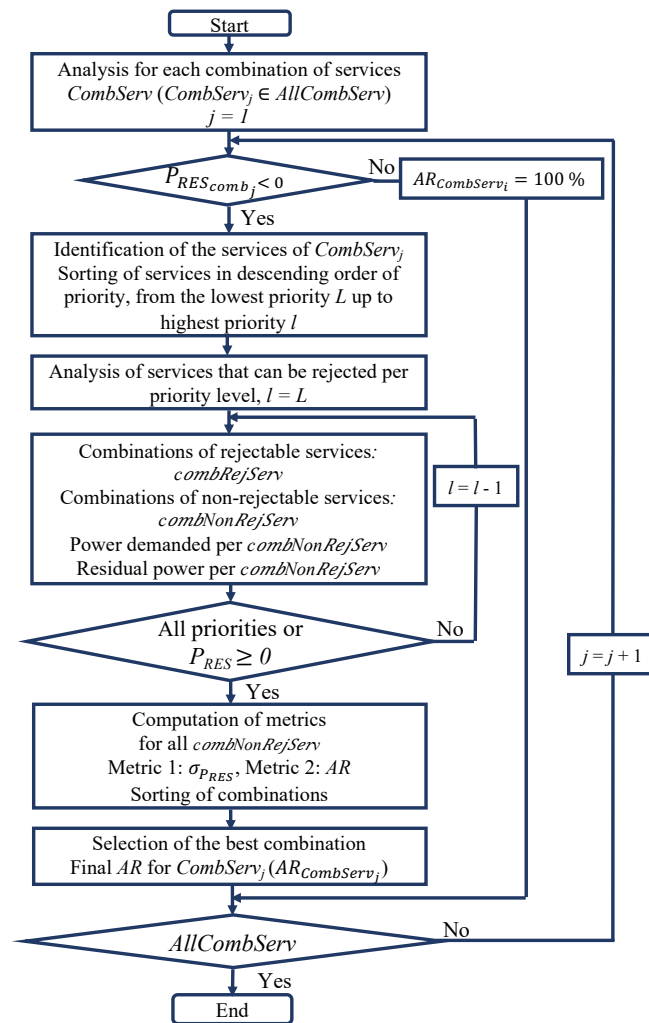


Figure 3. Flow chart for AR computation of OPTTs when there exist service rejection.

Table 3. Parameters of services or power demands.

Parameter	Description	Unit/Comment
N	Number of services	Integer number
S_k	Service identifier	$k \in \{1, \dots, N\}$
L	Number of priority levels	Integer number
l	Priority identifier	$l \in \{1, \dots, L\}$
P^k	Power demanded by service k	Power units
T_{init}^k	Starting time of service k	Time units
T_d^k	Lifetime of service k	Time units
T_{sbw}^k	Backward time-shifting of service k	Time units
T_{sfw}^k	Forward time-shifting of service k	Time units
U^k	Priority of service k	Integer number

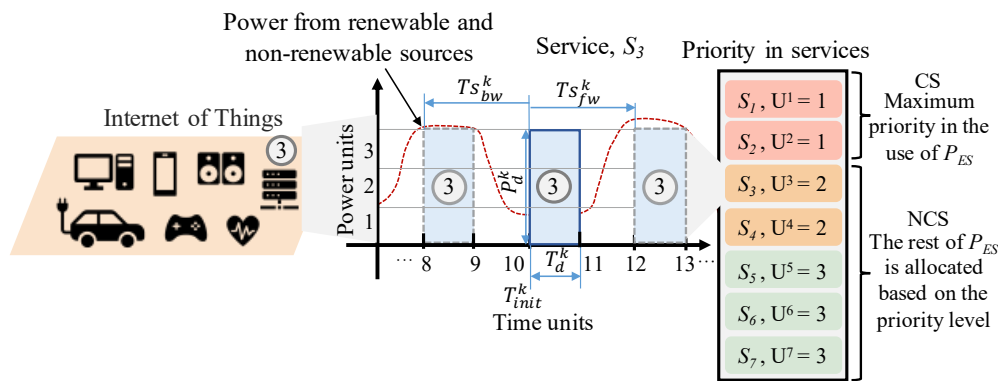


Figure 4. Graphical representation of a service and interaction with the available power. Parameters of S_3 : $N = 1, L = 3, l = 2, P_d^3 = 3, T_{init}^3 = 10, T_d^3 = 1, T_{s_{bw}}^3 = 2, T_{s_{fw}}^3 = 2,$ and $U^3 = 2.$

4. ILP Problem Formulation: OptTs

The proposed adaptive energy management approach uses a discrete-time model divided into time slots of equal duration that in practical implementations can correspond to different time units (e.g., 10-min or one-hour time intervals). This time model offers flexibility to the system in order to deal with various applications, processes, and environments. For instance, if services evolve slowly, then time slots that correspond to extended periods (e.g., units of hours) could be used. Conversely, if the system presents rapid changes or transitions, time intervals representing small units of time (e.g., units of minutes or units or tens of seconds) could be employed. Thus, a service S_k may have a T_d^k of one or several time slots and could be shifted zero, one, or more than one time slots, if needed. When considering these conditions, the adaptive consumption of available supply is achieved through the minimization of power wasted. This problem is defined by an objective function and several constraints, as shown below.

4.1. Objective Function

With finite P_{ES} and P_D values, the objective to minimize the unused or remaining available power can be conceptually expressed as the difference between these parameters, as shown in Equation (8). In each time slot i , this objective function of can be expressed, as shown in Equation (9).

$$\text{minimize } \{P_{ES} - P_D\} \tag{8}$$

$$\forall i \in W : \text{minimize } \left\{ \sum_{i=1}^W (P_{ES}[i] - P_D[i]) \right\} \tag{9}$$

The difference between P_{ES} and P_D is defined as Residual Power (P_{RES}), and it is given by:

$$\forall i \in W : P_{RES} = \sum_{i=1}^W (P_{ES}[i] - P_D[i]) \tag{10}$$

Thus, the objective of the proposed adaptive energy management model is to minimize the P_{RES} , given the constraints presented below.

Domain Constraints

The role of energy provisioning by the ES and the non-negative P_{RES} is ensured by Equations (11) and (12), respectively.

$$P_{ES}[i] \geq 0 \tag{11}$$

$$(P_{ES}[i] - P_D[i]) \geq 0 \tag{12}$$

Given that, mathematically, the P_D can be greater than P_{ES} , a situation that occurs due to a low generation or high load, this negative difference or missing power is denoted as Lack Power (P_{LACK}). The P_{LACK} represents the amount of energy that is needed to process all demands under the current conditions of the system. This parameter can be expressed as:

$$P_{LACK}[i] = \begin{cases} |P_{ES}[i] - P_D[i]| & \text{if } P_{ES}[i] < P_D[i], \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

4.2. Capacity Constraint

The maximum capacity of the system is determined by the maximum P_{ES} value and it is defined by Equation (14).

$$\forall k \in N, \forall l \in L : \sum_{k=1}^N \sum_{l=1}^L P_d^{k,l} \times x_k[i] \leq P_{ES}[i], x_k \in \{0, 1\} \tag{14}$$

where the decision variable x_k represents the allocation of energy resources for the processing of the service S_k .

$$x_k[i] = \begin{cases} 1 & \text{if the service } S_k \text{ is processed at} \\ & \text{time slot } i, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

In the proposal, it is assumed that the processing of complete services, i.e., partial processing of a service is not allowed. In fact, within the framework of a DR system, these services are considered to be rejected or not-processed demands. In addition, the consistency between the service and its priority is validated by Equation (16). This condition ensures the existence of the service for its corresponding unique priority level.

$$P_d^{k,l} = \begin{cases} \text{Power demanded by } S_k \text{ with priority } l & \text{if the } S_k \text{ with priority } l \text{ exists, at } T_{init}^k \text{ and for } T_d^k, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

4.3. Time Constraints

The analysis of the whole system starts at a time zero ($t = 0$), which is assured by Equations (17) and (18). Moreover, Equation (19) ensures a finite maximum time horizon Equation (19), in which all services and time shiting values (T_{bw}^k, T_{fw}^k) can be analyzed.

$$\forall k \in N : T_{init}^k \geq 0 \tag{17}$$

$$\forall k \in N : \{T_{init}^k - T_{bw}^k\} \geq 0 \tag{18}$$

$$\forall k \in N : W \geq \max\{T_{init}^k + T_d^k + T_{fw}^k\} \tag{19}$$

In addition, related to the domain constraint presebted in Equations (11), (20) and (21) ensure the supply of non-zero power during a time interval m , within the total scope of the system.

$$T_{init}^{P_{ES}} \geq 0 \tag{20}$$

$$W \geq T_{init}^{P_{ES}} + m \tag{21}$$

For simplicity, in the nomenclature in the rest of the paper, the time slot identifier has been omitted. However, this parameter has been considered in the design of the scheduling strategies presented in Sections 5 and 7.

4.4. Hardness of the Problem

The energy resources allocation for a set of services, each with a finite power demanded and a priority level subject to maximum available capacity, as described in Equation (14), can be seen as the objective of the Knapsack problem of placing the most valuable or useful items without overloading the knapsack [26]. Thus, following this analogy, we can conclude that the problem of minimizing P_{RES} while respecting the capacity and time constraints falls into the general category of the Knapsack Problem, so that its complexity is \mathcal{NP} -hard.

5. Exact Solution: OPTTs

In order to find the optimal service scheduling that minimizes the P_{RES} , a brute-force search combinatorial algorithm OPTTsCOST has been developed following the criteria that are presented in Section 3. This algorithm computes all possible combinations of N services (i.e., S_1, S_2, \dots, S_N), considering all possible values of time-shifting (i.e., $\{T_{sbw}^k, \dots, 0, \dots, T_{s_{fw}}^k\}$) to which the services may be subject. After analyzing all of the combinations, the one with the best performance metrics is selected, which represents the best combination of services with a P_D (P_{Dcomb}) that produces the minimization of P_{RES} and allows for the execution of as many services as possible. Figure 5 depicts an example of the scheduling strategy for $N = 3$ services.

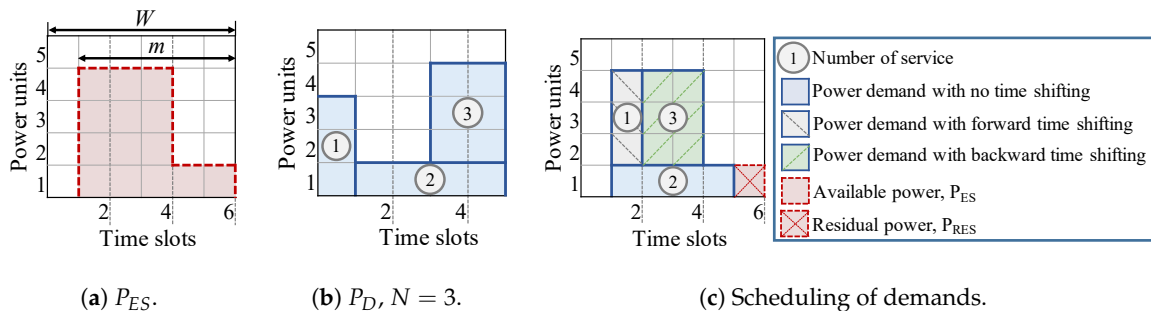


Figure 5. Example of scheduling strategy for $N = 3$ services and $T_s = 1$ [time slots]. Time-shifting applied to services: $T_{sbw}^1 = 1$ for S_1 , $T_s^2 = 0$ for S_2 , and $T_{s_{fw}}^3 = 1$ for S_3 .

5.1. Metrics

5.1.1. Standard Deviation of Residual Power ($\sigma_{P_{RES}}$)

This performance metric measures the amount of P_{RES} of a given combination of services ($CombServ$). The better use of P_{ES} produces a lower $\sigma_{P_{RES}}$ value. If $P_{ES} = P_D$ if all power is used $\sigma_{P_{RES}}=0$. The expression of $\sigma_{P_{RES}}$ within m is:

$$\forall j \in AllCombServ : \sigma_{P_{RES}_j} = \sqrt{\frac{\sum (P_{REScomb_j})^2}{m}} \quad (22)$$

5.1.2. Acceptance Ratio (AR)

This performance metric measures the percentage of processed services. If $P_{ES} < P_D$, then one or several services should be rejected ($RejServ$), using certain criteria as shown in Figure 3. The AR is given by:

$$AR = \frac{N - RejServ}{N} \times 100\% \quad (23)$$

when considering the AR metric, the missing power can be rewritten as $P_{LACK(AR=100\%)}$. In addition, the value of m , so that all power demands are met ($AR = 100\%$) can be written as:

$$m \simeq \frac{\sum_{k=i}^N P_d^k \times T_d^k}{P_{ES}} \tag{24}$$

where, the numerator and denominator represent the mean values of consumption and provision, respectively. In the case that all services have the same duration (T_d^k) and consumption (P_d^k), m can be defined, as indicated in Equations (25) and (26).

$$mInit = \left\lceil \frac{\sum_{k=i}^N P_d^k \times T_d^k}{P_{ES} - mod(P_{ES}, P_d^k)} \right\rceil \tag{25}$$

$$m = \begin{cases} mInit + T_d^k - mod(mInit, T_d^k) & \text{if } mod(mInit, T_d^k) \neq 0, \\ mInit & \text{if } mod(mInit, T_d^k) = 0. \end{cases} \tag{26}$$

5.1.3. Standard Deviation of Time-Shifting (σ_{Ts})

This performance metric measures, the T_s^k (backward or a forward) performed by the different services that form a *CombServ*. The ideal value is $\sigma_{Ts} = 0$, because it represents that all services are executed respecting their initial execution time (i.e., no time-shifting application). The σ_{Ts} can be expressed as:

$$\forall j \in AllCombServ : \sigma_{Ts_j} = \sqrt{\frac{(\sum_{k \in N} T_s^{k,j})^2}{N}} \tag{27}$$

5.2. Optimal Power Management Strategy OPTTs

The strategy OPTTs is composed of a main algorithm, which is responsible for analyzing all combinations of services, and a secondary algorithm, which computes the AR metric, as shown in Figures 3 and 6, respectively. For practical reasons in the nomenclature, in the rest of the document, the parameter that represents the service priority level (l) has been omitted; however, it has been considered in the design of the scheduling strategies. Additionally, to simplify the analysis, especially for the computation of complexity (Section 5.3), we consider $T_{sbw}^k = T_{sfw}^k = T_s, \forall S_k$. The steps that were carried out by OPTTs are summarized, as follows.

5.2.1. Variations per Service (*VarServ*)

A variation of a service is the result of the application of a specific discrete time-shifting value on the T_{init}^k . The algorithmic strategy analyzes all possible variation within $\{T_{sbw}^k, \dots, 0, \dots, T_{sfw}^k\}$. The total number of variations analyzed by OPTTs are defined as *AllVarServ* and they are given by.

$$AllVarServ = 2 \times N \times Ts + N \tag{28}$$

5.2.2. Combinations of Services (*CombServ*) and Computation of Metrics

The algorithmic strategy simultaneously analyzes the configuration of all N services. Specifically, it performs the combinatorial analysis of all existing variations of a service with all variations of the other services. The set of N different variations is defined as a combination of services (*CombServ*), which demands a certain power level P_{Dcomb} . Analysis of all variations of all services (*VarServ*) produces a total number of combinations *AllCombServ*, which is expressed as:

$$AllCombServ = (2 \times Ts + 1)^N \tag{29}$$

This step mainly contributes to the growth of complexity; for instance, $N = 10$ and $T_s = 4$, produce over three-billion combinations to be processed. For each *CombServ*, the algorithm evaluates the performance metrics.

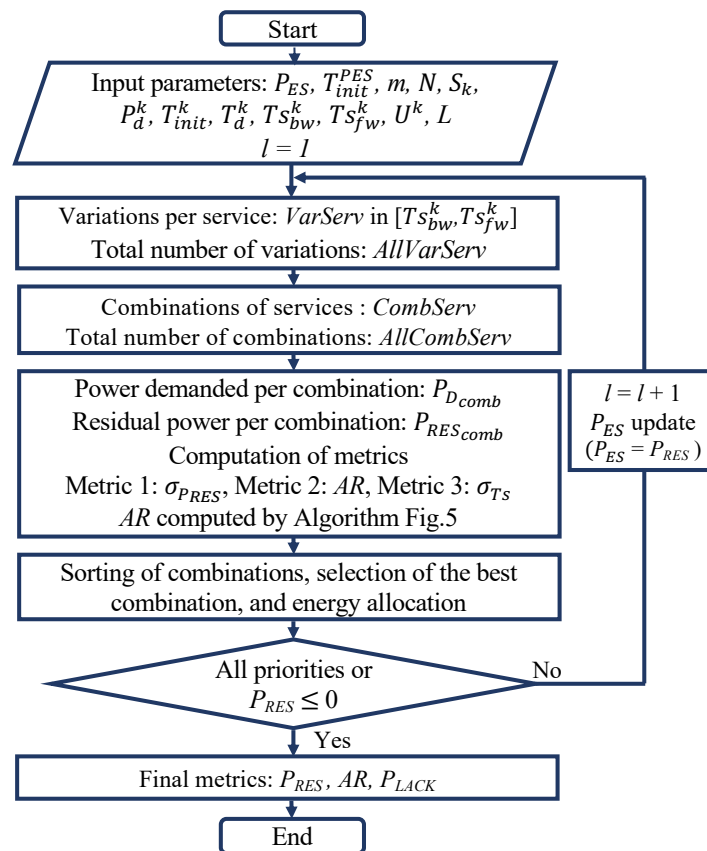


Figure 6. Flow chart of the exact scheduling algorithm.

5.2.3. Sorting of Combinations and Selection of the Best Combination

Nested quicksort applied to all combinations, according to: (i) ascending $\sigma_{P_{RES}}$, (ii) descending AR , and (iii) ascending σ_{T_s} . Subsequently, the best combination (i.e., the first in the sorted list) is selected, the power is allocated to the services that can be processed, and the metric P_{RES} is computed. The best *CombServ* represents the optimal scheduling of services, minimizing the P_{RES} value. In the example presented in Figure 5, when only considering a priority level, the combination that enables the P_{RES} minimization is shown in Figure 5c.

5.2.4. Iterative Analysis of Priorities

The algorithmic strategy OPTTs repeat the steps above for each priority level until all priority levels are analyzed or until the system has no power ($P_{RES} \leq 0$). Finally, the algorithm presents the optimal service scheduling and the performance metrics P_{RES} , AR , and P_{LACK} .

5.3. Complexity Analysis of OPTTs

The computational complexity of OPTTs, as shown in Equation (29), is related to the exploration of all search space, which is given by the values of N and T_s^k . Equation (29) shows that an increase in N leads to a non-linear growth in complexity; however, the impact due to the exclusive action of T_s^k is not evident. The first derivative of the linearized version of Equation (29) is evaluated with respect to N in order to analyze the contribution of T_s^k in the growth of the problem, as shown in Figure 7. Subsequently, the obtained results allow for us to verify that an increase in the value of T_s^k also causes

a non-linear growth of the search space. Thus, the increase in N or T_s^k leads a non-linear growth of complexity, which can be reflected in greater use of computational resources or an increase in the execution time. The main (Figure 6) and secondary (Figure 3) algorithms contribute to the increase of complexity; however, the major contribution comes from the three first steps of the main algorithm. The growth rate of OPTTs as a function of N can be expressed as:

$$f(N) = N + (2 \times N \times T_s + N) + (2 \times T_s + 1)^N \tag{30}$$

where, discarding low-order terms, the last term in Equation (30) reveals that OPTTs has an exponential complexity with an order of growth $\mathcal{O}(2^N)$, according to the Big-O notation. The complexity of OPTTs indicates the limits and drawbacks for applicability to large values of N and T_s^k and the computational capacity (in terms of processing and memory) needed for its execution.

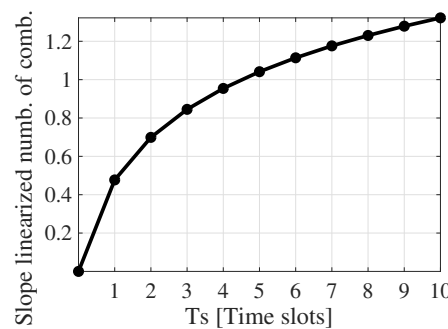


Figure 7. Example of growth rate of OPTTs based on T_s . Parameter: $T_s^k = \{0, \dots, 10\}$.

6. Analysis of Heuristic Strategies

The \mathcal{NP} -hard nature of the problem (Section 4.4) and the intractability of OPTTs for large values of N or T_s^k (Section 5.3) motivates the development low complexity strategies to find feasible solutions within a reasonable computing time. In this context, Section 4.4 has proven that the P_{RES} minimization falls into the category of a 1/0 Knapsack Problem. Specifically, it can be categorized as a multiple-choice Knapsack Problem [27], due to the production of the different variations (*VarServ*) belonging to the same service. Subsequently, the existing literature proposes a number of techniques and strategies for efficiently solving this problem [27,28], taking into account that, in some cases, the reduction of the execution time is achieved by relaxing the exact solution. Based on the literature surveyed and to cover as far as possible the main categories of methods or techniques that were developed for solving the 1/0 Knapsack Problem (in our case, the P_{RES} minimization problem), we have examined four strategies. According to the degree of difficulty in implementation, the examined strategies are: (i) a Pre-partition strategy (FASTTs), based on a divide-conquer approach of the total number of analyzed services; (ii) a Greedy strategy, based on a constructive algorithm in which the best *VarServ* is selected based on a performance criterion (e.g., tradeoff P_{RES} and AR); (iii) a metaheuristic Genetic algorithm that evolves as a function of the analyzed population and genetic operator (crossover and mutation) applied; and, (iv) a Dynamic programming approach, which is an optimization and programming technique that can be applied for efficiently solving a variety of computational problems. Table 4 summarizes the features and implementation challenges of the possible heuristic strategies.

Based on a preliminary assessment of the strategies presented in Table 4, in this article we have decided to analyze, in detail, the Pre-partitioning strategy, as shown in Section 7, because its implementation requires few modifications on the algorithm OPTTs. Besides, preliminary results show that this method can produce high-quality solutions in a reduced time when compared to the optimal strategy. The implementation and evaluation of the other strategies will be addressed in detail in future work.

Table 4. Heuristics strategies analyzed to solve the P_{RES} minimization problem efficiently.

Strategy/Technique	General Description	Features and Challenges
Pre-partitioning Strategy	Strategy based on a divide-and-conquer method. In this strategy, the original set of N is divided into subsets. The OPTTs strategy is applied to each of these subsets. This operation reduces the complexity by reducing the $VarSer$ and the $CombServ$ in each processed subset. This heuristic strategy iteratively processes all sub-problems, and the solution is obtained as a combination of partial solutions.	<ul style="list-style-type: none"> - Easy implementation from OPTTs. - The strategy can be seen as a generalized implementation of OPTTs. - The size of the search space can be controlled, because it can decrease as divisions increase. - The control of the accuracy and running time within certain thresholds can be set based on the number of divisions implemented.
Greedy Approach	Iterative and constructive algorithm, in which the $VarSer$ to be processed, one at a time, is selected based on the value of a parameter (e.g., $P_d^{k,l}$, T_d^k or the ratio between $P_d^{k,l}$ and T_d^k), respecting the P_{ES} and with the aim of first minimizing P_{RES} and then increasing AR .	<ul style="list-style-type: none"> - The strategy must be built from scratch and there are many feasible implementations. - Good running time performance and possible configuration for achieving a linear complexity growth. However, there is no guarantee on the quality of the solution, because it can vary widely according to the criterion established to select the $VarSer$.
Genetic Algorithm	Strategy inspired by the behavior of biological systems. First, an initial population of chromosomes is created, in which each chromosome represents a possible $CombServ$ (solution) formed by randomly selected $VarSer$. Later, the population evolves along with generations considering the action of the genomic operators (crossover and mutation) and the selection of fittest individuals (combinations with the best P_{RES} metric). Finally, in the last generation, the solution with the best fitness function (minimum P_{RES}) is selected.	<ul style="list-style-type: none"> - Medium complexity implementation due to features and requirements of the proposal (e.g., time-shifting, priorities, rejection). - Nondeterministic solution. It depends on the random values chosen in the creation of populations (initial and offspring) and the selection of genetic operators (crossover and mutations). - Good running time performance, but the final solution may require many generations if the parameters for population creation or stop criteria are not properly established.
Dynamic Programming Approach	Strategy that uses a dynamic programming method. The algorithm simplifies the analysis of N simultaneous services by analyzing and solving them one at a time. Systematically, using a bottom-up approach, the results of a previous set of services (considering their $VarSer$) are stored and used for solving a greater N .	<ul style="list-style-type: none"> - Good running time performance, but difficult implementation due to the two-dimensional behavior of the problem (power and time parameters) and multiple-choice analysis for $VarSer$ selection.

7. Heuristic Solution: FASTTs

The proposed strategy is defined as FASTTs and it is inspired by the divide-and-conquer algorithm [29]. This approach is a well-known design technique and it is the key to solving a variety of problems, such as classification, binary search, or the multiplication of large numbers [29]; it has been proven to produce efficient solutions with little or none loss of accuracy [30], and it only requires few modifications on OPTTs for implementation. The main idea of FASTTs is to reduce the set of total combinations to be explored through the division of the original set of N services into two or more subsets of equal or similar size and complexity, which are defined as *partitions* and they represent the individual subproblems to be solved using OPTTs. These partial solutions are then merged to obtain the best possible (optimal or suboptimal) scheduling of services while keeping the objective of P_{RES} minimization. Thus, FASTTs reduces the complexity by analyzing the sum of the combinations of all partitions, rather than the combinations produced by the original set of services. The total number of possible partitions ($numPart$) is given by:

$$1 \leq numPart \leq N \tag{31}$$

The heuristic strategy can be seen as a generalization of OPTTs (when the $numPart = 1$ partition), and it might impose limits on the performance with respect to OPTTs. For instance, if the system requires a faster running time, a greater number of *partitions* can be performed at the possible cost of performance degradation; conversely, if the system demands a very low-performance degradation, then a few number of partitions can be considered at the cost of an increase in complexity. Figure 8 depicts an overview of OPTTs and FASTTs for $N = 8$ services. In addition, the size of each partition ($lenPart$) is also restricted to being an integer due to the integer nature of N .

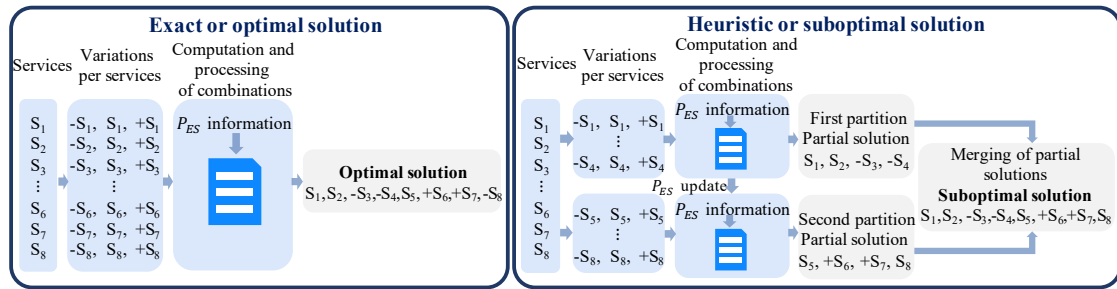


Figure 8. Overview of OPTTs and FASTTs strategies. Parameters: $N = 8$, $T_s^k = 1$, and $numPart = 2$ partitions for FASTTs.

7.1. Suboptimal Power Strategy FASTTs

Figure 9 explains the heuristics algorithm FASTTs, the steps carried out are briefly summarized below, and its complexity is analyzed in Section 7.2.

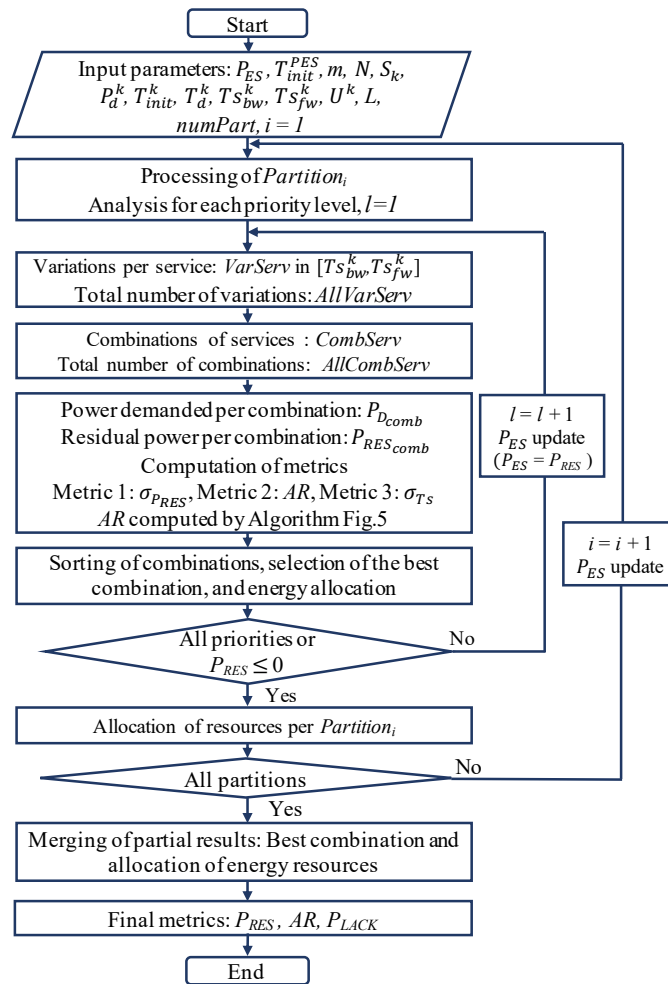


Figure 9. Flow chart of the heuristic algorithm.

7.1.1. Partition Phase

The set of N services is divided into $numPart$ partitions of equal or similar size.

7.1.2. Computation of Combinations per Partition

For each partition, the algorithm calculates the variations per service (*VarServ*), the combinations of services (*CombServ*), and their metrics $\sigma_{P_{RES}}$, *AR*, and σ_{Ts} , then it selects the best combination of services for each of the priority levels within the partitioned domain. After that, the P_D and P_{RES} are calculated. Thus, the P_{RES} of a previous partition is the P_{ES} of the next partition.

7.1.3. Iterative Process

The algorithm analyzes all partitions in order from *Partition*₁ up to *Partition*_{numPart}.

7.1.4. Merging of Partial Solution

To obtain the final scheduling of the services, all of the partial solutions are combined. Subsequently, the final metrics P_{RES} , *AR*, and $P_{LACK(AR=100\%)}$ are computed.

7.2. Complexity Analysis of FASTTs

The complexity of FASTTs is related to the number of combinations that must be analyzed, which is calculated as the sum of the combinations of all the partitions. The first steps performed by the heuristic (i.e., variations per service and combinations) define its complexity and as a function of *N* the growth rate of FASTTs is given by Equation (32). Moreover, if all of the partitions have the same size, the growth rate can be expressed, as shown in Equation (33).

$$f(N) = N + (2 \times N \times Ts + N) + \sum_{i=1}^{numPart} (2 \times Ts + 1)^{lenPart_i} \tag{32}$$

$$f(N) = N + (2 \times N \times Ts + N) + numPart \times (2 \times Ts + 1)^{\frac{N}{numPart}} \tag{33}$$

Thus, discarding the low-order terms in Equation (33) the complexity of FASTTs is $\mathcal{O}(2^{\frac{N}{numPart}})$, i.e., the heuristic strategy presents a non-linear growth rate; however, it is much smaller than OPTTs. In this context, in order to assess the difference between the exact and heuristic solutions in terms of the number of combinations and, subsequently, in terms of complexity and execution time, the relative gain parameter (G_T) has been defined, as indicated in Equation (34).

$$G_T = \frac{(2 \times Ts + 1)^N}{\sum_{i=1}^{numPart} (2 \times Ts + 1)^{lenPart_i}} \tag{34}$$

The G_T parameter only considers the most dominant terms of Equations (30) and (32) and allows an estimate of how many times less-complex or how many times faster the heuristic solution is, as compared with the optimal. Figure 10 shows growth rate for OPTTs and FASTTs, considering for this latter $numPart = 2$ and $numPart = 3$ partitions. This example reveals a considerable reduction in the size of the problem (for $N > 2$, because no partition is performed when $N = 1$) when compared to OPTTs. For example, for $N = 10$ services the exact algorithm produces exactly 282.475.249 combinations, while, with $numPart = 2$, this number is reduced to 33,614 combinations ($G_T \simeq 8403$), and with $numPart = 3$ the number decreases down to 3087 combinations ($G_T \simeq 91,504$). Assuming a running time of 100 [hours] for OPTTs, FASTTs is able to solve the problem (suboptimal solution) in approximately 43 [seconds] if the $numPart = 2$, and in roughly 4 [seconds] if the $numPart = 3$. The reduction in complexity would not only be reflected in a shorter execution time, but also in a reduction of the computational requirements (processing and memory). Thus, FASTTs could also be deployed at the edge of the network or near to the end user (i.e., at fog computing level).

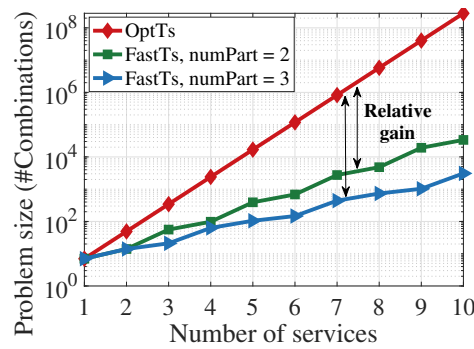


Figure 10. Comparison of problem size between OPTTs and FASTTs. Parameters: $N = \{1, \dots, 10\}$, $T_s = 3$ for all S_k .

8. Energy Management Algorithmic Strategies a Service Function Chains

Figure 11 illustrates an example of the deployment of the algorithmic strategies OPTTs and FASTTs decomposed as VNFs, forming Service Function Chains (SFCs), and coordinated by the NFV Orchestrator (NFVO). This implementation can exploit all the benefits that the NFV realm can offer i.e., the VNFs could be moved, migrated, shared, deployed in parallel mode, or used by other SFCs if necessary. In addition, the operation of algorithmic strategies through SFCs can be further improved by addressing aspects, such as optimal allocation of VNFs or the optimal composition and allocation of SFCs when considering energy efficiency, as discussed in [31,32], respectively. These considerations are beyond the scope of this paper and they can be addressed in future work.

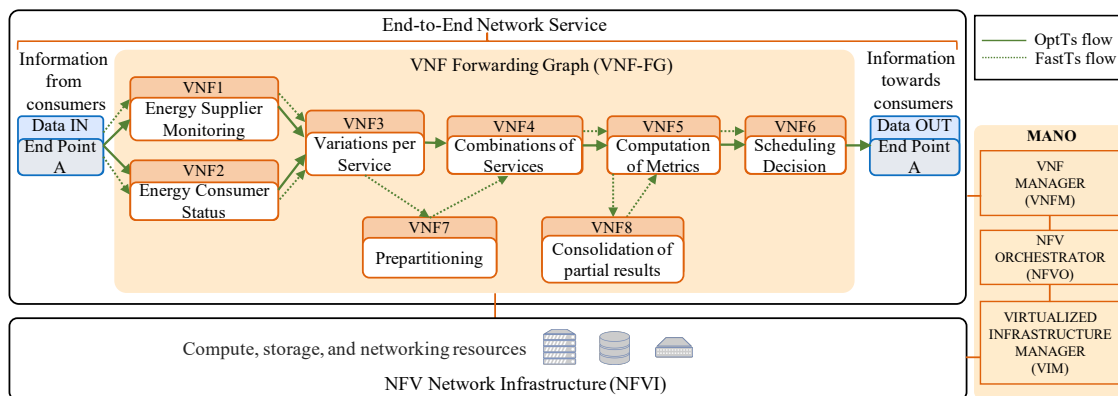


Figure 11. Example of the deployment of OPTTs and FASTTs strategies as SFCs in the NFV-enabled Energy Manager (EM).

Depending on the application scope (i.e., requirements for accuracy, scalability, or speed in the allocation of energy resources) and the amount of available physical resources (NFVI), the EM could choose the execution of OPTTs and/or FASTTs. The SFCs that are associated with the energy management strategies are listed below.

- $SFC_{OPTTs}: VNF_1 - VNF_2 - VNF_3 - VNF_4 - VNF_5 - VNF_6.$
- $SFC_{FASTTs}: VNF_1 - VNF_2 - VNF_3 - VNF_7 - VNF_4 - VNF_5 - VNF_8 - VNF_5 - VNF_6.$

In practical implementations, some VNFs can be concurrently executed, such as in the case of VNF_1 and VNF_2 . In addition, certain VNFs may demand high processing and memory resources for their execution, such as VNF_4 and VNF_8 . For instance, VNF_4 requires high computational capabilities for computing all possible combinations, which can exceed tens or hundreds of millions of combinations for OPTTs. Instead, VNF_8 demands both high memory and processing capabilities to store the partial solutions and consolidate them in order to obtain the final service scheduling.

9. Evaluation

In this section, a performance evaluation of the exact and heuristic strategies is carried out through extensive simulations and according to the metrics P_{RES} , AR , P_{LACK} . The obtained results are compared with traditional scenarios in which no strategy is applied (i.e., when the $T_s^k = 0$). First, the case studies are presented in Section 9.1. Later, the evaluation of the strategies in small and large scale scenarios is discussed in Sections 9.2 and 9.3, respectively. Finally, a summary of the results obtained is presented in Section 9.4. The strategies are implemented while using Matlab (Matlab R2017b) and they were run on a machine with a 3.33 GHz \times 12 cores Intel Core i7 Extreme processor, 24 GB RAM, and 24GB of SWAP memory.

9.1. Case Studies

The strategies are evaluated in six case studies, which are summarized in Table 5. These cases studies correspond to small and large-scale scenarios. The differentiation in the scenario sizes is given by the value of N , because this parameter has a direct impact on complexity, as discussed in Section 5.3 (Equation (30)) and Section 7.2 (Equation (33)). Thus, case studies with $N > 20$ services are categorized as large-scale scenarios; otherwise, they are classified as small scenarios. The first four cases in Table 5 correspond to the small-scale scenarios in which both OPTTs and FASTTs are evaluated. The fifth case in Table 5 is used to analyze FASTTs and the modification of partitions in large-scale scenarios. Finally, although case F is a small-scale scenario, only the evaluation of FASTTs is carried out due to the number of services and T_s^k that is not feasible for OPTTs. In this regard, the consumption information for the HEMS scenario of case F is adapted from [33] and detailed in Table 6.

Table 5. Summary of simulation parameters.

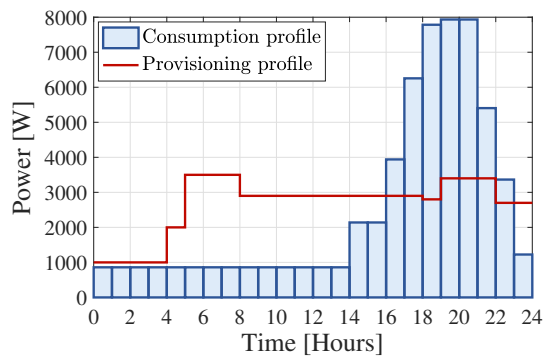
Scenario	P_{ES}	$T_{init}^{P_{ES}}$	m	N	T_{init}^k	T_d^k	P_d^k
Case A: Demands within P_{ES}	3 constant within m	3, 2, 1, for $T_d^k = 1, 2, 3$	given by Equation (26)	9	4, $\forall S_k$	1–3	1, $\forall S_k$
Case B: Demands outside P_{ES}	2 constant within m	T_d^k	given by Equation (26)	8	0, for $S_1 - S_4$ and $T_d^k + m$, for $S_5 - S_8$	1–3	1, $\forall S_k$
Case C: Random Td	2 constant within m	0	6	6	$k - 1$, $\forall S_k$	uniform dist. random [1–3], $\forall S_k$	1, $\forall S_k$
Case D: Random Pd and Td	3 constant within m	0	10	6	$k - 1$, $\forall S_k$	uniform dist. random [1–3], $\forall S_k$	uniform dist. random [1–3], $\forall S_k$
Case E: Large scale Random Pd and Td	3 constant within m	0	50, 100, 1000	50, 100, 1000	$k - 1$, $\forall S_k$	uniform dist. random [1–3], $\forall S_k$	uniform dist. random [1–3], $\forall S_k$
Case F: HEMS	according to Figure 12a and within m	0	24	20	according to Table 6	according to Table 6	according to Table 6

For the case F, the priorities of the energy demands are detailed in Table 6, for the rest of cases there is a single priority level (i.e., $l = 1$). Additionally, for all cases, W is given by Equation (19). In the simulations, four generation and consumption profiles are used. Figure 12 illustrates the profiles used in the evaluation of FASTTs in HEMS, the rest of profiles are depicted in Figure 13. Although in Figure 13, for simplicity in the analysis, a flat-energy profile is used, this does not mean a restriction for algorithmic strategies, which are capable of working with any demand and supply profile. The selected profiles allow for the analysis of the operation of the strategies in: (i) shortage power states, due to a high concentration of demands, as shown in Figure 13a, where the time-shifting of some demands enable full energy utilization; (ii) periodic transitions between periods of scarcity and abundance of power, as shown in Figure 13b, where energy is only harnessed if all demands are shifted in time; (iii) more realistic situations where the services have different (random) values of P_d^k and T_d^k , as depicted in Figure 13c; and, (iv) consumption peaks due to efficient load distribution and that

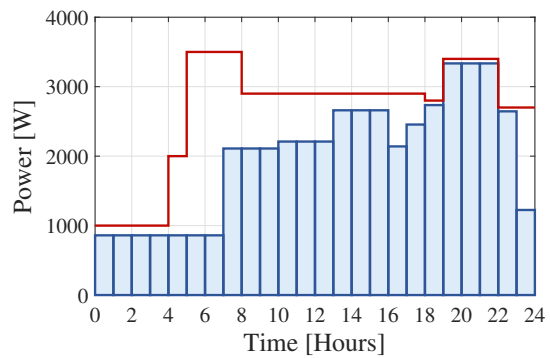
produce an extra effort in generation as shown in Figure 12a. The simulations have been repeated 50 times, when considering a confidence interval of 95%, in order to ensure the quality and stability of results for the random scenarios.

Table 6. Description of Home Energy Management Systems (HEMS) scenario that corresponds to the Case F in Table 5.

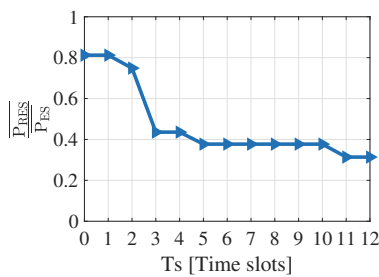
Load Description	Demanded Power [W]	Quantity	l	T_{init}^k	T_d^k
Freezer	210	1	1	0	24
Refrigerator	650	1	1	0	24
Oven	1800	1	3	16	3
Lighting	25	9	1	17	7
TV	140	1	1	18	5
Laptop	90	1	1	17	5
PC	140	1	1	18	6
Vacuum Cleaner	600	1	1	19	3
Water Heater	2000	1	2	17	6
Air-Conditioner	1280	1	2	14	7
Washing Machine	1350	1	3	19	3
Dishwasher	1250	1	3	18	3



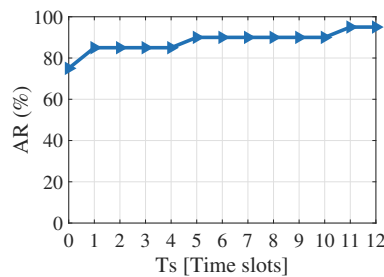
(a) Generation and consumption profiles before the application of FASTTs



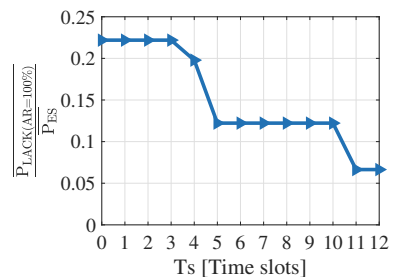
(b) Generation and consumption profiles after the application of FASTTs



(c) HEMS: P_{RES} FASTTs

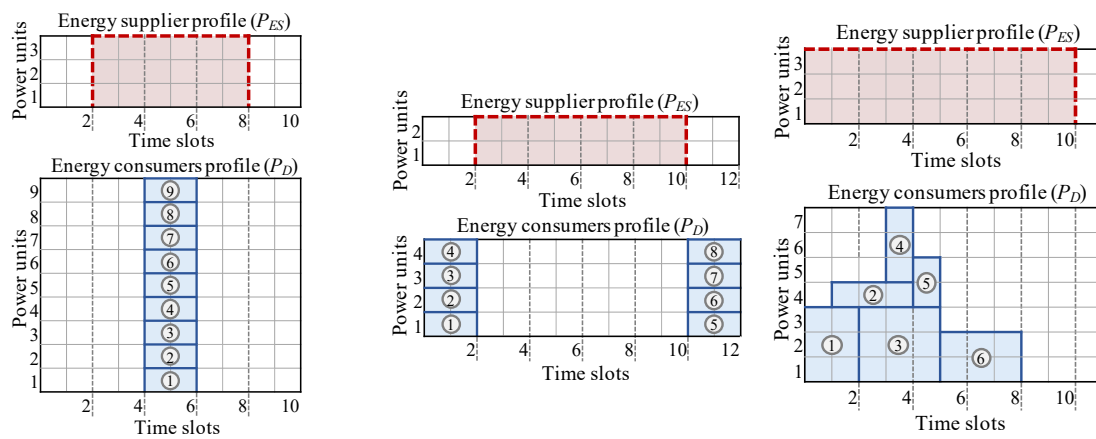


(d) HEMS: AR FASTTs



(e) HEMS: P_{LACK} FASTTs

Figure 12. Performance evaluation of FASTTs in the HEMS scenario. Parameters: $N = 20$ and $numPart = 2$, the rest of parameters according to Case F in Table 5.



(a) Energy and consumption profiles of Case A. Example for $T_d^k = 2$ [time slots], for all S_k . (b) Energy and consumption profiles of Case B. Example for $T_d^k = 2$ [time slots], for all S_k . (c) Example of the energy and consumption profiles for Random Cases (Cases C, D, E).

Figure 13. Energy and consumption profiles of the case studies.

9.2. Analysis for Small-Scale Scenarios

9.2.1. Results in Small-Scale Scenarios

The evaluation of OPTTs and FASTTs is presented in Figures 14 and 15, and it shows that the operation of the strategies contributes to a better use of P_{ES} , which is reflected in a minimization of P_{RES} and $P_{LACK(AR=100\%)}$, and in an increase of AR. In all scenarios, OPTTs and FASTTs improve the performance of the system by enabling the processing of services that could not be executed under normal conditions ($T_s^k = 0$). As the value of time-shifting increases, the system has more facility to move the workloads and, progressively, it is able to use all P_{ES} (i.e., $P_{RES} = 0$ and $AR = 100\%$). Table 7 summarizes the improvements (AR gain) achieved by OPTTs and FASTTs. The most notable improvement is obtained in Case B, in which, under normal conditions, no service could be executed due to lack of energy. In the random cases, instead, lower AR gains are reached when compared to Case A and Case B, because the conditions of the scenarios make it such that it is not possible consume all P_{ES} or reach an $AR = 100\%$. An example of the combinations processed and the computational resources needed for the execution of the algorithmic strategies is presented in Figure 16 for Case B. In the scope of simulation, Figure 16 shows that *FastTs* requires approximately 4% RAM usage and 44% CPU usage when compared to capacity used by *OptTs* to process 100% of energy demands.

Table 7. AR metric for small-case scenarios.

Scenario	No Strategy	OptTs		FastTs	
	Initial AR	Final AR	AR Gain	Final AR	AR Gain
Case A	33.33%	100%	66.67%	92.59%	59.26%
Case B	0%	100%	100%	100%	100%
Case C	74%	93%	19%	92%	18%
Case D	67%	95%	28%	94%	27%
Case F	75%	–	–	95%	20%

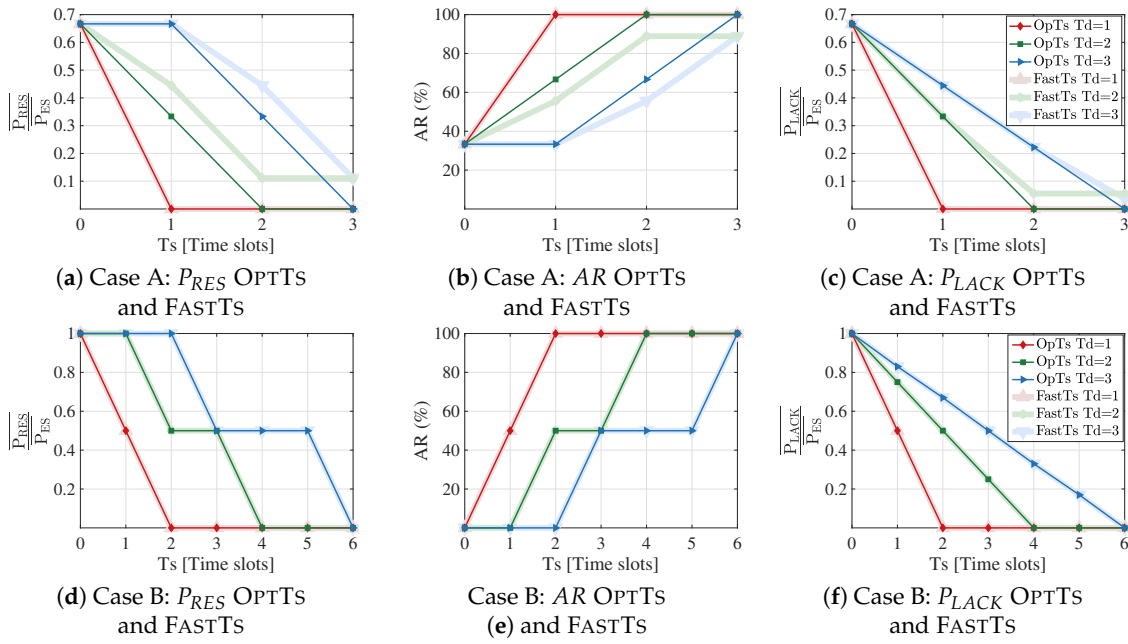


Figure 14. Performance evaluation of the OPTTs and FASTTs for Case A and Case B. **Parameters:** Based on Table 5.

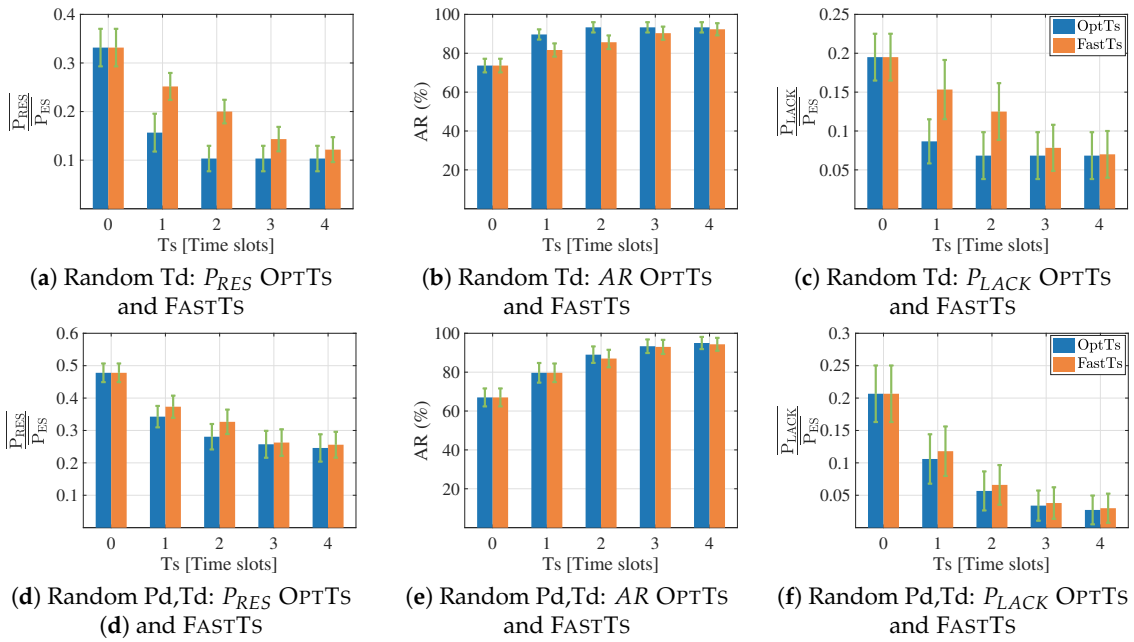


Figure 15. Performance evaluation of OPTTs and FASTTs for Random Cases in small-scale scenarios. **Parameters:** $N = 6$ and $numPart = 2$ for all cases, the rest of parameters according to Case C and Case D of Table 5.

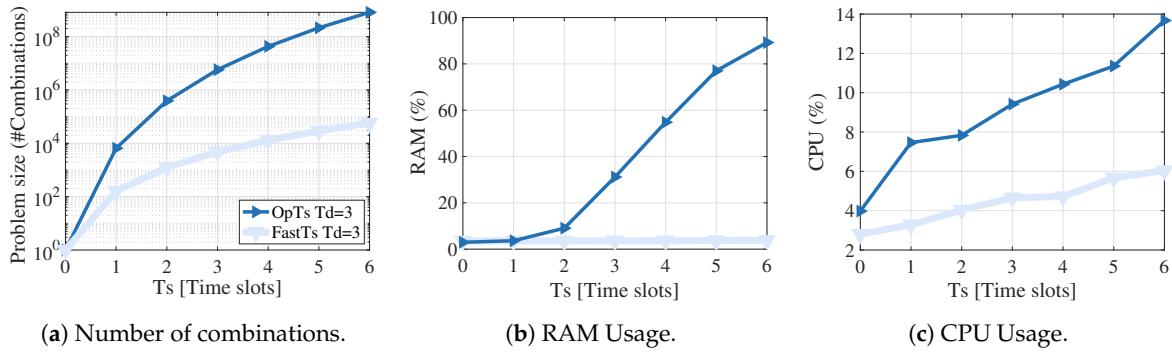


Figure 16. Combinations processed and computational resources used to execute OPTTs and FASTTs within the scope of simulation. Parameters: According to the Case B in Table 5, with $numPart = 2$ for FASTTs.

Regarding the application of FastTs in the HEMS scenario, Figure 12b shows that, even though the improvement in the AR metric is only 20%, as shown in Table 7, the application of the adaptive management model produces a reduction of the peak power. This operational feature in energy systems is essential to allow uniform power distribution among users and avoid extra generation from non-renewable sources. The evaluation in the HEMS scenario is also useful to verify that the FastTs strategy not only allows to exceed the limit in the number of services processed by OPTTs, but also the maximum values of T_s^k . In case F, a $T_s^k = 12$ h enables the processing of 95% of energy demands.

9.2.2. Comparison between OPTTs and FASTTs

To evaluate the difference between strategies, the criterion of approximation ratio (ρ) has been adopted [28]. This parameter estimates how many times bigger the approximate solution is compared to the exact result, and it is typically defined as the ratio between the suboptimal and optimal solutions [28]. However, because P_{RES} and P_{LACK} may be zero, its definition has been adapted to the conditions of the strategies, as shown below:

$$\rho = 1 - \frac{1}{T_s^k} \sum_{i=1}^{T_s^k} \frac{|OptTs_i - FastTs_i|}{disOptTs_i} \tag{35}$$

$$disOptTs_i = \begin{cases} 1 & \text{if } OptTs_i = OptTs_0, \\ |OptTs_i - OptTs_0| & \text{if } OptTs_i \neq OptTs_0. \end{cases} \tag{36}$$

where, the first term in Equation (35) represents the exact solution, while the second term corresponds to the mean absolute error of all time-shifting values, except for $T_s^k = 0$. As shown in Equation (35), each absolute error i is weighted to the $disOptTs_i$ parameter (Equation (36)), which corresponds to the maximum distance between the optimal value ($OptTs_i$) and the initial conditions of the system ($OptTs_0$, when $T_s^k = 0$) in order to obtain the proportional error of each time-shifting i . The ρ parameter ranges from 0 (no improvements in the system) to 1 (when FASTTs and OPTTs produce the same results); an intermediate value ($0 < \rho < 1$) represents the similarity or closeness factor to the optimal solution ($FASTTs = OPTTs \times \rho$). For a better understanding of the computation of the ρ factor, Figure 17 presents an example for metric AR of Case A. The computations of the absolute errors that correspond to $T_s^k = \{1, \dots, 3\}$ and the resulting ρ_{AR} are presented in Equation (37).

$$\rho_{AR} = 1 - \frac{1}{3} \times \left(\frac{66.7 - 55.6}{66.7 - 33.3} + \frac{100 - 88.9}{100 - 33.3} + \frac{100 - 88.9}{100 - 33.3} \right) = 0.78 \tag{37}$$

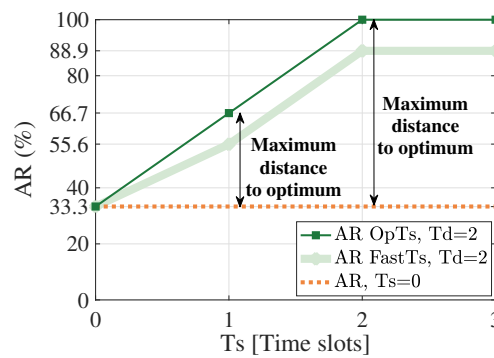


Figure 17. Example to evaluate the difference between OPTTs and FASTTs. Parameters: *Case A*, AR , and $T_d^k = 2$ [time slots] for all S_k .

The result of Equation (37) shows that the heuristic solution is similar to the optimal solution in a factor equal to 0.78 (78% similarity), or that FASTTs can produce a solution that is within ~ 1.3 times the optimal result. The ρ factors for small-scale scenarios are summarized in Table 8 and reveal that FASTTs produces not only near-optimal or optimal solutions, but also a stable performance. Even in the worst case (i.e., $\rho_{P_{LACK}} = 0.71$), the heuristic solution is within only ~ 1.4 times the optimal result. In addition, for random scenarios we observe that the best results are achieved for *Case D*. Therefore, the conditions of this case study are taken as the baseline for the evaluation of FASTTs for larger N values.

Table 8. Approximation ratio for small-scale scenarios.

ρ	Case A			Case B			Case C	Case D
	Td1	Td2	Td3	Td1	Td2	Td3	\bar{Td}	\bar{Td}, \bar{Pd}
$\rho_{P_{RES}}$	1	0.78	0.83	1	1	1	0.70	0.87
ρ_{AR}	1	0.78	0.83	1	1	1	0.73	0.97
$\rho_{P_{LACK}}$	1	0.92	0.98	1	1	1	0.71	0.95

9.2.3. Running Time Analysis

The simulations results show that FASTTs is hundreds or thousands of times faster than OPTTs, and the specific difference or real gain in time (G_R) between both strategies not only depends on the number of combinations to be processed, but also on the problem instance and the computer equipment used. Table 9 summarizes the average running and the G_R factor for all small-scale scenarios.

Table 9. Average running time in small-scale scenarios.

Cases	Running Time OPTTs [s]	Running Time FASTTs [s]	G_R
<i>Case A</i>	5.04×10^5	1.00×10^2	$\sim 5000 \times$
<i>Case B</i>	9.72×10^5	2.20×10^2	$\sim 4400 \times$
Random scenarios, 50 iterations	1.87×10^4	8×10^1	$\sim 230 \times$
<i>Case F</i>	-	3.53×10^0	-

9.3. Analysis for Large-Scale Scenarios

A set of simulations for large N values have been conducted in order to demonstrate the scalability of FASTTs. These tests have been limited up to $N = 1000$; however, this is by no means a restriction of the applicability of FASTTs to larger scenarios.

Results Large-Scale Scenarios

For these scenarios, the number of partitions has been calculated so that the size of each partition is smaller than four services (e.g., $numPart = 15$ partitions for $N = 50$ services). This criterion has been considered based on the high quality results and ρ factor obtained for *Case D* (Table 8). For practical reasons, only the evaluation of FASTTs for $N = 100$ is shown in Figure 18, and the rest of metrics for the different N values and partitions schemes are summarized in Table 10. The simulation results demonstrate that FASTTs enables a better P_{ES} utilization (decrease of P_{RES} and P_{LACK} , and increase of AR), and in all cases, it produces improvements greater than 13%. In addition, the metrics obtained for the different N values (Table 10) are very similar to each other because of the use of the same power generation and consumption profiles (Figure 13c), which can be used in order to estimate the performance metrics for larger scenarios ($N > 1000$) as long as the profiles of Figure 13c are used.

The results presented in Table 10 also show that an increase in the number of partitions can significantly reduce execution time, keeping quality in solutions. This reduction is mainly observed for the smaller scenarios (e.g., $N = 50$), because the total execution time grows linearly based on the number of partitions processed. To enhance the running time for larger scenarios, a method that could be complemented to FASTTs is the parallel processing of partitions (each partition solved in a different core), so the total execution time would decrease proportionally as the number of cores increases.

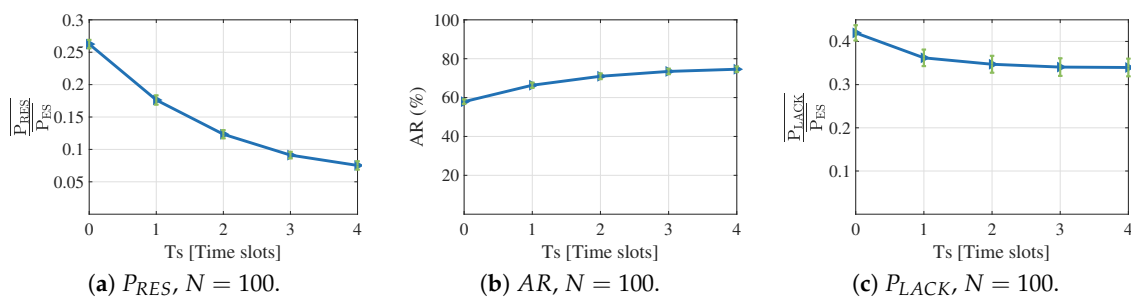


Figure 18. Example of performance evaluation of FASTTs for a random case in a large-scale scenario. **Parameters:** According to Case E of Table 5, for $N = 100$ and with $numPart = 30$.

Table 10. Average values of metrics and running time for FASTTs in large-scale scenarios for $T_s = 4$.

Parameters		$\overline{\frac{P_{RES}}{P_{ES}}}$		$\overline{\frac{P_{LACK}}{P_{ES}}}$		AR (%)		GR (%)	Running Time, 50 Iterations [s]
N	numPart	No Strategy	FASTTs	No Strategy	FASTTs	No Strategy	FASTTs	FASTTs	FASTTs
50	15		0.08		0.34		74.56	15.36	2.13×10^3
	20	0.26	0.09	0.42	0.35	59.20	72.72	13.52	3.04×10^2
	25		0.10		0.36		72.66	13.46	5.84×10^1
100	30		0.08		0.34		74.55	16.65	5.16×10^3
	40	0.26	0.09	0.42	0.35	57.90	73.02	15.12	8.01×10^2
	50		0.10		0.36		72.68	14.78	1.67×10^2
1000	300		0.07		0.35		74.77	15.97	1.67×10^5
	400	0.26	0.08	0.43	0.35	58.80	73.99	15.19	4.33×10^4
	500		0.10		0.36		72.89	14.09	1.32×10^4

9.4. Summary of Results

The simulation results demonstrate that the exact and heuristic strategies enable the efficient (optimal) use of P_{ES} , which is reflected in the decrease (minimization) of P_{RES} and P_{LACK} and in the increase of AR as the time-shifting value increases. From the analyzed scenarios, we have verified that OPTTs and FASTTs offer improvements that range from small increases in consumption and the processing of services to the full use of available power and, consequently, the processing of all services, as shown in *Case B*, where the final $AR = 100\%$. In addition, the evaluation of strategies

demonstrates that an appropriate value of T_s^k for the full utilization of P_{ES} is the mean size of T_d^k , since this value represents the complete displacement of the service. However, when considering that the computational complexity depends on T_s^k , as shown in Equation (30), smaller values of T_s^k equal to $T_s^k = 1$ [time slot] or $T_s^k = 2$ [time slots] can be used for practical applications. The simulations reveal that the greatest improvements are obtained with small T_s^k values. For example, for all random scenarios, with $T_s^k = 1$ [time slot], the system reaches improvements over the 50%, with $T_s^k = 2$ [time slots] the improvements are 30% and, as the value of T_s^k increases, the percentage of improvement obtained decreases. Thus, this condition demonstrates that, for real implementations, large intervals of T_s^k are not required to achieve notable improvements in energy consumption.

Regarding the heuristic strategy, the numerical results of the metrics P_{RES} , AR , P_{LACK} and ρ (Table 8) indicate that FASTTs is able to produce near-optimal or optimal solutions (e.g., in Case B) at a very low complexity cost (Table 9). In small scenarios, FASTTs offers solutions, which, in the worst case, are within only ~ 1.4 times the optimal result and are at least three orders of magnitude faster than OPTTs. In addition, in large-scale scenarios, we have verified that FASTTs improves the energy utilization (decrease of P_{RES} and P_{LACK} and increase of AR) and offers improvements of at least 13% for all cases, as long as the strategy considers partitions smaller than six services to guarantee its performance. The simulation results reveal that a controlled increase in the number of partitions can produce high-quality solutions but with a reduced running time, which, in all cases, is of at least an order of magnitude lower than the original partition scheme.

Finally, as a final result of the whole paper, Table 11 summarizes the requirements, proposed solutions, and algorithmic strategies in order to carry out an ICT-enabled ecosystem for adaptive energy management restricted to availability.

Table 11. Summary of requirements and proposed solution for adaptive energy management in an Internet of Things (IoT) and NFV enabled ecosystem.

Requirement	Proposed Solution	Comments
Adaptive consumption constrained to availability	<ul style="list-style-type: none"> - Use of management mechanisms to adapt consumption to generation: (i) time-shifting capabilities on energy demands; (ii) prioritization of energy supply according to the priority level of the energy demand; and, (iii) the rejection of energy demands if available supply is insufficient to cover all consumption. - Implementation of the management mechanisms through algorithms, which must be deployed in infrastructures with high computational capacity due to the complexity involved in the adaptive energy management process. 	The adaptation process must be carried out constantly, and one or all of the proposed management mechanisms can be used for this purpose.
Consumer-side participation	<ul style="list-style-type: none"> - Availability of user-side connectivity and the use of IoT technologies to participate in the energy management ecosystem. - Interaction with the ES (specifically with the EM) through a dynamic and programmable underlying network (e.g., an SDN-based solution). This network must guarantee a secure and reliable exchange of consumption data and instructions between ES and ECs. 	The participation of the ECs in the energy management process is established through contracts with the ES. These agreements detail all of the technical aspects that are related to the energy demands (e.g., priority) and the possible incentives (e.g., reduce bills) that the ECs can receive by modifying consumption patterns to the availability.
Dynamic behavior and scalable computational capacity for adaptive energy management	<ul style="list-style-type: none"> - Dynamic behavior by implementing algorithmic management strategies through VFNs forming SFCs. These VFNs can be created, activated, or modified on demand according to the conditions of consumption and generation. - Scalable computational capacity (mainly in terms of processing power and memory) available when NFV technology (i.e., algorithmic strategies) is deployed on cloud computing infrastructures. - An NFV-enabled solution also provides the management entities (e.g., the NFVO) so that the ES, EM, and ECs work orchestrated. 	The NFV technology could be deployed on the operations centers available in current smart grids. Additionally, the communications standards, such as SCADA, could migrate to programmable solutions such as SDN. Through these updates, current energy systems could begin their transition to adaptive consumption management.

Table 11. Cont.

Requirement	Proposed Solution	Comments
The primary use of renewable energy	<ul style="list-style-type: none"> - Specification of the contribution of renewable energy sources in the total supply. In the characterization of the ES in Equation (3) is considered to be the factor w_R that could be set to the value of 1 to promote the exclusive use of green energy. - Dynamic consumption, according to the sporadic energy production from green sources such as solar or wind, by encouraging early or delayed consumption according to availability. To leverage green energy production capacity, all proposed management mechanisms can be used; however, the use of time-shifting capabilities is preferred to avoid the rejection of demands. 	To ensure energy sustainability, future energy, and communication systems demand the progressive use of green energy sources. Additionally, the generation from renewable energy sources is a key enabler in the road to a true IoE. In this regard, adaptive energy management solutions are essential to delivering reliable and high-performance energy systems.
Adaptive energy management with applicability to small-scale scenarios	<ul style="list-style-type: none"> - Use of the algorithmic strategy OPTTs deployed on an NFV and IoT enabled adaptive energy management ecosystem. The OPTTs solution produces the optimal utilization of available supply but at the cost of high computational resources and running time. - Use of the algorithmic strategy FASTTs deployed on an NFV and IoT enabled adaptive energy management ecosystem. The FASTTs solution produces suboptimal utilization of available supply, but with lower resource utilization and running time, as compared to OPTTs. 	The optimal solution OPTTs is implemented using a brute-force search method and it has an exponential complexity that depends on the maximum values of N and T_s^k , as shown in Equation (30). Reported limits in the simulation scope are $N = 9$ services and $T_s^k = 6$ time slots.
Adaptive energy management with applicability to large-scale scenarios	<ul style="list-style-type: none"> - Use of the algorithmic strategy FASTTs deployed on an NFV and IoT enabled adaptive energy management ecosystem. The performance and the quality of the solutions of FASTTs can be modified by varying the number of partitions used. 	The heuristic solution FASTTs is implemented while using a divide-and-conquer approach. The complexity of FastTs in non-linear, but the growth rate is much lower than OPTTs. Reported limits on simulation scope are $N = 1000$ services and $T_s^k = 12$ time slots.

10. Conclusions

This paper studies and formulates an NFV-Enabled energy management model for renewable and non-renewable energy sources in order to analyze the requirements and concerns about service scheduling algorithms. Strategies, OPTTs, and FASTTs to enable efficient utilization of available power are proposed for small and large-scale IoT scenarios. The evaluation results demonstrate that the proposed algorithmic solutions that are supported by NFV result in improvements that range from small increases in the consumption and processing of services up to the full utilization of P_{ES} and the processing of all demands (e.g., in *Case B*, $AR = 100\%$). This better P_{ES} utilization is reflected in a decrease of P_{RES} and P_{LACK} and in an increase of AR with respect to the non-application of the energy management solution (i.e., $T_s^k = 0$); specifically, we have verified that the greatest improvements in energy consumption are achieved with small T_s^k values (e.g., $T_s^k = 1$).

In terms of complexity, it is verified that the P_{RES} minimization problem, when considering management strategies, such as the optimal workload scheduling and service rejection, is \mathcal{NP} -hard and that its optimal solution, OPTTs, has an exponential complexity that depends on the values of T_s^k and N (Equation (30)). This demands a robust and scalable computing infrastructure like the one offered by NFV. In this context, in order to propose less complex and more scalable solutions, some heuristic approaches have been analyzed, and a pre-partition strategy, FASTTs, has been evaluated. Regarding the comparison between the exact and the heuristic solutions, in small scenarios, FASTTs is at least three orders of magnitude faster than OPTTs, producing similar or equal results (e.g., in *Case B*). In the worst case, the heuristics solution is within only ~ 1.4 times the optimal result. In addition, in large-scale scenarios, the evaluation of FASTTs, for different services and partitioning schemes also demonstrates improvements over 13% in all cases.

Future work will be devoted to the analysis of the incorporation of an energy storage component for ensuring the execution of CS, the study of the service prioritization capability in the context of network slicing, and the development of more scalable and faster approaches that are based on the rest of the heuristic strategies presented in Table 4.

Author Contributions: Conceptualization, C.T. and X.H.; methodology, C.T. and X.H.; software, C.T. and X.H.; validation, C.T. and X.H.; formal analysis, C.T. and X.H.; investigation, C.T. and X.H.; resources, C.T. and X.H.; data curation, C.T. and X.H.; writing—original draft preparation, C.T. and X.H.; writing—review and editing, C.T. and X.H.; visualization, C.T. and X.H.; supervision, C.T. and X.H.; project administration, C.T. and X.H.; funding acquisition, X.H. and C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Ministerio de Ciencia e Innovación of the Spanish Government under project PID2019-108713RB-C51.

Acknowledgments: Christian Tipantuña acknowledges the support from Escuela Politécnica Nacional (EPN) and from Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) for his doctoral studies at Universitat Politècnica de Catalunya (UPC).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Global e-Sustainability Initiative. *SMARTer2030, ICT Solutions for 21st Century Challenges*; GeSI, Accenture Strategy: Brussels, Belgium, 2015.
2. Lima, J.M. *Data centres of the World Will Consume 1/5 of Earth's Power by 2025*; Technical Report; Euromoney Institutional Investor PLC: London, UK, 2017.
3. Carrasco, J.M.; Garcia Franquelo, L.; Bialasiewicz, J.T.; Galván, E.; Portillo Guisado, R.C.; Martín Prats, M.d.l.Á.; León, J.I.; Moreno-Alfonso, N. Power-electronic systems for the grid integration of renewable energy sources: A survey. *IEEE Trans. Ind. Electron.* **2006**, *53*, 1002–1016. [CrossRef]
4. Medina, J.; Muller, N.; Roytelman, I. Demand response and distribution grid operations: Opportunities and challenges. *IEEE Trans. Smart Grid* **2010**, *1*, 193–198. [CrossRef]
5. Tipantuña, C.; Hesselbach, X. NFV/SDN Enabled Architecture for Efficient Adaptive Management of Renewable and Non-renewable Energy. *IEEE Open J. Commun. Soc.* **2020**, *1*, 357–380. [CrossRef]
6. ETSI. Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges & Call for Action. *ETSI White Pap.* **2012**, 1–16. [CrossRef]
7. Shakerighadi, B.; Anvari-Moghaddam, A.; Vasquez, J.C.; Guerrero, J.M. Internet of things for modern energy systems: State-of-the-art, challenges, and open issues. *Energies* **2018**, *11*, 1252. [CrossRef]
8. Cisco, V. Cisco visual networking index: Forecast and trends, 2017–2022. *White Pap.* **2018**, *1*, 1.
9. Zhou, K.; Yang, S.; Shao, Z. Energy internet: The business perspective. *Appl. Energy* **2016**, *178*, 212–222. [CrossRef]
10. Huang, B.; Bai, X.; Zhou, Z.; Cui, Q.; Zhu, D.; Hu, R. Energy informatics: Fundamentals and standardization. *ICT Express* **2017**, *3*, 76–80. [CrossRef]
11. Ghatikar, G.; Ganti, V.; Matson, N. Demand Response Opportunities and Enabling Technologies for Data Centers: Findings from Field Studies. *Lawrence Berkeley Natl. Lab.* **2012**, LBNL-5763E, 1–48.
12. Wei, M.; Hong, S.H.; Alam, M. An IoT-based energy-management platform for industrial facilities. *Appl. Energy* **2016**, *164*, 607–619. [CrossRef]
13. AlFaris, F.; Juaidi, A.; Manzano-Agugliaro, F. Intelligent homes' technologies to optimize the energy performance for the net zero energy home. *Energy Build.* **2017**, *153*, 262–274. [CrossRef]
14. Alliance, O. *OpenADR 2.0 Profile Specification B Profile*; Technical Report; OpenADR Alliance: Morgan Hill, CA, USA, 2013.
15. Basmadjian, R.; Botero, J.F.; Giuliani, G.; Hesselbach, X.; Klingert, S.; De, H. Making Data Centres Fit for Demand Response: Introducing GreenSDA and GreenSLA Contracts. *IEEE Trans. Smart Grid* **2017**, *9*, 3453–3464. [CrossRef]
16. Niedermeier, M.; De Meer, H. Constructing dependable smart grid networks using network functions virtualization. *J. Netw. Syst. Manag.* **2016**, *24*, 449–469. [CrossRef]
17. Mijumbi, R. On the energy efficiency prospects of network function virtualization. *arXiv* **2015**, arXiv:1512.00215.
18. GreenTouch. G.W.A.T.T. (Global What if Analyzer of NeTwork Energy ConsumpTion). Online Web Tool to Provide an End-to-End View of the GreenTouch Portfolio of Technologies and Solutions on Network Energy Consumption. 2015. Available online: <http://alu-greentouch-dev.appspot.com/intro/1> (accessed on 1 September 2020).

19. Li, X.; Ferdous, R.; Chiasserini, C.F.; Casetti, C.E.; Moscatelli, F.; Landi, G.; Casellas, R.; Sakaguchi, K.; Chundrigar, S.B.; Vilalta, R.; et al. Novel Resource and Energy Management for 5G integrated backhaul/fronthaul (5G-Crosshaul). In Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017; pp. 778–784.
20. Al-Quzweeni, A.N.; Lawey, A.Q.; Elgorashi, T.E.; Elmoghani, J.M. Optimized energy aware 5G network function virtualization. *IEEE Access* **2019**, *7*, 44939–44958. [[CrossRef](#)]
21. Tipantuña, C.; Hesselbach, X.; Sánchez-Aguero, V.; Valera, F.; Vidal, I.; Nogales, B. An NFV-Based Energy Scheduling Algorithm for a 5G Enabled Fleet of Programmable Unmanned Aerial Vehicles. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 4734821. [[CrossRef](#)]
22. Tsoukalas, L.; Gao, R. From smart grids to an energy internet: Assumptions, architectures and requirements. In Proceedings of the 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, 6–9 April 2008; pp. 94–98.
23. Strasser, T.; Andrés, F.; Kathan, J.; Cecati, C.; Buccella, C.; Siano, P.; Leitao, P.; Zhabelova, G.; Vyatkin, V.; Vrba, P.; et al. A review of architectures and concepts for intelligence in future electric energy systems. *IEEE Trans. Ind. Electron.* **2014**, *62*, 2424–2438. [[CrossRef](#)]
24. ETSI. Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. *ETSI Netw. Funct. Virtualisat. Ind. Specif. Group GS NFV-EVE* **2015**, *5*, V1.
25. Keles, C.; Kaygusuz, A.; Alagoz, B.B. Multi-source energy mixing by time rate multiple PWM for microgrids. In Proceedings of the 4th International Istanbul Smart Grid Congress and Fair, ICSG 2016, Istanbul, Turkey, 20–21 April 2016. [[CrossRef](#)]
26. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st ed.; WH Free. Co.: San Francisco, CA, USA, 1979; p. 340.
27. Pisinger, D. *Algorithms for Knapsack Problems*; Dept. of Computer Science, University of Copenhagen: København, Denmark, 1995.
28. Vazirani, V.V. *Approximation Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
29. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press: Cambridge, MA, USA, 2009.
30. Shao, B.B.; Rao, H.R. A parallel hypercube algorithm for discrete resource allocation problems. *IEEE Trans. Syst. Man Cybern. Part Syst. Hum.* **2006**, *36*, 233–242. [[CrossRef](#)]
31. Tomassilli, A.; Huin, N.; Giroire, F.; Jaumard, B. *Energy-Efficient Service Chains with Network Function Virtualization*; HAL: Houston, TX, USA, 2016.
32. Kim, S.; Han, Y.; Park, S. An energy-aware service function chaining and reconfiguration algorithm in NFV. In Proceedings of the 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W), Augsburg, Germany, 12–16 September 2016; pp. 54–59.
33. Ayan, O.; Turkey, B. Energy Management Algorithm for Peak Demand Reduction. In Proceedings of the 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA), Bourgas, Bulgaria, 3–6 June 2018; pp. 1–4.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).