

Received July 6, 2021, accepted August 4, 2021, date of publication September 3, 2021, date of current version September 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110246

Heuristic Strategies for NFV-Enabled Renewable and Non-Renewable Energy Management in the Future IoT World

CHRISTIAN TIPANTUÑA^{1,2}, (Member, IEEE), XAVIER HESSELBACH², (Senior Member, IEEE), AND WALTER UNGER³

¹Department of Electronics, Telecommunications and Computer Networks, Escuela Politécnica Nacional, Quito 170517, Ecuador

²Department of Network Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

³Department of Computer Science, RWTH Aachen University, 52056 Aachen, Germany

Corresponding author: Christian Tipantuña (christian.jose.tipantuna@upc.edu)

This work was supported by the Ministerio de Ciencia e Innovación of the Spanish Government under Project PID2019-108713RB-C51.

The work of Christian Tipantuña was supported in part by the Escuela Politécnica Nacional and in part by Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT).

ABSTRACT The ever-growing energy demand and the CO₂ emissions caused by energy production and consumption have become critical concerns worldwide and drive new energy management and consumption schemes. In this regard, energy systems that promote green energy, customer-side participation enabled by the Internet of Things (IoT) technologies, and adaptive consumption mechanisms implemented on advanced communications technologies such as the Network Function Virtualization (NFV) emerge as sustainable and de-carbonized alternatives. On these modern schemes, diverse management algorithmic solutions can be deployed to promote the interaction between generation and consumption sides and optimize the use of available energy either from renewable or non-renewable sources. However, existing literature shows that management solutions considering features such as the dynamic nature of renewable energy generation, prioritization in energy provisioning if needed, and time-shifting capabilities to adapt the workloads to energy availability present a complexity NP-Hard. This condition imposes limits on applicability to a small number of energy demands or time-shifting values. Therefore, faster and less complex adaptive energy management approaches are needed. To meet these requirements, this paper proposes three heuristic strategies: a greedy strategy (GreedyTs), a genetic-algorithm-based solution (GATs), and a dynamic programming approach (DPTs) that, when deployed at the NFV domain, seeks the best possible scheduling of demands that lead to efficient energy utilization. The performance of the algorithmic strategies is validated through extensive simulations in several scenarios, demonstrating improvements in energy consumption and processing of demands. Additionally, simulation results reveal that the heuristic approaches produce high-quality solutions close to the optimal while executing among two and seven orders of magnitude faster and with applicability to scenarios with thousands and hundreds of thousands of energy demands.

INDEX TERMS Energy efficiency, energy management, demand response, NFV, IoT, power consumption, workload scheduling, genetic algorithm, greedy algorithm, dynamic programming, renewable energy.

I. INTRODUCTION

Traditionally, worldwide energy provisioning has been dominated by fossil fuels such as petroleum, coal, and natural gas, resulting in an increase in CO₂ emissions and global warming [1]. In the near future, this dependence could potentially lead to an energy crisis due to the risk of depleting fossil fuels, the infeasibility of meeting the ever-growing

energy demand, the increased cost of energy production, and the high impact on climate and the environment. To ensure human society's development, a zero-carbon alternative is the use of green energy from renewable energy sources, like solar or wind, into the world energy matrix. It is conjectured that more than 50% of projected global energy needs can be satisfied by utilizing the Earth's green energy [2]. Then, the evolution toward ecosystems completely powered by renewable energy is seen as a very promising approach to tackle sustainability issues and reduce the carbon footprint.

The associate editor coordinating the review of this manuscript and approving it for publication was Ricardo de Castro¹.

In this context, major IT providers, including Google [3], Microsoft [4], and Apple [5], as well as mobile network operators [6], are already promoting all computing and networking infrastructure being fully supported by renewable energy. Despite the multiple benefits that green sources can offer, their intermittent behavior, added to the inefficient use of the generated energy (also from non-renewable sources), requires the development of consumption management solutions to maintain power reliability, continuity, and quality. In this regard, consumer-side participation and Demand-Response (DR) schemes are effective solutions to adapt the energy consumption patterns to energy supply dynamically. Adaptive DR schemes or programs based on agreements between the Energy Supplier (ES) and the Energy Consumers (ECs) promote the exchange of indications and requests between parties to adapt consumption in response to changes in energy generation with the motivation of energy bill reduction and/or free usage periods for end users [7].

Technically, an adaptive, environmentally friendly DR energy management system needs robust and scalable Information and Communications Technologies (ICT) both to facilitate the interaction between the ES and the ECs and to deploy the management strategies that lead to efficient energy utilization [8]. In this context, modern energy systems can incorporate diverse Internet of Things (IoT) technologies to extend energy management to the end user and improve control, management, and monitoring tasks. Currently, there is an important proliferation of Internet connectivity worldwide, and some studies estimate that by 2022 the number of IoT devices (energy consumers) will surpass 28.5 billion (i.e., 10 billion more devices than in 2017) [9]. This fact reveals both the feasibility of implementing efficient energy management solutions with consumer-side participation and the application scope (e.g., energy management in homes, in buildings, or even for entire countries).

A critical feature of adaptive energy consumption systems is the computational capacity (mainly in terms of memory and processing power) needed to execute the management strategies carried out through algorithmic solutions. Traditionally, the ES has used operations centers and, recently, Data Centers (DCs) infrastructures in which a variety of management policies have been implemented [10]. However, the dynamic behavior of the generation and consumption ecosystem has prompted these DCs to evolve to cloud computing infrastructures in which sophisticated technologies such as Network Function Virtualization (NFV) can be deployed [8]. NFV technology provides to energy systems programmability by facilitating the deployment of different management strategies through virtual components, called Virtual Network Functions (VNFs), forming Service Functions Chains (SFCs), which can be activated or modified when necessary while executing on generic hardware [11]. NFV also provides energy systems, the Management and Orchestration (MANO) entities, to coordinate all actions between the ES and the ECs and the underlying network (e.g., a Software Defined Networking (SDN) solution). Thus,

an NFV-based approach can manage computational resources on demand, which allows adaptive energy management to be scalable and be carried out in reduced execution times, especially for delay-sensitive applications.

An important consideration in the use of NFV technology is the fact that in an NFV environment, there are many software components (e.g., virtual machines, containers, hypervisors, and software of networking components) that evolve and need to be updated or modified over time. The correct operation of these components impacts the performance of the deployed network service (e.g., as the network service for energy management presented in this paper). Then, to avoid disruptions on services and also to warranty availability and reliability, alternatives such as redundancy of elements and entities (physical and logical), as well as the use of protection and restoration schemes are effective methods that must be considered in the deployment and execution of a network service in the NFV realm.

Although an NFV-enabled energy management solution, deployed on cloud computing infrastructures, can offer programmability, scalability, and high computational capacity, the inherent complexity of the algorithmic management solutions to be executed in the NFV domain is still an important, open issue. Specifically, adaptive energy management that produces the optimal energy utilization by employing mechanisms such as the time-shifting capability to adapt the consumption to availability has proven to have an NP-Hard complexity [8]. This complexity level constrains the applicability of consumption management to small-scale scenarios and a few time-shifting values [12].

Considering that an NFV-enabled and environmentally friendly adaptive energy consumption solution can increase overall energy utilization and prevent energy outages while producing cost reductions for the ES and ECs, developing less complex and faster management strategies is of the utmost importance. This is the main contribution of this work. Specifically, this paper proposes heuristic strategies that require less computational capacity and running time compared to that demanded by the optimal solution [12]. The reduced complexity delivered from the heuristics enables energy management for large-scale scenarios (e.g., for thousands or hundreds of thousands of energy demands) and potential deployment not only on cloud computing infrastructures but also on edge or fog computing infrastructures, and even on Home Energy Management Systems (HEMS).

A. CONTRIBUTION

This paper starts describing the NFV-enabled ecosystem for adaptive energy management to facilitate the understanding of the management mechanisms and the proposed algorithmic solutions. The first sections of this manuscript present the entities that compose the NFV-enabled ecosystem, the management mechanisms to adapt consumption to available generation, and the mathematical characterization of the ES and the ECs. If necessary, detailed information about an NFV-enabled architecture for adaptive energy

management is available in our previous work [8]. The architecture in [8] is the reference environment for deploying the heuristics proposed in this paper.

Subsequently, this paper presents the adaptive energy management model using an Integer Linear Programming (ILP) formulation. For solving the ILP model, an optimal algorithmic strategy defined as OptTsCost is developed. In this context, in [12], we present an optimal approach defined as OptTs, which, using time-shifting capabilities and a brute-force search, performs a combinatorial analysis to find the best scheduling of demands that produce optimal energy utilization. To select the best combination of energy demands, OptTs uses a nested sort procedure based on performance metrics related to the amount of residual power achieved, the demands that are processed, and the application of time-shifting. Apart from degrading the operation of the algorithm, this sorting method makes it difficult to include new management mechanisms (e.g., prioritization of supply) for adaptive consumption. Thus, OptTsCost is an evolution of OptTs and has two main differentiators: (i) prioritization of energy supply by including a priority schema linked to energy demands; and (ii) selection of the best combination of energy demands based on a cost function. This latter feature facilitates both the search for the optimal combination of energy demands and incorporating different performance metrics to the energy management model. Additionally, the development of OptTsCost allows us to identify all the aspects and concerns related to the algorithmic implementation of adaptive energy management.

Given that the proposed adaptive energy management model has an NP-Hard complexity [13], and the exact solution OptTsCost has exponential growth, this paper presents three heuristic strategies: (i) a greedy strategy based on a constructive algorithm and defined as GreedyTs, (ii) a genetic-algorithm-based solution identified as GATs, and (iii) a solution based on a dynamic programming approach and described as DPTs. These algorithmic strategies and the optimal solution, when deployed at the NFV domain, seek the best possible scheduling of demands that leads to efficient utilization of available power. Specifically, the heuristic strategies aim to extend the applicability of the energy management solution to scenarios with thousands of demands. As a reference point, [12] reports that OptTs is restricted to a maximum of nine energy demands and a time-shifting of three time slots (same limits for OptTsCost).

With the objective of further scaling up the operation of both the optimal and heuristic strategies, this paper also presents a prepartitioning strategy based on a divide-and-conquer method. The main idea of this complementary mechanism is to divide the original problem (i.e., optimal scheduling of energy demands) into a set of small sub-problems of less complexity that can be solved iteratively by the algorithmic management strategies (both OptTsCost and the heuristics). In summary, for the adaptive energy management in IoT scenarios, this paper presents eight algorithmic strategies: first, an optimal solution defined as

OptTsCost, then three heuristic approaches described as GreedyTs, GATs, and DPTs, and, finally, four heuristic solutions obtained as a result of the application of the prepartitioning method on the original strategies, which are identified as OptTsCostPart, GreedyTsPart, GATsPart, and DPTsPart.

The performance evaluation of the algorithmic management strategies is carried out through extensive simulations in several scenarios, including an online scenario and a HEMS in which real consumption data has been used. The simulation results demonstrate that applying the adaptive energy management model through the proposed algorithmic strategies (optimal and heuristics) produces an improved overall performance of the generation and consumption ecosystem, reflected in increased processed demands and a decrease in residual power. The major contributions of this paper are summarized as follows:

- The description of an IoT and NFV-enabled ecosystem for the adaptive energy management of available supply, whether or not it is renewable.
- An energy management model for adaptive consumption constrained to availability and the corresponding ILP formulation.
- An optimal brute-force-search-based algorithmic strategy defined as OptTsCost to solve the ILP optimally.
- Three heuristic strategies identified as GreedyTs, GATs, and DPTs to solve the adaptive energy management problem in reduced running time compared to the optimal solution, OptTsCost, and with applicability to IoT scenarios with thousands or hundreds of thousands of energy demands.
- The application of a pre-partitioning method inspired in a divide-and-conquer approach to scale up the operation of both optimal and heuristic strategies.
- The evaluation of the exact and heuristic strategies through extensive simulations in different scenarios to confirm the improvements in the management of energy consumption and demand processing.

The most relevant acronyms and notations used throughout the paper are presented in Table 1 and Table 2, respectively. The rest of the paper is organized as follows. Section II discusses the related work. Section III introduces the NFV-enabled ecosystem in which the algorithmic strategies, both optimal and heuristics, are intended to be deployed. Section IV formally presents the problem of efficient and adaptive consumption of available power and the ILP model. Section V explains the OptTsCost strategy for solving the ILP problem optimally. The heuristic algorithmic strategies are addressed in Section VI. Evaluation results are analyzed in Section VII. Finally, conclusions and future work are presented in Section VIII.

II. RELATED WORK

This section reviews the related work. Section II-A describes the potential use of NFV technology for resource management, including energy. Section II-B discusses some

TABLE 1. List of acronyms.

Acronym	Definition
CS	Critical Services
DC	Data Center
DR	Demand Response
DP	Dynamic Programming
EC	Energy Consumer
EM	Energy Manager
ES	Energy Supplier
HEMS	Home Energy Management System
ICT	Information and Communications Technologies
ILP	Integer Linear Programming
IoE	Internet of Energy
IoT	Internet of Things
MANO	Management and Orchestration
MMKP	Multidimensional Multi-choice Knapsack Problem
NCS	Non-Critical Services
NFV	Network Functions Virtualization
PAR	Peak-to-average ratio
SDN	Software Defined Network
SFC	Service Function Chain
VNF	Virtual Network Function

TABLE 2. List of notations.

Notation	Description	
Energy supply	m	Time interval in which energy exists
	P_{ES}	Power provided by the energy supplier
	P_{NR}	Power from non-renewable energy sources
	P_R	Power from renewable energy sources
	$T_{init}^{P_{ES}}$	Starting time of P_{ES}
	w_R	Weight associated to renewable energy
Consumption and services	i	Service identifier, $i \in \{1, \dots, N\}$
	j	Priority identifier, $j \in \{1, \dots, L\}$
	L	Number of priority levels
	N	Total number of services
	t_i	Starting time of service S_i
	d_i	Duration of service S_i
	p_i	Power demanded of service S_i
	q_i	Priority level of service S_i
	u_i	Time shifting value of service S_i
	P_D	Aggregated power demanded
W	Maximum analysis time horizon	
Metrics	AR	Acceptance ratio
	P_{LACK}	Missing power
	P_{RES}	Residual power
	$\sigma_{P_{RES}}$	Standard deviation of P_{RES}
	σ_{T_s}	Standard deviation of T_s^k
Strategies	$disOptT_s$	Maximum distance to optimum
	GAT_s	Genetic algorithm based strategy
	$GreedyT_s$	Greedy strategy
	G_T	Relative gain in time
	DPT_s	Dynamic programming Based Strategy
	$OptT_sCost$	Optimal strategy based on a cost function
ρ	Approximation ratio	

IoT-based energy management approaches. Finally, Section II-C summarizes the differentiators of our proposal.

A. USE OF NFV TECHNOLOGY FOR ENERGY MANAGEMENT

NFV technology has been shown to be an effective platform for deploying management applications, optimization models

(mainly based on heuristic approaches), and network services to meet the diverse requirements of customers and vertical markets [14]. Based on network parameters (e.g., traffic load, energy consumption estimations, or active users), the NFV architectural framework can be set to perform actions such as optimized routing of traffic flows, activation of devices, and allocation of resources (both physical and virtual) to achieve desired performance metrics or functionalities (e.g., low-latency requirements [15]). Regarding energy consumption and management in the NFV realm, many studies have focused on the energy-aware operation of VNFs and SFCs, such as in [16] and [17]. Other studies have explored the potential of NFV for resource and energy management in IoT-enabled environments outside the NFV infrastructure. For instance, in [18], Wantamane *et al.* present an NFV framework that executes an application for real-time synchronization of machine-to-machine sensors nodes, enabling the deployment of a building energy management system.

In mobile communications landscape, NFV-based solutions for energy management have also been analyzed. In these cases, the authors exploit the virtualization and management capabilities of NFV to minimize the energy footprint in different portions of the 5G network infrastructure (i.e., access, transport, and core networks). To achieve energy efficiency, the authors propose linear programming models and optimization algorithms to improve resource utilization in terms of both cost and performance, as shown in [19]. To deal with complexity issues and for real-time applications, the authors also present heuristic strategies (e.g., genetic-based algorithms) as indicated in [20]. The energy management applicability for particular 5G use cases also has been investigated. For instance, in [21], the authors propose an NFV-enabled energy management scheme for a drone fleet with 5G connectivity based on an optimal scheduling algorithm that aims to ensure a given level of service availability.

Considering that NFV-based management policies can be implemented into DCs (cloud computing infrastructures) belonging to the energy utilities [10], the evolution of NFV-based schemes for energy management seems to be a natural process in smart grids. Initial studies demonstrate that NFV technology can be used to virtualize components of power grids (e.g., advanced metering infrastructure), as shown in [22], producing better performance in exchanging information on energy production and consumption. Recent works, instead, analyze the use of NFV and ICT technologies to improve communications among the components of smart grids [23]. For instance, in [24], Yang *et al.* present an NFV/SDN-based model that slices the resources in core networks and coordinates the activation of SFCs to meet end-to-end low latency requirements for mission-critical energy services. Meanwhile, in [25], the authors present an SDN/NFV-based infrastructure that offers optimal placement and dynamic resource allocation of middleboxes, enabling them to meet both cyber-security and low-latency communication requirements in smart grids. The works reviewed in

this section demonstrate that NFV is a suitable environment for deploying management solutions in different domains (e.g., 5G networks and power grids).

B. IoT-BASED ENERGY MANAGEMENT APPROACHES

Many studies have shown that IoT technologies are essential in modern energy systems by offering improvements in monitoring, control, management, and automation processes [26]. These technologies are key enablers for deploying the Internet of Energy (IoE) [27]. When they are integrated into energy systems, they can be used to implement different approaches such as DR programs or renewable energy management systems [28]. For instance, DCs participating in DR schemes can adapt their power profile (e.g., through workload consolidation or migration) according to available supply [10], easing the incorporation of renewable energy into the grid [29]. Also, DCs can execute mechanisms to coordinate generation and consumption adaptation among the ES and the ECs, respectively [8]. Apart from DCs, other ICT infrastructures widely used to implement energy management solutions are the demand-side facilities [30]. In these installations, DR schemas are relatively easy to be implemented due to the infrastructure available (e.g., automated systems and control devices in smart homes). Also, the customer can participate directly in the energy management process.

For demand-side management, the literature presents many DR schemas and models in industrial facilities but mainly in HEMS. In [31], Wei *et al.* propose an IoT-based common information model and communication framework to deploy a DR energy management system for industrial consumers. The experimental results in [31] show that an IoT-based platform allows for the rapid and low-cost implementation of an integrated energy management system that can reduce the peak-to-average ratio (PAR) and overall energy costs. Meanwhile, in HEMS, existing proposals encompass both complete systems and algorithmic strategies. In [32], the authors present a HEMS that adaptively manages renewable-energy production (from photovoltaic panels) through the use of wireless smart devices (i.e., sensors and actuators) controlled by a central hub in which DR strategies are executed.

Regarding algorithmic strategies for adaptive energy consumption in HEMS, existing research works have mainly focused on PAR reduction (by reshaping the demand profile) [33], user utility maximization [34], consumption cost minimization [35], and incorporation of renewable energy [36]. Energy management problems are modeled through optimization techniques such as ILP [37], or other approaches, such as game theory [33]. Offline [38] and online [39] algorithms are used to solve these models. Also, heuristic solutions for rapid convergence and simple steps are proposed to reach efficient energy management [36]. For instance, in [37], the authors propose an ILP model that allows maximizing consumer utility (or minimizing energy cost) by adjusting the hourly load level of a given consumer

in response to hourly electricity prices. The approach shown in [37] is centralized, and all the decisions are taken entirely by the ES. Due to the complexity of centralized schemes (especially if they produce optimal solutions), distributed models are also proposed. In [33], for instance, the authors propose a demand-side management algorithm using a game-theoretical approach in which each user (player) tries to minimize their consumption.

Due to the complexity of optimal algorithmic strategies in HEMS, some researchers focus exclusively on heuristic approaches. In [35], for instance, Chavali *et al.* propose an approximate greedy algorithm, in which each EC schedules the consumption of appliances in response to varying electricity prices. The optimization model in [35] is based on minimizing cost functions for each EC. These functions consider the constraints of the appliances and user preferences in the starting consumption time. The results in [35] show that efficient load scheduling results in lower cost for the ECS and the ES, and reduced PAR and load fluctuations. In [38], instead, the authors propose a strategy that computes load scheduling considering photovoltaic availability. The optimization problem is targeted at minimizing the cost of energy and time-based discomfort. Also, an inclining block rate scheme (i.e., a higher rate for each incremental block of consumption) is incorporated into the model to reduce the PAR.

C. DIFFERENTIATORS OF OUR PROPOSAL

Most studies in the NFV realm address energy management and efficiency by minimizing consumption or encouraging energy savings, as shown in [16]. Our proposal instead leverages the dynamic, programmable, and scalable features offered by NFV technology to deploy an adaptive energy management solution conditioned on availability (whether renewable or not) and carried out through algorithmic strategies (using optimal but mainly heuristic approaches). The proposed solution also exploits the manageability of the ECs enabled through massive connectivity and IoT technologies for energy management. In this regard, considering that the IoT already connects billions of devices and keeps growing exponentially (e.g., 28.5 billion IoT devices estimated in 2022 [9]), the proposed model and strategies in this paper could potentially be applied to manage a plethora of ECs.

Therefore, an NFV-and IoT-based approach is a feasible alternative for adaptive energy management and it is aligned with the requirements of future energy systems [26]. This is because of the following features: (i) use of advanced ICT infrastructures in the energy management process (algorithmic management strategies deployed at NFV domain); (ii) customer-side participation and use of renewable energy sources (the proposed energy management model considers the parameters of ECs and prioritizes the use of green energy); and (iii) adaptive energy consumption adjusted to the generation (achieved by diverse management mechanisms in the algorithmic strategies). Furthermore, the proposed

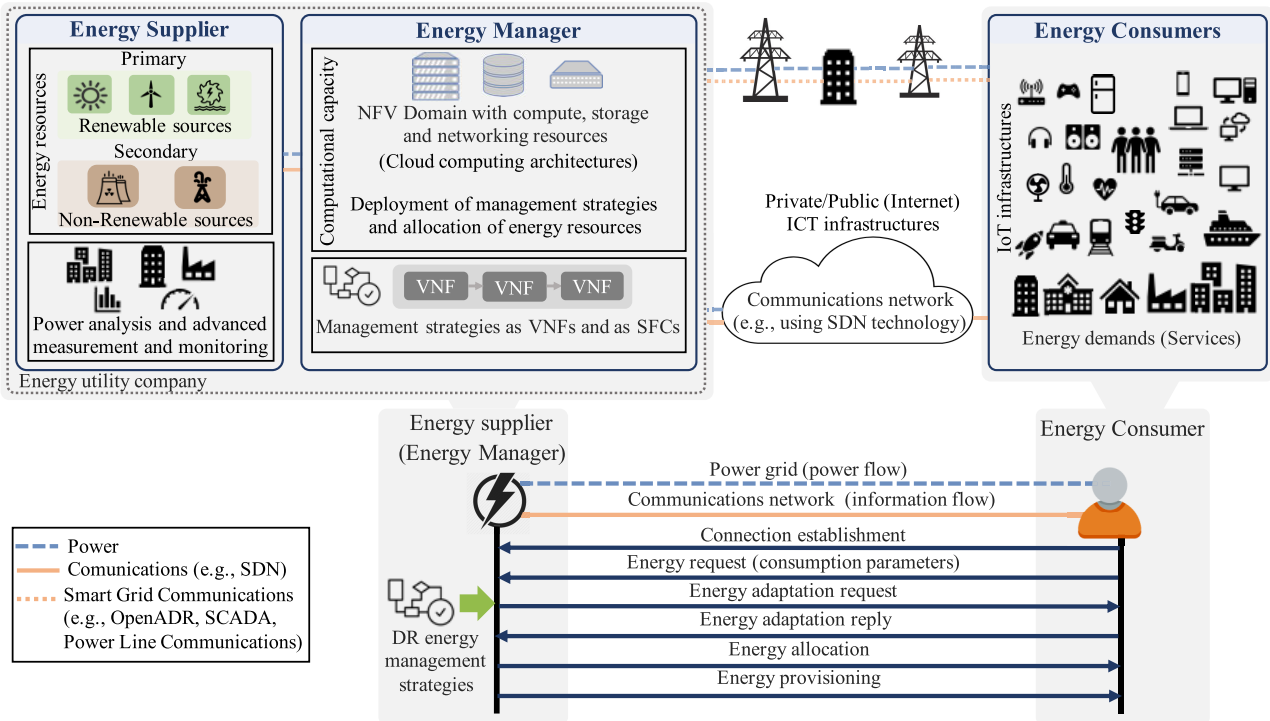


FIGURE 1. Overview of an NFV-enabled ecosystem for adaptive energy management, adapted from [8].

energy management model does not focus exclusively on the supplier side (as shown in [22]), the customer side (like the demand-side approaches discussed in Section II-B), or specific infrastructures or users (e.g., energy management in buildings [18]). This paper, on the contrary, provides the mathematical models of the ES, the ECs, and the adaptive consumption process. Also, the proposed algorithmic strategies can be applied to scenarios with thousands or hundreds of thousands of energy demands in contrast to most of the existing approaches which, are limited to small-scale scenarios (e.g., in HEMS with up to 10 demands [33] or at most up to 100 demands [35] if heuristic methods are used).

III. NFV-ENABLED ENERGY MANAGEMENT ECOSYSTEM

This section presents an overview of an NFV- and IoT-enabled ecosystem for energy management and the mechanisms used to adapt consumption to available supply.

A. ENTITIES IN THE ENERGY MANAGEMENT ECOSYSTEM

Figure 1 illustrates an NFV-enabled ecosystem for adaptive energy management [8]. The ecosystem is composed of three entities, which are described below.

1) ENERGY SUPPLIER (ES)

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

2) ENERGY CONSUMERS (ECs)

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

3) ENERGY MANAGER (EM)

The EM is part of the energy utility, and it corresponds to the ICT infrastructure (e.g., cloud computing facilities) in which the NFV technology is deployed. It provides the control, management, and orchestration functions of energy resources and demands. The NFV paradigm gives the EM: (i) reconfigurable behavior by activating or creating

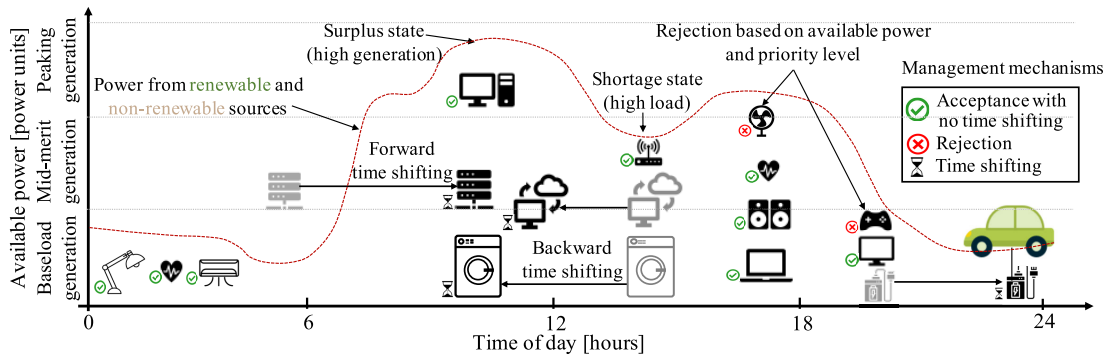


FIGURE 2. Example of application of the management mechanisms for adaptive energy management, adapted from [8].

VNFs (or complete SFCs) according to the algorithmic management strategy to be implemented and the applicability environment; (ii) scalability, due to the on-demand use of computational capacity; and (iii) the MANO entities (NFV Orchestrator, VNF manager, and Virtualized Infrastructure Manager) so that ES, EC, and the underlying communications systems (e.g., an SDN network) work in an orchestrated manner. These features create a robust energy management solution that can be applied to different scenarios (e.g., HEMs or smart cities).

The EM enables two-way interaction between the ES and the ECs. This cooperation between parties can be supported by contracts or agreements in which technical and economic terms are defined [10]. Technically, through a handshake process, as shown in Fig. 1, the ECs notify the ES (technically to the EM) of their demands. Then, with the generation and consumption information, the EM runs the management algorithmic strategies (through mechanisms such as time-shifting implemented as VNFs in the NFV domain) to determine the consumption conditions. Specifically, the EM delivers the ECs an optimal power scheduling scheme that enables them to adapt the consumption patterns to available supply (i.e., the EM performs adaptive energy management). Different communications systems can provide scalable, secure, and reliable connectivity to exchange data on energy management (e.g., parameters of demands and instructions of consumption) between the ES and the ECs. However, SDN technology is one of the best alternatives due to its compatibility with NFV and IoT [40].

B. MANAGEMENT MECHANISMS FOR ACHIEVING ADAPTIVE CONSUMPTION

The desynchronization between generation and consumption (mainly due to the recent rise of intermittent green energy sources) causes inefficient energy utilization. This condition appears in the energy system as an increase in PAR and costs (both for the ES and the ECs), energy shortages, and even waste of energy if the available resource cannot be used or stored. To solve this challenge, a feasible alternative is adaptive energy management. It consists of adapting consumption patterns to the available supply through diverse procedures

on energy demands. In this regard, this section presents three management mechanisms that, when incorporated in the algorithmic strategies deployed in the NFV domain, enable adaptive energy management. Figure 2 shows an example of the application of these mechanisms, and their description is presented below.

1) USE OF TIME-SHIFTING CAPABILITIES

The time-shifting denoted as u is the finite displacement (forward or backward) on the execution time of energy demand [12]. This mechanism allows the ES (through the EM) to increase or decrease consumption by anticipating or delaying the execution of energy demands during periods of energy surplus or energy shortage, respectively. The ES can encourage the ECs to advance or delay their demands by offering a set of reduced energy tariffs depending on the time-shifting performed.

2) PRIORITIZATION OF ENERGY SUPPLY

The ES (technically the EM) can use a prioritization scheme to differentiate the energy resource allocation and the application of management mechanisms such as time-shifting and rejection (no energy allocation) on energy demands. The number of priority levels, the priority level for each energy demand, the actions per each priority, and the limits of each management mechanism (e.g., maximum time-shifting interval) are agreed upon between the EM and the ECs (through contracts) [8].

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

- 1) *Critical Services (CS)*: The CS have the highest priority level, denoted as $j = 1$, with $j \in \{1, \dots, L\}$, in the proposed energy management model. CS cannot be interrupted or shifted to earlier or later periods, and they

cannot be rejected. The ES prioritizes the allocation of the energy resource available to CS. The services in emergency scenarios (e.g., life support devices) and natural disasters (e.g., for search and rescue operations) are examples of CS.

2) *Non-critical services (NCS)*: The NCS are shiftable (advanced or delayed execution). However, once operation begins, they cannot be interrupted until the operation completes. Multiple priority levels can be used for the NCS (e.g., $j \in \{2, \dots, L\}$). Thus, after energy allocation for CS, the remaining supply is distributed to the NCS based on their priority level (from $j = 2$ up to $j = L$). The use of different priority levels for NCS seeks the best energy utilization and optimal comfort for the ECS. Examples of services (IoT infrastructures and applications) within this category include entertainment services (non-essential audio and video systems), dishwashers, washing machines, water pumps, water, heaters, and fans.

3) REJECTION OF ENERGY DEMANDS

Services are rejected when the available supply is insufficient to cover all demand (due to energy shortage or high load) or when the service(s) cannot be adapted to the energy profile even if time-shifting is used [8]. The rejection criteria are based on the priority level of NCS. Thus, services with a lower priority level (e.g., $l = L$) during periods of energy scarcity are more likely to be rejected.

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

C. MATHEMATICAL MODELS OF ENERGY GENERATION AND CONSUMPTION

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

1) ENERGY SUPPLIER MODELING

In the proposed energy management model, the ES is characterized by a total power supply capacity denoted as P_{ES} . It is *de facto* the power received at the point of consumption (regardless of losses). The P_{ES} has an initial time defined as $T_{init}^{P_{ES}}$ and a finite duration denoted as m . It is equal to the sum of power coming from renewable (e.g., solar, wind, or hydroelectric) and non-renewable (e.g., coal or natural gas) energy sources, defined as P_R and P_{NR} . Considering that ES can prioritize green energy use, a weight $w_R \in [0, 1]$ is included in the P_{ES} to control the provisioning capacity from non-renewable sources.

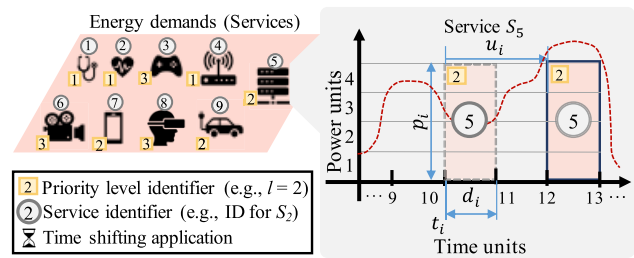


FIGURE 3. Graphical representation of a service with time-shifting application. Parameters of S_5 : $N = 9$, $L = 3$, $t_5 = 10$, $d_5 = 1$, $p_5 = 3$, $q_5 = 2$, $u_5 = +2$ (forward).

For sustainable reasons and as a requirement for future energy systems [27], the participation of the P_{NR} in the P_{ES} is expected to be minimal and in the best scenario equal to zero (i.e., when $w_R = 1$). The mathematical expressions that represent the P_{ES} , the P_R , and the P_{NR} , are shown in Eq. 1, Eq. 2, and Eq. 3, respectively.

$$P_{ES} = P_R + P_{NR} \quad (1)$$

$$P_R = P_{ES} \cdot w_R \quad (2)$$

$$P_{NR} = P_{ES} \cdot (1 - w_R) \quad (3)$$

2) ENERGY CONSUMERS MODELING

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

The consumption model considers a total of N energy demands or services. Each service i , with $i \in \{1, \dots, N\}$ is identified as S_i . It has an initial time, an interruptible duration, and a fixed power level denoted as t_i , d_i , and p_i , respectively. The power demanded or consumed by all services for which the ES (technically the EM) can allocate energy is denoted as P_D . This value represents the total power consumption. Also, each service S_i has a priority level identified as q_i that can be affected by a time-shifting value (in the case of the NCS) denoted as u_i . The time-shifting can be backward (i.e., when $t_i - u_i$) for the anticipated execution of a service S_i , equal to zero (i.e., $u_i = 0$ or not time-shifting applied) for the normal processing of a service S_i , or forward (i.e., when $t_i + u_i$) for the delayed execution of a service S_i . Normally a service S_i cannot only use a specific value of u_i but move in an interval $\{t_i - u_i, \dots, t_i, \dots, t_i + u_i\}$. Table 3 summarizes the parameters related to the services. An example of these parameters is illustrated in Fig. 3.

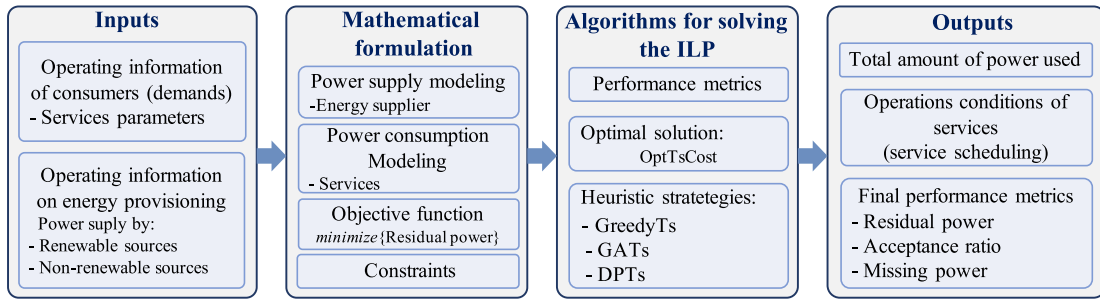


FIGURE 4. Schematic of the management model for adaptive energy consumption.

TABLE 3. Parameters of services or energy demands.

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

IV. MATHEMATICAL FORMULATION OF THE ENERGY MANAGEMENT MODEL

The energy model for adaptive energy consumption is illustrated in Fig. 4. This section presents the related mathematical formulation. Section IV-A describes the assumptions considered in the proposed model. Section IV-B presents the ILP formulation for adaptive energy management, while Section IV-B analyzes its complexity.

A. ASSUMPTIONS RELATED TO THE ENERGY MANAGEMENT MODEL

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

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

B. ILP PROBLEM FORMULATION FOR ADAPTIVE ENERGY MANAGEMENT

1) PRIORITIZATION IN THE USE OF GREEN ENERGY

Before discussing the objective function to optimize, this section presents the mathematical expression to prioritize green energy use for scenarios in which this process is applicable. Different approaches can be used to promote renewable sources; however, a simple solution is to establish a cost function related to power consumption, as shown in Eq. 4. The general cost function in Eq. 4, defined as $Cost_{PD}$, comprises individual costs associated with the consumption of renewable and non-renewable power, denoted as $Cost_{PR}$ and $Cost_{PNR}$, respectively. The ES can tune the $Cost_{PD}$ by modifying the weights w_1 and w_2 in the range $[0, 1]$ according to some operating parameters (e.g., P_R available). In this context, to promote the use of green energy, the $Cost_{PR}$ can be set to a minimum value (i.e., if $w_1 \ll w_2$) or zero (i.e., if $w_1 = 0$ or $Cost_{PR} = 0$), in such a way that the total cost only depends of the use of P_{NR} .

The $Cost_{PNR}$ can be defined as a value proportional to the amount of the P_{NR} consumed, as shown in Eq. 5. In this regard, the task of the ES (technically the EM) is to find the best supply-consumption conditions (e.g., encouraging consumption during a surplus of wind or solar energy) to obtain a minimum value of the $Cost_{PNR}$ (Eq. 6). Thus, the minimization of $Cost_{PNR}$ is equivalent to the minimization of P_{NR} (as shown in Eq. 7) or the prioritization of the consumption of P_R .

$$Cost_{PD} = w_1 \cdot Cost_{PR} + w_2 \cdot Cost_{PNR} \quad (4)$$

$$Cost_{PNR} = \begin{cases} 0 & \text{if } P_{ES} = P_R, \\ \sum_{i=1}^n p_i, n \in N & \text{otherwise.} \end{cases} \quad (5)$$

$$\text{minimize } \{Cost_{PNR}\} \quad (6)$$

$$\text{minimize } \{P_{NR}\} \quad (7)$$

2) OBJECTIVE FUNCTION

Adaptive energy management is achieved by adapting consumption to generation and aims at the optimal use or consumption of the available supply (whether renewable or not). In this proposal, the optimal adaptive consumption is obtained by minimizing the wasted or unused available

power, which is mathematically expressed as the difference between the P_{ES} and the P_D , as shown in Eq 8. Considering the analysis for each time slot k , Eq 8 becomes Eq 9. In addition, for simplicity the difference between P_{ES} and P_D is referred to as residual power and is denoted as P_{RES} . Thus, the objective function (linear function) in the proposed adaptive energy management model is summarized in the minimization of the P_{RES} , respecting the constraints shown in Section IV-B3.

$$\text{minimize } (P_{ES} - P_D) \quad (8)$$

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

3) CONSTRAINTS

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

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

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

Considering that each variation r , with $r \in V_i$, demands a certain amount of power during a finite interval and at a given starting time, the problem of the adaptive energy management using the time-shifting mechanism consists of choosing the best variations per class V_i (i.e., if x_{ir} takes on a value of 1), such that utilization of the available supply P_{ES} is maximized (minimization of the P_{RES}). In this context, the set of N or n (with $n \subset N$) variations analyzed is defined as a *combination of variations* or simply a *combination* and is denoted as $Comb_f$ (e.g., $Comb_1$). The algorithmic strategies (optimal or heuristics) have the task of finding the best combination among all possible combinations denoted as $AllComb$ ($Comb_f \in AllComb$) produced due to different variations per class V_i . The selection of the best combination is explained in Section IV-B5.

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

$$C1 : P_{ES}[k] \geq 0 \quad (10)$$

$$C2 : (P_{ES}[k] - P_D[k]) \geq 0 \quad (11)$$

$$C3.1 : \sum_{j=1}^L y_{ij} = 1, i \in \{1, \dots, N\} \quad (12)$$

$$C3.2 : y_{ij} \in \{0, 1\}, i \in \{1, \dots, N\}, j \in L \quad (13)$$

$$C4 : \sum_{i=1}^N \sum_{r \in V_i} p_{ir}[k] \cdot x_{ir} \leq P_{ES}[k] \quad (14)$$

$$C5.1 : \sum_{r \in V_i} x_{ir} = 1, i \in \{1, \dots, N\} \quad (15)$$

$$C5.2 : x_{ir} \in \{0, 1\}, i \in \{1, \dots, N\}, r \in V_i \quad (16)$$

$$C6 : t_i \geq 0 \quad (17)$$

$$C7 : \{t_i - u_i\} \geq 0 \quad (18)$$

$$C8 : W \geq \max\{t_i + d_i + u_i\} \quad (19)$$

$$C9 : T_{init}^{P_{ES}} \geq 0 \quad (20)$$

$$C10 : W \geq \{T_{init}^{P_{ES}} + m\} \quad (21)$$

4) METRICS TO EVALUATE THE COMBINATION OF VARIATIONS

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

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

$$\sigma_{P_{RES}_f} = \sqrt{\frac{\sum (P_{RES_{comb}_f})^2}{m}} \quad (22)$$

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

variations (services), the AR can be expressed as:

$$AR_f = \frac{N - RejServ}{N} \cdot 100 \% \quad (23)$$

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

$$P_{LACK_f} = \begin{cases} |P_{ES} - P_{DComb_f}| & \text{if } P_{ES} < P_{DComb_f}, \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

In addition, with the mean value of the P_{ES} , the estimation of the interval m to promote an $AR = 100\%$ (if consumption can be adapted to the energy profile) is given by Eq. 25.

$$m \simeq \frac{\sum_{i=1}^N P_i \cdot d_i}{P_{ES}} \quad (25)$$

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

$$\sigma_{Ts_f} = \sqrt{\frac{(\sum_{i \in N} u_i)^2}{N}} \quad (26)$$

5) ADAPTIVE ENERGY MANAGEMENT BASED ON A COST FUNCTION

The selection of the best combination of variations (i.e., the combination that produces the minimization of the P_{RES}) can be established based on the best result of one (mainly based on the values of $\sigma_{P_{RES_f}}$) or all metrics in Section IV-B4. Nevertheless, to obtain the optimal allocation of energy resources and the corresponding optimal consumption, all the metrics and parameters related to adaptive energy management must be considered together. To this end, a possible mechanism shown in [12] is to perform a nested sorting to the set of all combinations $AllComb$ based on an increasing value of $\sigma_{P_{RES_f}}$ and σ_{Ts_f} and a decreasing value of σ_{Ts_f} . As a result of this process, the first combination in the sorted list represents the optimal service scheduling that enables the minimization of the P_{RES} . Although the nested sorting method delivers the best combination of variations, it is limited to a small number of metrics because each

new metric increases the complexity of the process by a factor equal to the size of the set $AllComb$; also, this method does not incorporate priority information. To overcome these limitations, in the proposed energy management model, the selection of the best combination is carried out in a single sorting step using the cost function shown in Eq. 27, which allows incorporating different parameters (more than three metrics) to cope with the specific requirements of the ES or the ECs.

The cost function in Eq. 27 is composed of individual costs related to $\sigma_{P_{RES}}$, AR , σ_{Ts} , and the cumulative value of priorities of the variations in the combination, which are denoted as $Cost_{P_{RES_f}}$, $Cost_{AR_f}$, $Cost_{U_f}$, and $Cost_{Q_f}$ and are given by Eq. 28, Eq. 29, Eq. 30, and Eq. 31, respectively. In addition, the values of the costs in Eq. 27 can be tuned by the EM according to the preference for some parameter(s) using the weights α , β , γ , and δ in the range [0,1]. For analytical simplicity, these weights are set to one.

$$Cost_{comb_f} = \alpha \cdot Cost_{P_{RES_f}} + \beta \cdot Cost_{AR_f} + \gamma \cdot Cost_{Q_f} + \delta \cdot Cost_{U_f} \quad (27)$$

$$Cost_{P_{RES_f}} = \begin{cases} \sigma_{P_{RES_f}} \cdot \varepsilon & \text{if } AR_f = 100\%, \\ \sigma_{P_{RES_f}} & \text{otherwise.} \end{cases} \quad (28)$$

$$Cost_{AR_f} = \begin{cases} 0 & \text{if } AR_f = 100\%, \\ RejServ \cdot \mathcal{M} & \text{otherwise.} \end{cases} \quad (29)$$

$$Cost_{U_f} = \begin{cases} 0 & \text{if } \forall Var_r^{S_i} : u_{ir} = 0, \\ \sum_{i=1}^n \sum_{r \in V_i} u_{ir} & \text{otherwise.} \end{cases} \quad (30)$$

$$Cost_{Q_f} = \begin{cases} 0 & \text{if } \forall Var_r^{S_i} : q_{ir} = j, \\ \sum_{i=1}^n \sum_{r \in V_i} q_{ir} & \text{otherwise.} \end{cases} \quad (31)$$

In Eq.28, the parameter ε can be set in the range [0,1] ($\varepsilon = 0.5$) and can be used to differentiate the $Cost_{P_{RES_f}}$ of the different combinations of variations if $P_{RES} > 0$. In Eq.29, instead, the parameter \mathcal{M} represents a big value (e.g., $\mathcal{M} = 1000$) and is used to penalize those combinations that deliver more rejected variations. In addition, Eq.29 can include the cumulative value of priorities of variations (i.e., $\sum_{i \in RejServ} q_i$) to penalize those combinations that produce higher priority rejected services.

The corresponding cost function is calculated for each combination of variations f , producing a list of costs denoted as $AllCost$. From this list, the best cost (OptCost) is the one with the lowest value, as shown in Eq. 32, and corresponds to the best combinations of variations ($OptComb$, with $OptComb \in AllComb$) that enable adaptive management through the minimization of the P_{RES} .

$$OptCost = \arg \min_{Cost_{comb_f} \in AllCost} Cost_{comb_f} \quad (32)$$

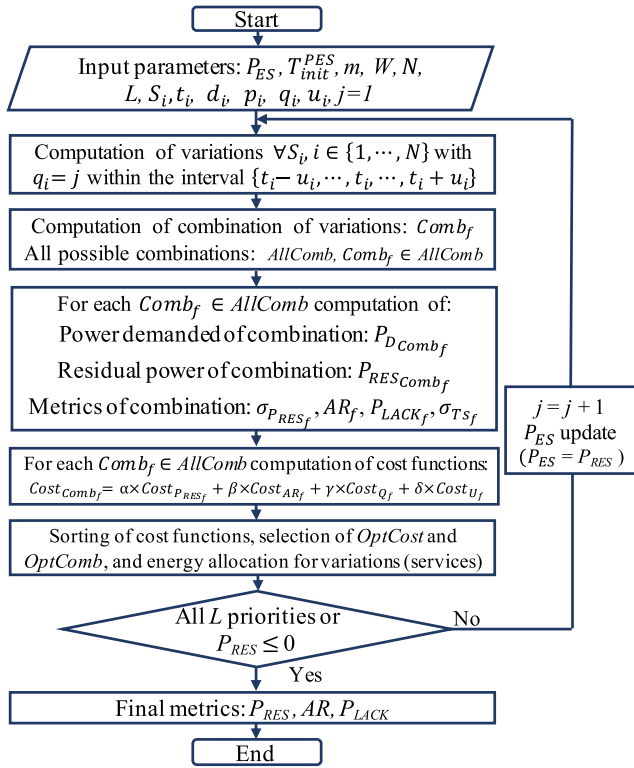


FIGURE 5. Flow chart of OptTsCost.

C. HARDNESS OF THE PROBLEM

The objective of adaptive energy management in our proposal consists of maximizing the use of the available energy supply (minimization of the P_{RES}) through the execution (selection) of the best variations of services (one variation per class V_i , with $i \in N$). This process is analogous to the objective of the multi-dimensional multi-choice knapsack problem (MMKP) of choosing the most valuable items of a set of classes (one item per class) without overloading the knapsack [41], which has been demonstrated to be \mathcal{NP} -hard [41]. Based on this analogy, we can then conclude that the proposed energy model falls in the MMKP classification and presents a complexity that is \mathcal{NP} -hard.

V. OPTIMAL SOLUTION: OptTsCost

To optimally solve the ILP model in Section IV-B, different approaches can be used. This section presents an exact or optimal algorithmic strategy defined as OptTsCost based on a brute-force exhaustive search paradigm in which the entire search space is explored to find the optimal solution. The algorithm is explained in Fig. 5 and starts with the computation of the variations of the services considering all values within the time-shifting interval (i.e., $\{t_i - u_i, \dots, t_i, \dots, t_i + u_i\}$). With the information on variations, the algorithm builds all possible combinations of variations. It then computes the metrics and cost function for each combination. Later, it selects the best combination based on the minimum value of the $Cost_{comb_f}$. This process is

carried out iteratively for each priority level, and at the end the strategy OptTsCost delivers the optimal service scheduling that produces the minimization of the P_{RES} . Figure 6 shows a summarized example of the application of strategy OptTsCost for $N = 4$ services.

The complexity in OptTsCost depends on the steps carried out. Considering for analytical simplicity the same value for the time-shifting backward and forward (u_i), the growth rate of OptTsCost as a function of N , u_i , and L can be summarized in Eq. 33. The terms that comprise this expression correspond to the number of services, variations, and combinations processed for each priority level. Of all the terms in Eq. 33, the most dominant is the third, which reveals that OptTsCost presents an exponential growth rate with an order of growth $O(2^N)$ (according to Big-O notation) that depends on the maximum values of N and u_i . This condition imposes the applicability of the OptTsCost to small-scale scenarios. For instance, using only $N = 10$ services and $u_i = 4$ time slots the algorithm has to produce and later analyze over three billion combinations to find the best service scheduling.

$$f(L, N, u_i) = N + ((2 \cdot N \cdot u_i + N) + (2 \cdot u_i + 1)^N) \cdot L \quad (33)$$

As an example of the deployment of management strategies as SFCs in the NFV domain, Fig. 7 illustrates the strategy OptTsCost decomposed into VNFs. These VNFs correspond to the different steps carried out by the algorithm to minimize the P_{RES} , can be deployed on virtual machines or containers, and can use on-demand computation resources according to the needs of the ES and the ECS.

VI. HEURISTIC STRATEGIES

Although the strategy OptTsCost is a powerful approach for finding optimal consumption conditions, evaluating all possible combinations of variations is computationally demanding and intractable for values of $N > 10$ services or $u_1 > 4$ time slots [12]. These operating conditions motivate the development of less complex approaches in which the optimal results can be relaxed to obtain reduced running time and lower utilization of computational capacity. In addition, a reduction of complexity can enable adaptive energy management in large-scale scenarios (e.g., for $N = 100$ services or $u_i = 10$ time slots).

Considering that the proposed energy management model is categorized as an MMKP, different existing techniques and strategies can be adapted to the context of adaptive energy management. In this regard, to tackle the exponential complexity of AI and to cover as far as possible the main categories of methods or techniques for efficiently solving the MMKP (in our proposal, the P_{RES} minimization problem) [13], three heuristic algorithmic strategies have been proposed: (i) a greedy strategy, (ii) a genetic algorithm-based solution, and (iii) a strategy based on a dynamic programming method. These strategies are described below.

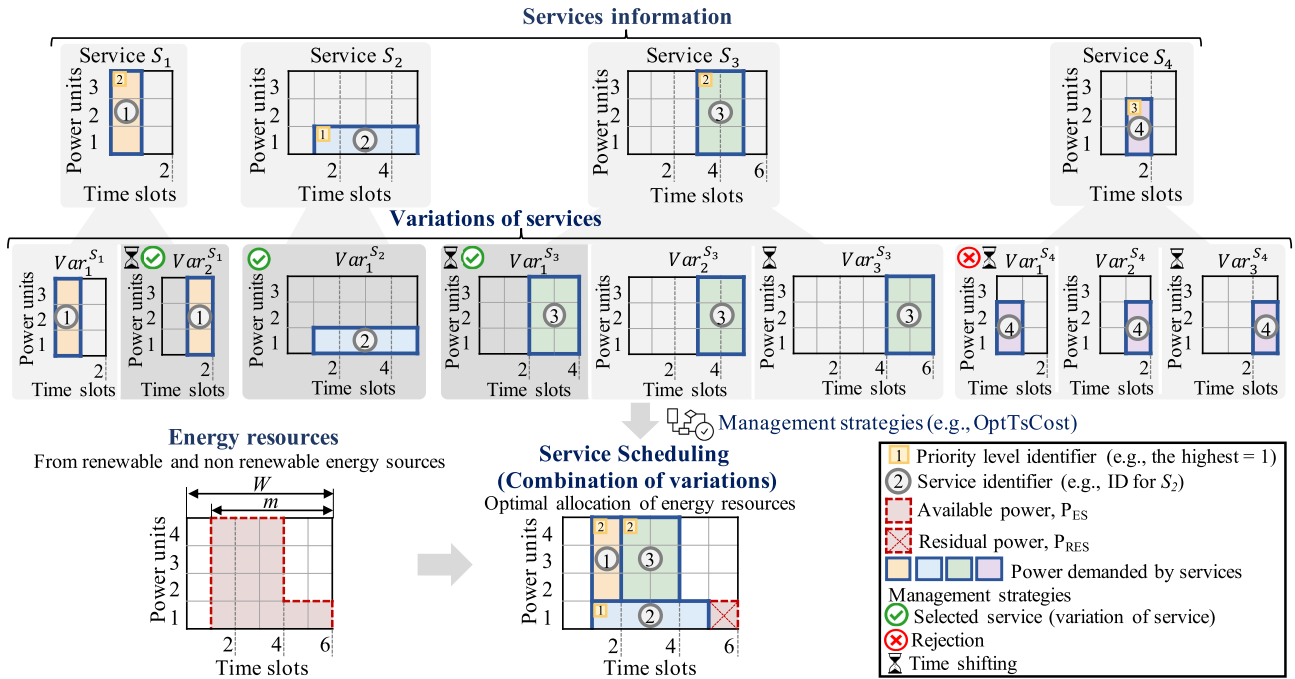


FIGURE 6. Example of application of the strategy *OptTsCost*. In this example, the optimal service scheduling that produces the minimization of the P_{RES} is achieved if S_1 is executed one slot in delay, S_2 is processed in its original execution time, S_3 is executed one slot in advance, and S_4 is rejected.

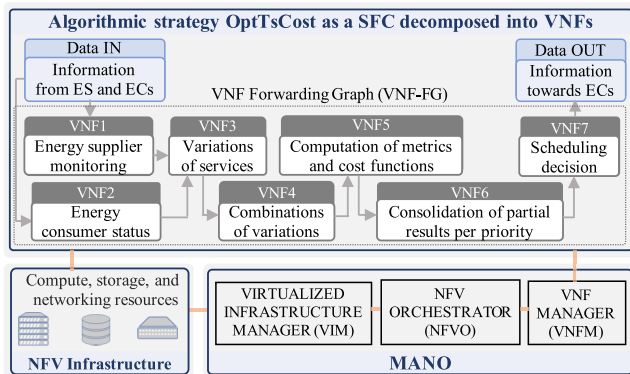


FIGURE 7. Example of the deployment of *OptTsCost* as a SFC in the NFV-enabled EM.

A. GREEDY STRATEGY: GreedyTs

Greedy algorithms are simple schemes intended to produce feasible solutions quickly. These algorithms are iterative and constructive in the sense that starting with an empty solution, in every iteration part of the solution is obtained (never changed later) so in the last iteration the complete solution is created. The decision in each iteration is made in an attempt to optimize a performance metric or maximize an immediate benefit (e.g., taking first the most valuable item and then the next most valuable in the 1/0 knapsack problem) [42]. In the context of adaptive energy management, the proposed greedy strategy defined as GreedyTs iteratively builds the optimal service scheduling that produces the minimization

of the P_{RES} . The algorithmic strategy GreedyTs is explained in Fig. 8, the main steps carried out are summarized below.

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

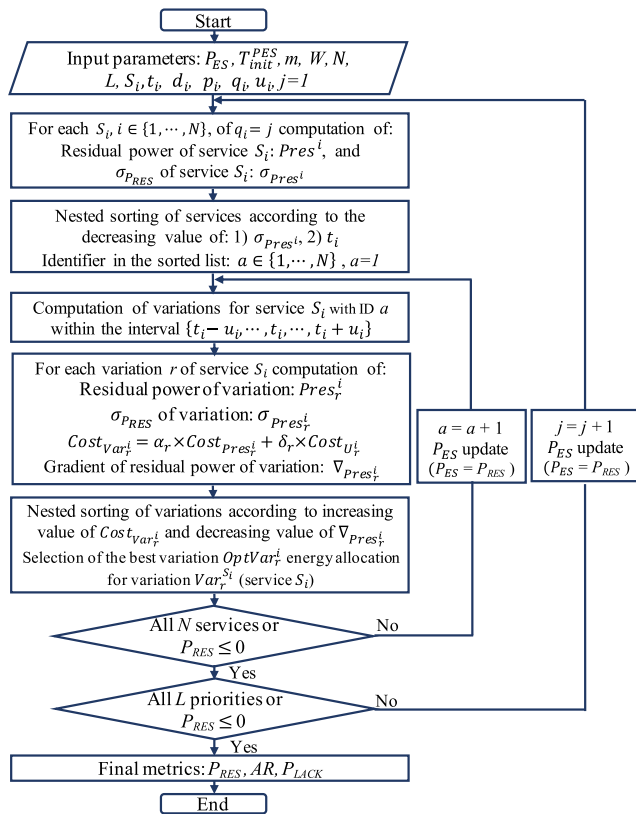


FIGURE 8. Flow chart of GREEDYT.

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

The complexity of GreedyTs is summarized in Eq. 34. In this expression, the second term is dominant and reveals that the growth rate of the algorithm is polynomial and depends on the maximum values of N and u_i . However, if one of the parameters remains constant (whether N or u_i), the growth rate of GreedyTs can become linear.

$$f(L, N, u_i) = N + (2 \cdot N \cdot u_i + N) \cdot L \quad (34)$$

B. GENETIC ALGORITHM BASED STRATEGY: GATs

Genetic algorithms belong to the class of evolutionary algorithms and are search techniques used to generate near-optimal solutions in varied optimization problems [43]. These algorithms are inspired by the biological evolution of the species over generations and by the process of natural selection of the fittest. Regarding the implementation, genetic algorithms follow principles of natural genetics and rely on operators such as reproduction, mutation, crossover, and selection [42].

The evolution process starts with a population of possible solutions to an optimization problem created using a random

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

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

In the context of adaptive energy management, the proposed genetic-based strategy defined as GATs iteratively creates a set of combinations of variations and selects the ones with the best P_{RES} values for the next generation. In the end, the algorithm delivers the best possible combination that leads to the minimization of the P_{RES} . The algorithmic strategy is explained in Fig. 9, and the main steps carried out are summarized below.

- 1) *Coding the solution:* In the genetic algorithms and adaptive energy management scope, a chromosome represents a combination of variations, and each gene represents a variation of service.
- 2) *Validation of variations:* For each priority level, in this step those variations that individually produce a negative residual power ($Pres_r^i$) are eliminated. This procedure allows reducing the number of combinations

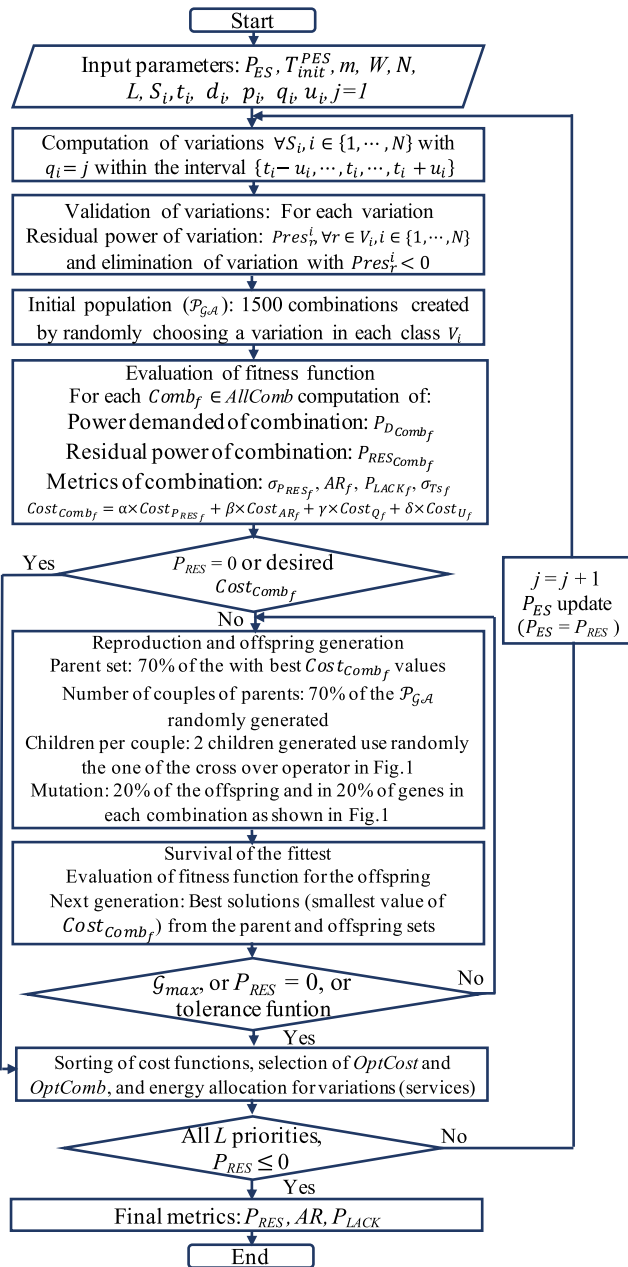


FIGURE 9. Flow chart of GATs.

with a $P_{REScomb} < 0$ and avoiding the analysis for the rejection of the corresponding service(s).

- 3) *Initial population*: It is recommended that the population (denoted as \mathcal{P}_{GA}) has hundreds or thousands of chromosomes to guarantee diversity in solutions [42]. In this regard, a preliminary analysis in [12] has demonstrated that if the number of combinations is lower than or equal to 2401 (a situation that occurs when $u_i = 3$ and $N = 4$), the exploration of the entire search space can be performed in less running time (e.g., units of seconds or less). Based on this reference and after preliminary tests (tests applied to

set other parameters of the algorithm) on the quality of the solutions produced, the initial population size has been set at a maximum of 1500 non-repeated combinations, which are generated by randomly choosing the variations (valid) of each class V_i . In the case that the theoretical number of combinations $((2 \cdot u_i + 1)^N)$ is lower than 1500, all possible combinations are used in the population. The population size remains constant for the rest of the algorithm steps.

- 4) *Evaluation of fitness function*: The fitness function in the adaptive energy management is represented by the cost function $Cost_{comb_f}$, which is computed for all combinations in the population. If within the initial population there are combinations that produce a desired performance metric or threshold (e.g., $P_{RES} = 0$ or a given $Cost_{comb_f}$), the combination with the best value (e.g., minimum $Cost_{comb_f}$) is chosen, and immediately the energy is allocated to the respective variations. Otherwise, the algorithm continues to iteratively create and analyze generations of combinations to obtain better values of $Cost_{comb_f}$.
- 5) *Reproduction and offspring generation*: For reproduction, 70% of the population (ensuring an even number of combinations) with the minimum cost function values ($Cost_{comb_f}$) is chosen to create the parent set (i.e., following an elitist approach). The combinations in this set are randomly selected to form pairs. At this point, the number of pairs is half the number of combinations of the paternal set because a parent can only participate in one pair at a time. Each pair produces two child solutions created by mixing the variation of progenitors and randomly using one of the three established crossover operators (one-point, multipoint, and uniform crossover operators). Figure 10 shows an example of obtaining offspring and the application of crossover operators. Twenty percent of the offspring obtained are affected by the mutation process. A mutation operator is applied to each solution (combination) of this group, which that affects approximately 20% (genes) of the variations. Usually, the mutation operator randomly exchanges/flips some genes in the analyzed chromosome. However, in the context of the proposal this procedure is performed by selecting a different variation from the one analyzed in the class V_i ; an example is shown in Figure 10. Throughout generating offspring and applying genetic operators, the created and mutated solutions (children) are conditioned to be different from the combinations of the current population (parents) to guarantee diversity in the search space.
- 6) *Survival of the fittest*: Following an elitist approach, the combinations with the minimum $Cost_{comb_f}$ s values from the parent and offspring sets are selected to build the next generation of individuals (of the same

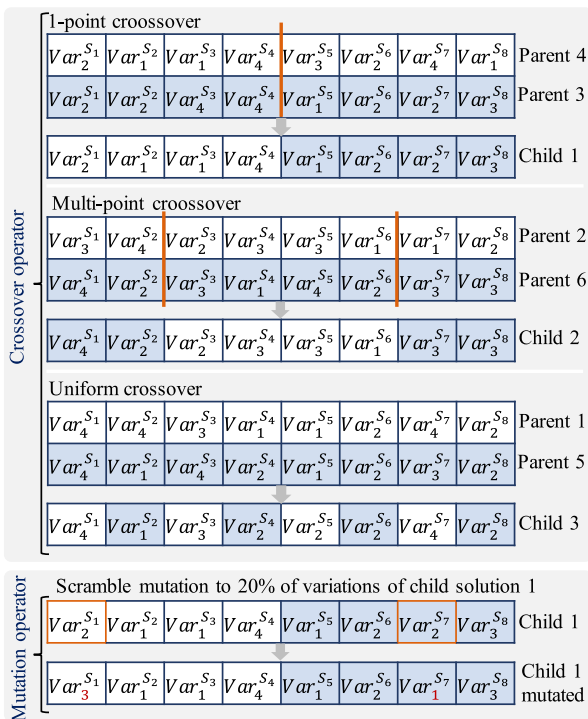


FIGURE 10. Example of crossover and mutation operators in the Genetic Algorithm for $N = 8$ services.

size as the current population \mathcal{P}_{GA}). The algorithm then continues the iterative process until any of the following stop conditions is met: (i) reach the maximum number of generations \mathcal{G}_{max} , which in the strategy is fixed to 20 generations based on previous validation of solutions, (ii) obtain at least a combination with $P_{REScomb} = 0$ (i.e., ensure the utilization of all available supply), or (iii) obtain a relative change in the value of the $P_{REScomb}$ lower than the tolerance function, which has been set as 1% of the $P_{REScomb}$ of the previous generation.

The complexity of GATs depends on the number of variations analyzed and the combinations explored over generations, as summarized in Eq. 35. The process of exploration of combinations is dominant in Eq. 35 and reveals that the growth rate of the algorithmic strategy can be: (i) exponential for values of N and u_i that produce a number of combinations smaller than 1500 (e.g., for $N = 4$ and $u_i = 2$) or (ii) polynomial if the population remains constant (1500 combinations) and there are variations (increases) in the number of generations used to reach feasible solutions. In a particular case, the growth rate of GATs can be linear if the number of generations remains constant as analyzed services increase.

$$f(L, N, u_i) = N + ((2 \cdot N \cdot u_i + N) + \mathcal{P}_{GA} \cdot \mathcal{G}_{max}) \cdot L \quad (35)$$

C. DYNAMIC PROGRAMMING STRATEGY: DPTS

Dynamic programming (DP) is a mathematical technique that solves a complex problem optimally or sub-optimally by breaking it into simpler subproblems. Then, each of those subproblems is solved (optimally) just once, and their solutions are stored (in a data structure, e.g., an array) so that they can be used (repeatedly if necessary) to solve the original problem. DP can be applied when a solution to a problem can be recursively described in terms of solutions to subproblems (i.e., when the subproblems overlap) [44]. For instance, in a naive recurrent Fibonacci computation, the same values are computed repeatedly for each new number. A simple computation for the second and third Fibonacci numbers requires the computation of the first Fibonacci number twice (one for the second and one for the third). DP solves this issue by storing the already computed values so that the second time they are needed, they can be obtained immediately. In addition, since the subproblems are interrelated, the final solution can be obtained easily using a traceback process through the partial solutions. A variety of computational and optimization problems can be addressed using DP approaches, including solving MMKP [45].

The storage of partial solutions in DP provides high-quality results (optimal or suboptimal) and reduced time (complexity) compared to other methods such as brute force strategies [44]. There are two ways to store the partial solutions so that these values can be reused: (i) a bottom-down approach, also known as tabulation, that iteratively solves all subproblems and uses their solutions to fill up a table (a data structure whereby starting from the first entry, all entries are filled one by one), then, the stored results in the table are used to compute the solution to the original problem (or bigger subproblems); and (ii) a top-down approach, also known as memoization, that also uses a tabular form to store partial solutions but differs from the tabulation approach because the table is filled on-demand. Before solving a subproblem in the top-down approach, the algorithm will search for its solution in the lookup table. If the solution has been stored, this result can be directly used; otherwise, the problem is solved, and its solution is stored in the table so that it can be used later.

The computational implementation of tabulation is based on an iterative method; instead, memoization exploits recursivity. If a certain problem requires all subproblems to be solved (as in the case of adaptive energy management, in which all services (variations) must be analyzed to determine the set of N services that minimize P_{RES}), tabulation usually outperforms memoization. This is because the former has no overhead for recursion, and the required solutions can be directly retrieved from values in the table. Moreover, tabulation has been proven to be an effective DP method for solving knapsack problems [46]. Based on this information, a tabulation-based approach has been chosen for implementing the proposed DP algorithmic strategy.

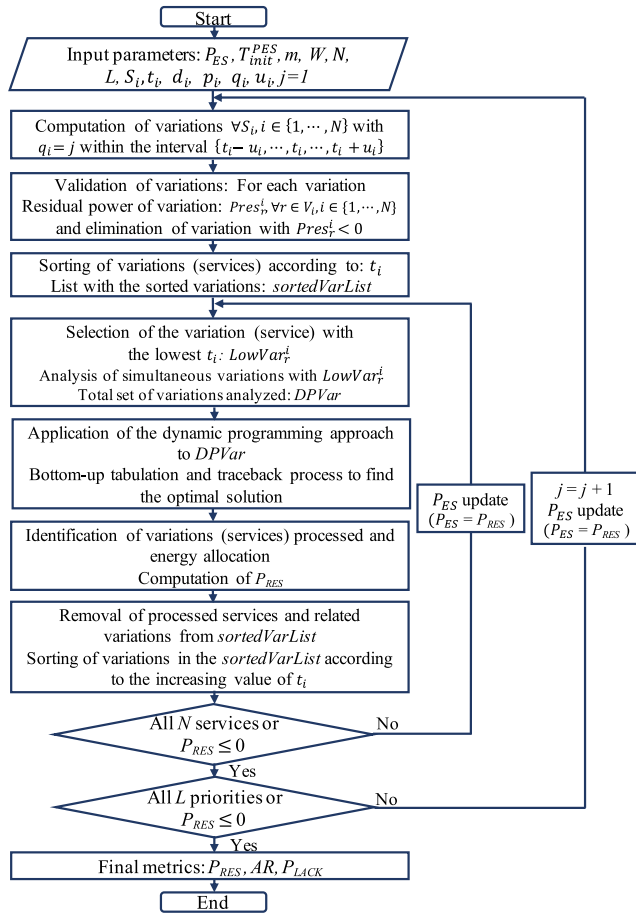


FIGURE 11. Flow chart of DPTs.

In the context of adaptive energy management, the DP-based strategy, defined as DPTs progressively analyses variations, and by means of a tabulation approach, it selects the ones that enable P_{RES} minimization while maximizing the AR . The algorithmic strategy is explained in Fig. 11, and the main steps carried out are summarized below.

- 1) *Analysis of services and variations:* Like the GreedyTs approach, the strategy DPTs focused on the analysis of variations (instead of combinations) for each priority level. As the first step, the algorithm computes the variations of services within the corresponding time-shifting intervals. Then, the variations that individually produce a negative residual power ($Pres^i$) are eliminated. This procedure, by reducing the search space for the best variations, contributes to speeding up the execution of the algorithm. Once the variations are validated, the algorithm sorts them based on the increasing value of the starting time (t_i). This criterion aims to maximize energy use and considers the time evolution of the available energy resource (i.e., T_{init}^{PES} and m). Thus, an efficient allocation in the first (earlier) services will promote a greater remaining available power (more freedom to select variations that minimize the P_{RES}) for the subsequent

services. At this point, the strategy DPTs selects the variation with the lowest starting time (i.e., the first one) from the sorted list *sortedVarList* and processes all concurrent/simultaneous variations (*DPVar*). A variation is considered concurrent if executed (coexists) within the lifetime (d_i) of the variation under analysis. The idea is to analyze a set of concurrent variations instead of individual variations and solve them using a DP method in the next step of the algorithm. Given that the DPTs algorithmic strategy progressively analyzes groups of simultaneous variations instead of all possible variations for practicality and to reduce the complexity of the associated search, the solution produced by the strategy is not optimal. However, studies prove that DP applied to MMKP can deliver high-quality approximate or suboptimal solutions reasonably quickly (compared to other methods such as brute-force) [47].

- 2) *Dynamic programming tabulation and the selection of processed variations:* In this step, the algorithm applies DP to the concurrent variations to select the ones whose execution optimizes P_{RES} and maximizes AR . For the implementation of DP, the algorithm uses a tabulation method and considers the energy resource that coexists with the time slots of variations analyzed. Figure 12 shows an example of the application of DP for a set of seven variations that belong to four services. To carry out DP on variations, the algorithm starts by creating a table with a number of rows equal to the energy capacity per time slot K plus one unit (e.g., $P_{ES}[k] + 1 = 3 + 1$ for time slot 10 in Fig. 12) and with a number of columns equal to the number of concurrent variations plus one unit (e.g., $n + 1 = 4 + 1$ in Fig. 12). A rigorous implementation of the energy model described in Section IV-B requires that the algorithm performs the DP method for each slot k . In this case, the algorithm determines the processed variations based on the results obtained during all time slots analyzed. A variation is considered as processed if the algorithm (using a DP method) can allocate energy for all time slots in which it exists (i.e., for d_i). Moreover, for a more exhaustive assessment of processed variations, the algorithm DPTs can include a simple greedy-based method that verifies the power demanded by each processed variation and the remaining available power. In the event that the level of P_{ES} remains constant during all time slots analyzed (as shown in Fig. 12), the algorithm could simplify the DP analysis to the first time slot (time slot 10 in the example), which can speed up the exploration of optimal variations of services. In any of the cases, before carrying out the DP, the algorithm evaluates the generation and consumption conditions for the set of variations analyzed. In the created DP table, the rows with identifier a represent all possible values of available power

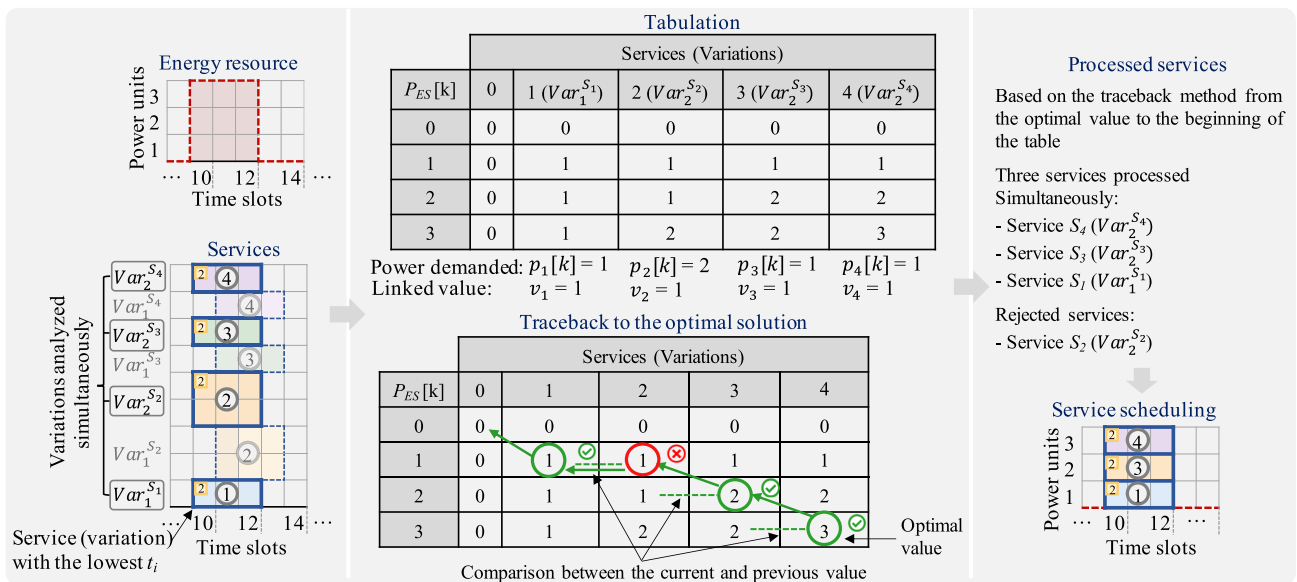


FIGURE 12. Example of application of the Dynamic Programming Algorithm for $N = 4$ services.

(i.e., $0 \leq a \leq P_{ES}$). In contrast, the columns with identifier b correspond to the individual information of variations (i.e., $0 \leq b \leq n$), including the information when no variation is selected (the first column). Based on the adaptation of the DP method to solve MMKP [45], the entries in the table of strategy DPTs correspond to the cumulative optimal value or profit due to the selection of variation(s) respecting the maximum energy capacity. The individual value linked to each variation (service) is identified as v_b , and the entry stored in the table in row a and column b is denoted as $\mathcal{V}[a, b]$. The value v_b in the proposed strategy DPTs is assigned according to the priority level of the variation in analysis so that a higher value is assigned to a higher priority (e.g., $v_b = L$ if $q_b = 1$). If all the concurrent variations have the same priority level, the value v_b is unique and is set to one (i.e., $v_b = 1 \forall b \in n$, as shown in Fig. 12). An alternative criterion to establish the v_b value could be based on a cost function or metric (e.g., P_{RES}) that indicates the impact on energy use due to the selection (processing) of certain service(s).

The strategy DPTs systematically fills up the table column by column (i.e., one service at a time, from 0 up to n). For each column b , the algorithm progressively analyses each row a (from 0 up to P_{ES}) and assesses the selection (or not) of the variation with identifier b so that the value of the entry $\mathcal{V}[a, b]$ corresponding to the selected variation(s) \hat{n} , with $\hat{n} \subset n$ (i.e., $\mathcal{V}[a, b] = \sum_{b \in \hat{n}} v_b$) is maximized (optimal). In each row a , the algorithm verifies that the power demanded by the selected variation(s) fits the available energy capacity. Moreover, in each column and corresponding row, the algorithm verifies if the value for the entry

$\mathcal{V}[a, b]$ has been already computed in the previous column to avoid recomputation of the same value (i.e., using stored values to solve greater subproblems). This condition gives the bottom-up approach to the tabulation method and makes the space memory and running time rather efficient. In summary, the computation of any entry $\mathcal{V}[a, b]$ depends on the power demanded and values of the variations in column b , and values in the previous column of the table, as established by Bellman's equation [44], that adapted to the context of the strategy DPTs is shown in Eq. 36.

At the end, of the exploration of all entries, the best possible cumulative value is stored at the bottom right corner of the table (i.e., at $\mathcal{V}[a_{th}, b_{th}]$). In the example in Fig. 12, the best value is equal to three, which indicates that three variations (services) have been processed, and in this particular case, they use the P_{ES} optimally. To identify the variations that have been selected (processed), the algorithm uses a traceback method, in which, one by one, the columns (variations) are analyzed. The process starts with the value at entry $\mathcal{V}[a_{th}, b_{th}]$; this value is compared to the value at entry $\mathcal{V}[a_{th}, b_{th}-1]$. If these values are different, the algorithm has selected the variation in column b_{th} . Then, the next entry analyzed is $\mathcal{V}[(a_{th}-p_{b_{th}}), b_{th}-1]$, and the value in this entry is compared with the one of the previous column (same row). The process continues progressively until the algorithm reaches the beginning of the table (i.e., the upper left corner). In the example, in Fig. 12, the services S_1, S_3 , and S_4 are processed. Eq. (36), as shown at the bottom of the next page.

- 3) *Service scheduling*: The selected variations and all others belonging to the processed services (because a service can produce several variations) are removed

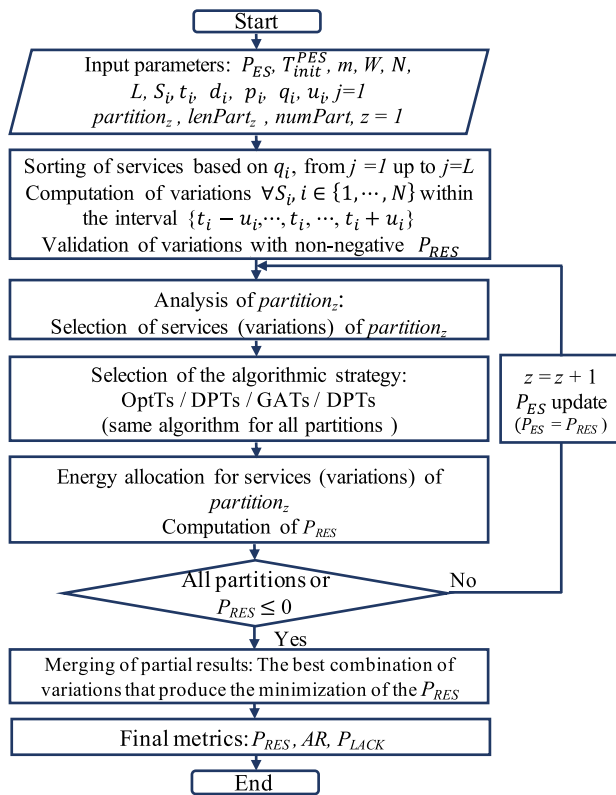


FIGURE 13. Flow chart of the prepartitioning method applied to the algorithmic strategies.

from the *sortedVarList*. Then, the algorithm proceeds to reorder the remaining variations based on the increasing value of t_i . The selection of variations through PD is executed progressively until the energy resource is not available or until all variations and levels of priority have been explored. At the end the algorithm presents the metrics P_{RES} , AR , and P_{LACK} .

The literature has demonstrated that a tabulation-based DP can solve an MMKP in pseudopolynomial time because a solution boils down to filling in values in the DP table (using two nested for loops) and each row is typically computed in constant time. Even DP for MMKP can deliver a polynomial time if the number of items to be packed into the knapsack is small [48]. Consequently, the complexity in the proposed strategy DPTs is pseudopolynomial. Specifically, the growth rate of DPTs depends on the maximum values of N , u_i , and mainly on the progressive DP analysis of the set of concurrent variations $DPVar$, as shown in Eq. 37.

$$f(L, N, u_i) = N + ((2 \cdot N \cdot u_i + N) + N \cdot DPVar) \cdot L \quad (37)$$

D. PREPARTITIONING STRATEGY

Preliminary tests on the heuristic strategies (which are discussed in detail in Section VII) have demonstrated high-quality results in energy use and reduced running time compared to OptTsCost (optimal solution) with applicability to scenarios in the range of thousands of energy demands. However, to further scale up the applicability of energy management to IoT scenarios with dozens or hundreds of thousands of services in this section, we propose a complementary method that could be applied to OptTsCost, as well as to GreedyTs, GATs, and DPTs to improve their performance in terms of scalability. The proposed idea is the application of a prepartitioning method on services. Inspired by a divide-and-conquer approach, a well-known design technique proven to produce efficient solutions with little or no loss of accuracy [49], the proposed prepartitioning method aims to iteratively analyze smaller subsets of simultaneous services instead of the original set of energy demands. These subsets are then solved using algorithmic strategies (optimal or heuristics), and their partial solutions are combined to obtain adaptive energy management for the original problem.

The prepartitioning method applied to the proposed algorithmic strategies is explained in Fig. 13. In this method, the total number of partitions is denoted as $NumPart$ ($1 < NumPart \leq N$), $partition_z$ (subset of services, $partition_z \subset N$), which has a length $lenPart_z$. This length can be the same (or approximately the same) for all partitions, or it can be different for each partition depending on factors such as the priority of services, application scenario, or other specific objectives required in energy management. In any case, the reduction of search space in the partition domain contributes to reducing the complexity of the original strategy, either optimal or heuristic. Iteratively, each $partition_z$ is solved by the selected algorithmic strategy. All partitions are resolved by the same algorithm, although a hybrid strategy (e.g., the joint application of DP and GATs) may be considered in future work. Once all the partitions have been solved, or the P_{ES} has been allocated, the prepartitioning method delivers the scheduling (suboptimal) for all N services and the metrics P_{RES} , AR , and P_{LACK} . In summary, the application of the prepartitioning method on OptTsCost, GreedyTs, GATs, and DPTs originates four additional heuristic strategies that are identified in this paper as OptTsCostPart, GreedyTsPart, GATsPart, and DPTsPart, respectively. Regarding the complexity, Eq. 38 summarizes the growth rate related to the application of the prepartitioning method. In this expression, the third term is dominant and represents the cumulative complexity of solving all $NumPart$

$$\mathcal{V}[a, b] = \begin{cases} 0 & \text{if } n = 0 \text{ or } P_{ES} = 0, \\ \max \{v_{b-1} + \mathcal{V}[a - p_{b-1}, b - 1], \mathcal{V}[a, b - 1]\} & \text{if } p_{b-1} < a, \\ \mathcal{V}[a, b - 1] & \text{otherwise.} \end{cases} \quad (36)$$

partitions by the selected strategy.

$$f(L, N, u_i) = N + (2 \cdot N \cdot u_i + N) \cdot L + \sum_{z=1}^{NumPart} StratPart_z \quad (38)$$

E. ADAPTATION OF ALGORITHMIC STRATEGIES FOR ONLINE SCENARIOS

Adaptive energy management can be implemented for offline or online approaches. In the offline approach, the service scheduling strategies, as shown in Section V, Section VI-A, Section VI-B, and Section VI-C, know in advance all generation and consumption parameters, and they are capable of performing both backward and forward time-shifting on services. The offline approach can be used to plan the distribution of energy resources, reshape the load profile (e.g., reduce peak loads), and prioritize the use of renewable energy sources, which can produce a reduction of overall operational cost and carbon emission levels and promote sustainability in the generation and consumption ecosystem [8]. Instead, in the online approach, the service scheduling algorithmic strategies have no future information about generation and consumption; the services are processed as time evolves, and only the forward time-shifting can be applied to services. The online approach represents the real-time dynamic of provisioning and consumption in which adaptive energy management must be performed. In this regard, this section presents the online version of the proposed service scheduling strategies, including the prepartitioning method. Fig. 14 explains the generic algorithm for adapting the developed service scheduling strategies (optimal and heuristics) for online applications. The main steps and additional features performed are described below.

- 1) *Initial analysis of services*: The online implementation of service scheduling strategies starts with the differentiation of the energy resources for the processing of CS (P_{CS}) and NCS (P_{NCS}). The proposed strategy assumes that once the service is accepted (in its first slot), there is P_{ES} for its completion (the model does not accept fractional processing, as discussed in Section IV-A). An additional feature for the online approach is the inclusion of a list named *waitingList*, which stores information on the variations of services that were not processed in their original starting time (t_i). This is due to energy allocation to higher-priority service(s).
- 2) *Analysis for CS*: If service S_i at time slot k is identified as a CS, the strategy allocates the demanded energy resource. Later, the strategy updates the P_{CS} for the rest of CS.
- 3) *Analysis for NCS*: If service S_i at time slot k is identified as an NCS, the strategy performs a similar analysis as for the offline approach, considering the variation at time slot k , the simultaneous variations with the variation in analysis, and the variations in *waitingList*. In this regard, a variation is considered simultaneously if it exists within the lifetime (d_i) of the analyzed

variation (service) at time slot k . Once the strategy selects the best combination/variation, the energy allocation is made, and the energy resources for subsequent NCS are updated.

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

VII. EVALUATION

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

A. SIMULATION SETUP

The algorithmic strategies have been implemented on Matlab R2018b and running on a machine with a 3.33 GHz · 12 cores Intel Core i7 Extreme processor and 24 GB RAM. The simulations leverage parallel processing with the concurrent use of up to 6 cores. The results obtained of metrics P_{RES} , AR , and P_{LACK} are compared to the benchmark scenario in which no management mechanism is applied (i.e., no time-shifting, prioritization, or rejection) [50].

B. DESCRIPTION OF GENERATION AND CONSUMPTION PROFILES FOR SIMULATED SCENARIOS

1) GENERATION AND CONSUMPTION PROFILES

To analyze the performance of the proposed algorithmic strategies in different generation and consumption conditions, four profiles have been considered. In these profiles, the total available energy is equal to the total energy demand. The profiles are summarized in Fig. 15 and Fig. 16a and are described below.

- *Profile I*: This profile allows the analysis of the performance of service scheduling strategies in total desynchronization periods of energy supply and consumption (i.e., during periods of scarcity and abundance of power), as shown in Fig. 15a. In this case, the P_{ES} is consumed only if time-shifting is applied to services. Moreover, in Profile I and Profile II, a flat-supply profile (representing a realistic generation scenario, as studied in [51]) has been chosen for simplicity in the analysis. However, the algorithmic strategies have no restrictions working with any demand and supply profile if needed.
- *Profile II*: This profile simulates a peak demand due to high load, as shown in Fig. 15b. Moreover, in this profile, services with different (random) values of p_i and d_i have been considered to simulate a more realistic consumption scenario.
- *Profile III*: This profile allows for the analysis of the performance of scheduling strategies in a futuristic environment 100% powered by green energy sources. This is a very promising approach to tackle sustainability

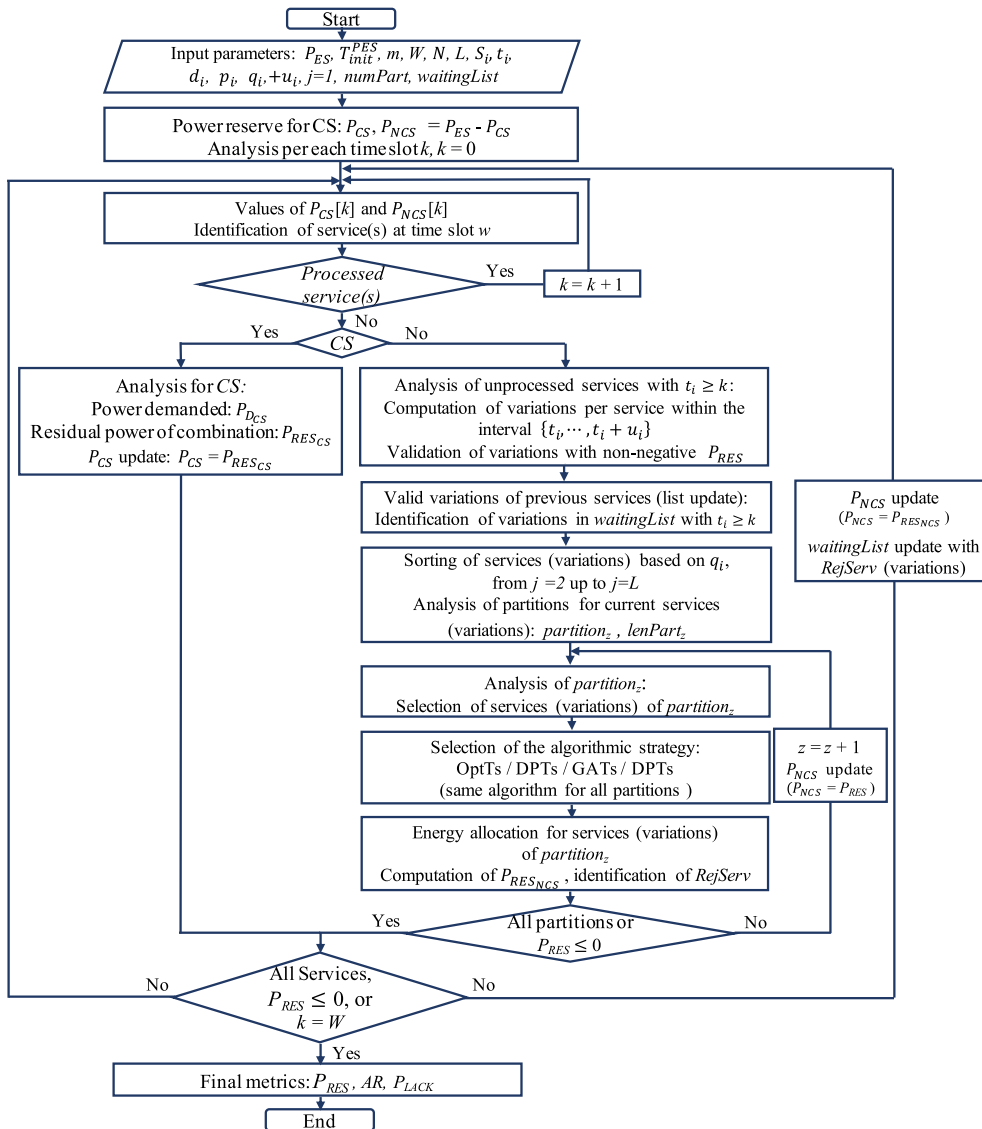


FIGURE 14. Flow chart of the adaptation of the service scheduling strategies for an online approach.

issues, increasing carbon emissions due to generation and consumption of non-renewable energy sources, and is an important requirement for deploying the IoE [27]. In this profile, as shown in Fig. 15c, the services have random values of p_i and d_i and are partially desynchronized with the P_{ES} . Moreover, the supply follows a Gaussian distribution to simulate the renewable generation patterns (e.g., through photovoltaic panels), as shown in [52].

- **Profile IV:** This profile allows for the analysis of the application of the heuristic scheduling strategies in a HEMS. Figure 16a shows the supply and consumption profiles for the simulated HEMS. The consumption of 20 services (appliances) is adapted from [53], contrasted with the data in [54], and summarized in Table 4.

TABLE 4. Description of consumption for the HEMS.

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

2) SCENARIO DESCRIPTION

Using the profiles described in Section VII-B1, seven application scenarios have been analyzed summarized in Table 5.

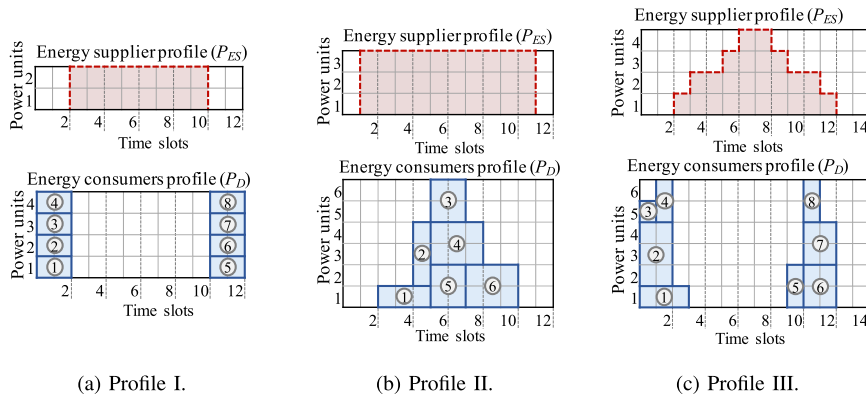
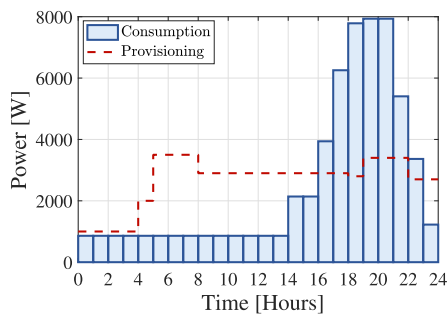
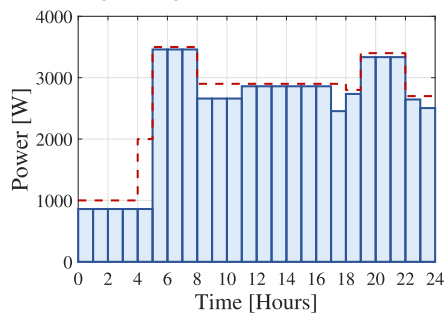


FIGURE 15. Power supply and consumption profiles for simulated scenarios.



(a) Profile IV: Power supply and consumption profiles before the application of service scheduling strategies.



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

FIGURE 16. Profile IV: Power supply and consumption profiles in the HEMS. Parameters: According to Table 4.

According to N , because this parameter is directly related to consumption and has a direct impact on complexity, as analyzed in Section V (Eq. 33), the scenarios are grouped in three categories: (i) small-scale scenarios, for $N \leq 20$ services in which both optimal and heuristic strategies have been analyzed, except for the HEMS in which only heuristics are applied; (ii) large-scale scenarios for $20 \leq N \leq 10^4$ services in which only heuristics are evaluated, including an online scenario for $N = 10^2$ services (i.e., other scenarios in Table 5 are offline); and (iii) very large-scale scenarios for $10^4 \leq N \leq 10^6$ services in which only the prepartitioned versions of *GreedyTs* and *DPTs* are analyzed. In all scenarios, the generation conditions are adapted to the consumption of

N services (i.e., $P_{ES} \geq P_D$). Moreover, in scenarios in which random values (e.g., p_i and d_i) or conditions (e.g., in GATs) are used, the simulations have been repeated 20 or 50 times (depending on the application scope) and considering a confidence interval of 95%, to ensure the quality and stability of results.

C. ANALYSIS OF RESULTS IN SMALL-SCALE SCENARIOS

This section presents the evaluation of the algorithmic scheduling strategies, both optimal and heuristics, in small-scale scenarios (first four scenarios in Table 5) based on the results of performance metrics (mainly AR), running time, and RAM and CPU usage in the simulation domain. To obtain the prepartitioned version of *OptTsCost*, *GreedyTs*, *GATs*, and *DPTs* (i.e., *OptTsCostPart*, *GreedyTsPart*, *GATsPart*, and *DPTsPart*) two partitions have used. Moreover, the results from heuristics are compared with the optimal bounds deliver by *OptTsCost*.

Figure 18 and Table 6 show the simulation results deliver by optimal and heuristics strategies in Scenario I. While, Fig. 19 presents the evaluation of *OptTsCost*, *GreedyTs*, *GATs*, and *DPTs* in Scenarios II, III and IV. The simulation results report that as the value of u_i increases, decreases the values of P_{RES} (e.g., in Fig. 18a) and P_{LACK} (e.g., in Fig. 18c) and increases the value of AR (e.g., in Fig. 18b). These values indicate that through the use of management mechanisms such as time-shifting, the algorithmic strategies can adapt energy demands to availability, being able to make an optimal use/consumption (e.g., 100% in Scenario I) of the energy produced (i.e., P_{RES} minimization) while allowing the processing of services (ECs) that under normal conditions (i.e., without management mechanisms) would be rejected. Consequently, the simulation results indicate that applying the proposed energy model through the algorithmic strategies such as *OptTsCost* promotes better planning, use, and distribution of P_{ES} , as well as potential reduction in peak consumption and energy costs.

Because the metric AR is a direct indicator of energy utilization and service processing, and for practicality in the presentation of results, only the maximum value of this metric (i.e., using the $\max\{u_i\}$, e.g., $\max\{u_i\} = 4$ in

TABLE 5. Summary of simulation parameters.

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

TABLE 6. Performance evaluation of optimal and heuristic strategies in scenario I.

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

TABLE 7. Maximum value of AR achieved by algorithmic strategies in small-case scenarios.

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

Scenario II) is presented for strategies with prepartitioning. Table 7 summarizes the values of AR for all optimal and heuristics strategies in small-scale scenarios. The maximum AR values (Final AR) obtained by the algorithmic strategies in all scenarios show improvements (AR Gain) over 15% (i.e., 15% more services processed) compared to the baseline case (if $u_i = 0$). Depending on the scenario and the strategy used, the improvements can reach 100%, such as in DPTs in Scenario I. Table 7 reports that the heuristic strategy that delivers the best AR gains in all scenarios is GATs (using in all cases a number less than 10 generations), with values that are the same or very similar to those obtained with OptTsCost (whose results are optimal). Moreover, Table 7 shows a

minimum difference between the values of AR produced by the original strategies (i.e., without prepartitioning) and their version with prepartitioning; in the worst case for GATsPart in Scenario II, this difference with GATs is less than 11%.

For the evaluation in a HEMS, only heuristic approaches have been considered due to the complexity of OptTsCost for values of $N > 8$ services or of $u_i > 6$ time slots. This scenario also allows us to analyze the performance of the heuristic strategies for energy demands with different priority levels, as shown in Table 4. Regarding the AR value reached in this scenario, although the percentage of improvement is lower than the rest of the scenarios in Table 4 (between 15%

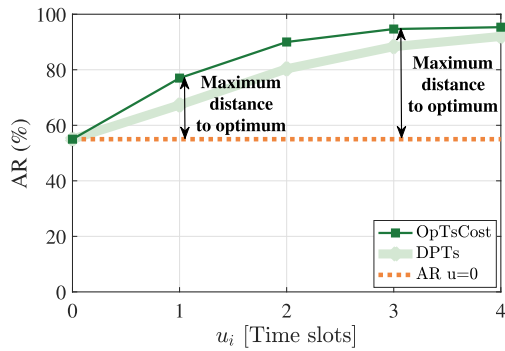


FIGURE 17. Difference of AR values of OptTsCost and DPTs in Scenario II.

to 20%), the adaptation of consumption patterns to P_{ES} , by the heuristic strategies, allows a reduction of peak power by more than 55% (from 8000W down to 3500 W, as shown in Fig. 16b) and obtain an $AR = 100\%$ such as reported the results for GATs.

To quantitatively evaluate the difference, in terms of AR , between the optimal strategy and the heuristics, the criterion of *approximation ratio* (ρ) has been adopted [55]. This parameter estimates how many times bigger the approximate result is compared to the optimal solution. Adapted to the conditions of the proposed energy model, the ρ parameter normalized to the maximum value of time-shifting ($\max\{u_i\}$) is defined by:

$$\rho = 1 - \frac{1}{\max\{u_i\}} \sum_{b=1}^{\max\{u_i\}} \frac{|Opt_b - SubOpt_b|}{disOpt_b} \quad (39)$$

$$disOpt_b = \begin{cases} 1 & \text{if } Opt_b = Opt_0, \\ |Opt_b - Opt_0| & \text{if } Opt_b \neq Opt_0. \end{cases} \quad (40)$$

where, the first term in Eq. 39 represents the optimal solution, while the second term corresponds to the mean absolute error of all time-shifting values, except for $u_i = 0$. In Eq. 39, each absolute error b is weighted to the $disOpt_b$ parameter (Eq. 40), which represents the maximum distance between the optimal value (Opt_b) and the baseline value (Opt_0 , when $u_i = 0$), to obtain the proportional error of each time-shifting b .

The ρ parameter ranges from 0 to 1, and this latter is produced if the optimal and suboptimal values are equal. An intermediate ρ value represents the similarity or closeness factor to the optimal solution (e.g., $DPTs = OptTsCost \cdot \rho$). For a better understanding of ρ factor, Fig. 17 presents an example for DPTs in Scenario II. Eq. 41, shows the analytical computation.

$$\rho_{AR} = 1 - \frac{1}{4} \cdot \left(\frac{77 - 67.3}{77 - 55} + \frac{90 - 80.3}{90 - 55} + \frac{94.6 - 88.3}{94.6 - 55} + \frac{95.3 - 92}{95.3 - 55} \right) = 0.76 \quad (41)$$

The result in Eq. 41 shows that DPTs is similar to OptTsCost in a factor equal to 0.76 (76% similarity), or that DPTs is able to produce a solution that is within ~ 1.3 -the optimal result. Table 8 summarizes the ρ factors for

TABLE 8. Approximation ratio of heuristic strategies in small-scale scenarios.

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

small - scale scenarios and reveal that the heuristics strategies produces near-optimal or optimal solutions and a stable performance. In the worst case ($\rho_{AR} = 0.70$ for DPTs in Scenario II), the heuristic strategy is within only ~ 1.4 - the optimal solution.

Regarding the running time, the evaluation results for the original strategies and those adapted to the prepartitioning method are summarized in Table 9 and Table 10, respectively. The results in Table 9 and Table 10 report that the heuristic strategies are executed (with a relative gain in time G_R) between two and seven orders of magnitude faster than OptTsCost. Moreover, the running time of the prepartitioned versions are slightly higher than the original versions (i.e., of GreedyTs, GATs, and DPTs), this due to the iterative process of the partitions and the subsequent union of partial solutions. This condition indicates that for small-scale scenarios, in terms of running time, it is preferable to apply the original heuristic strategies instead of their versions with prepartitioning.

To better describe the difference in running time and computational resources used by optimal and heuristics strategies, we have performed the analysis for Scenario I considering only the maximum value of time-shifting (i.e., $u_i = 6 \forall S_i$) and varying the services (i.e., $1 \leq N \leq 8$). The evaluation results in Fig. 20 for a single iteration report that: (i) the running time of the heuristics is at least two orders of magnitude less than OptTsCost (see Fig. 20a, as indicated in Table 9 and Table 10, (ii) that the use of the RAM of the heuristic strategies is between 2% and 3% of the amount used by OptTsCost (see Fig. 20b), and (ii) that the CPU usage of the heuristics is between 4% and 40% of the resource used by OptTsCost (see Fig. 20c).

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

D. ANALYSIS OF RESULTS IN LARGE AND VERY-LARGE SCENARIOS

This section presents the evaluation of the heuristic strategies in large-scale and very-large-scale scenarios (scenarios V, VI,

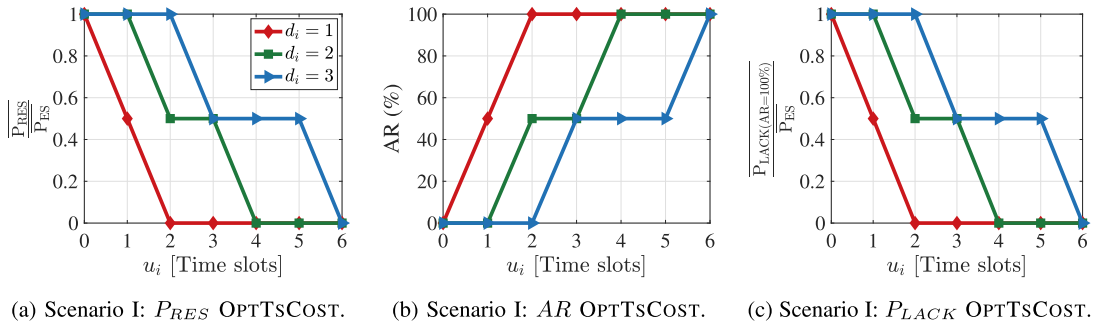


FIGURE 18. Performance evaluation of OptTsCost in Scenario I. Parameters: According to Table 5.

TABLE 9. Running time in seconds of algorithmic strategies and G_R of heuristics concerning OptTsCost in small-case scenarios.

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

TABLE 10. Running time in seconds of algorithmic strategies considering prepartitioning and G_R concerning OptTsCost in small-case scenarios.

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

TABLE 11. Maximum value of AR achieved by heuristic strategies in large-case scenarios.

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

and VII in Table 5) based on the results of performance metrics (mainly AR), running time, and RAM and CPU usage in the simulation domain. To obtain the prepartitioned versions of heuristic strategies, the number of partitions in each scenario and for each value of N is such that the length of each partition is equal to 10 services (e.g., for DPTsPart with $N = 10^4$ services, the $NumPart = 10^3$ partitions). This length has been chosen to produce equal-sized partitions (i.e., all partitions of 10 services). Moreover, based on results in small-scale scenarios, partitions of this length (e.g., in Scenario IV) have demonstrated to produce high-quality solutions. For practicality in the presentation of results, this section only shows the evaluation of the metrics P_{RES} , AR , and P_{LACK} for a single value of N in each scenario (e.g., $N = 10^4$ for Scenario V, $N = 10^3$ for Scenario VI, and $N = 10^6$ for Scenario VII). However, a summary of the evaluation of heuristics for all cases (i.e., all values of N), in terms of AR , running time, and computational

capacity usage, is presented in the corresponding figures and tables.

Figure 21, Table 11, and Table 12 show the simulation results produced by heuristics strategies in large-scale scenarios. Particularly, the results in the top of Fig. 21 report that as the value of u_i increases, decreases the values of the values of P_{RES} (see Fig. 21a) and P_{LACK} (see Fig. 21c) and increases the value of AR (see Fig. 21b). These values indicate that the proposed energy model implemented through the heuristic strategies enables efficient adaptive energy management in large-scale scenarios.

Table 11 summarizes the values of AR and AR gain achieved by heuristics strategies, both the original and prepartitioned versions. These results demonstrate that heuristics deliver improvements in services processing (energy use) ranging from 31% (e.g., GATs in Scenario V for $N = 10^4$) up to 79% (e.g., GreedyTs in Scenario V for $N = 10^4$). The best values of AR in Table 11 are

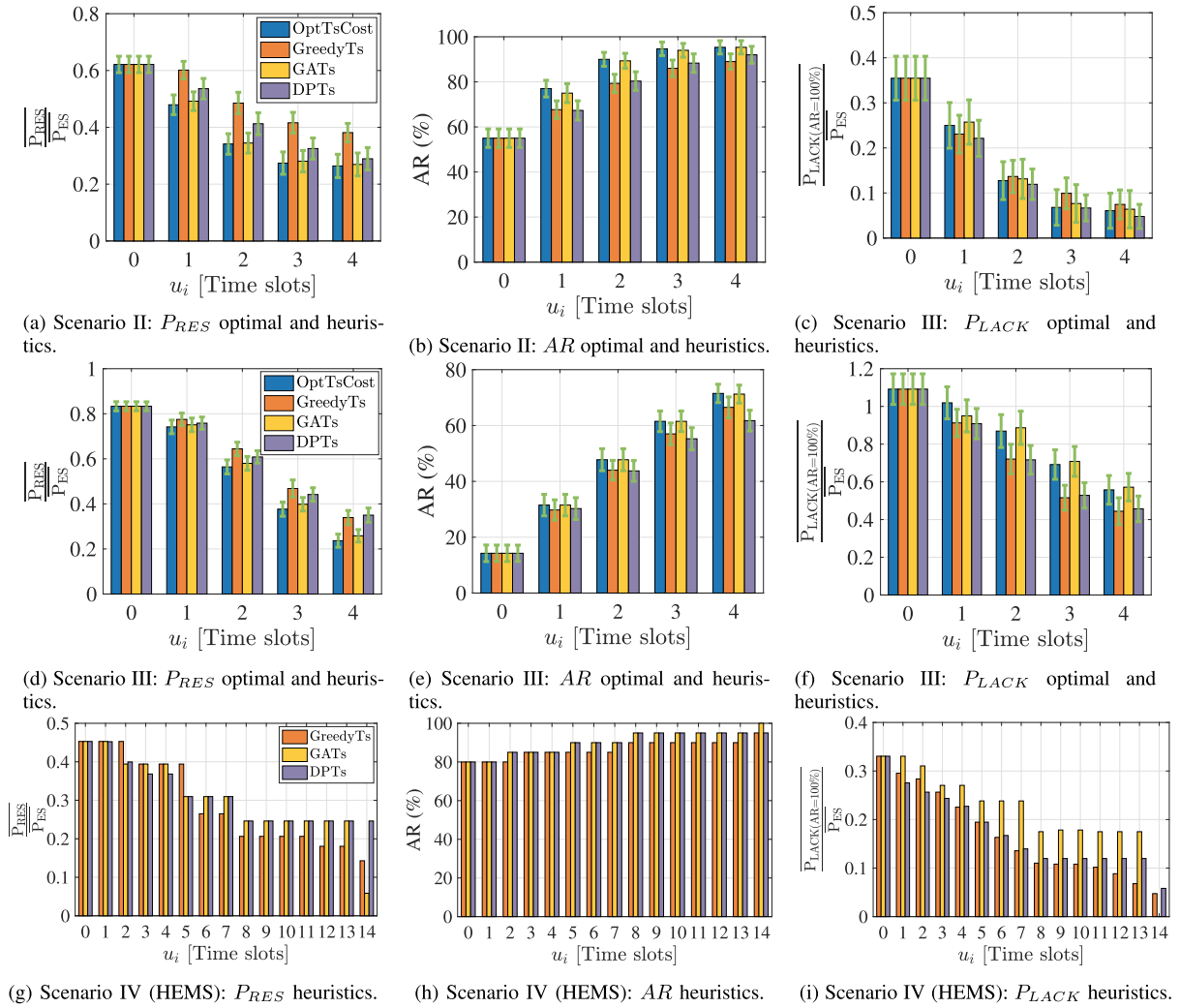


FIGURE 19. Performance evaluation of optimal and heuristic strategies in Scenario II, Scenario III, Scenario IV. Parameters: According to Table 5.

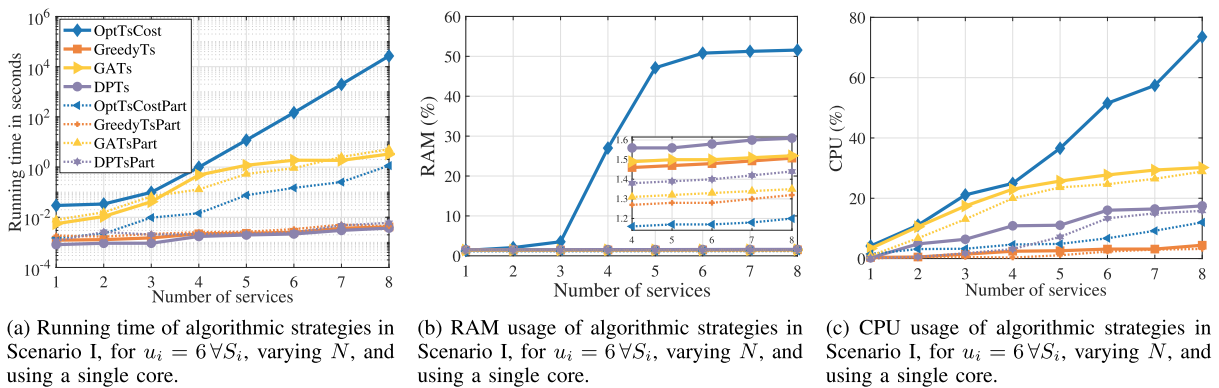


FIGURE 20. Example of performance evaluation of algorithmic strategies according to the running time and RAM and CPU usage in small-scale scenarios. The results in this example have been obtained using a single core (processor).

obtained by strategies GreedyTs and DPTs, while GATs produces the smallest improvements in all cases. Unlike the near-optimal solutions generated by GATs in small-scale scenarios, in large-scale scenarios, this strategy in its original

version has a degraded performance due to the small size of the population ($P_{GA} = 1500$ chromosomes for all scenarios), compared to problem size (especially if $N \geq 10^3$). This shortcoming can be solved by proportionally increasing the

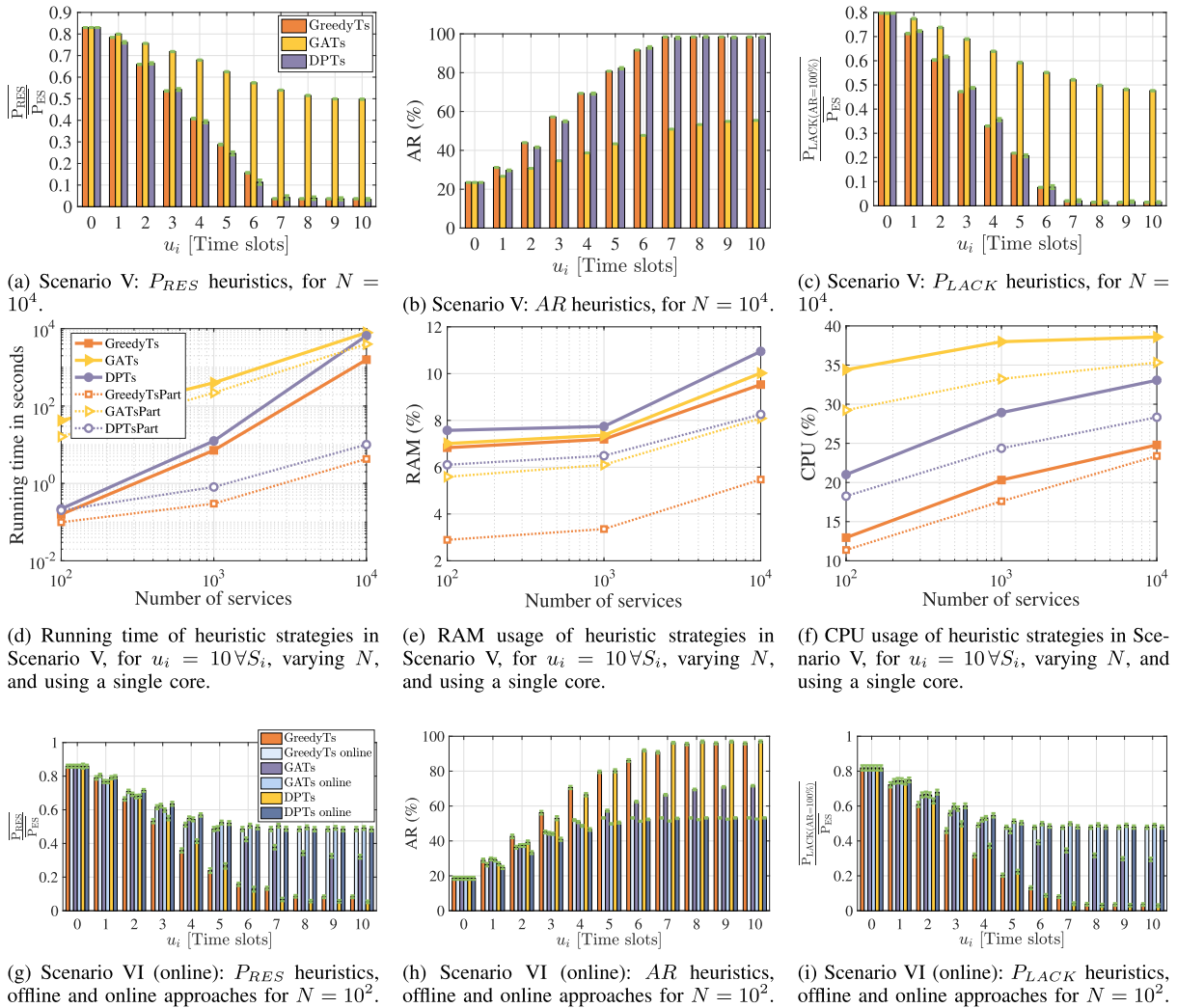


FIGURE 21. Example of performance evaluation of heuristic strategies for large-scale scenarios.

TABLE 12. Running time in seconds of heuristic strategies and G_R of pre-partitioned versions concerning the original version of heuristics in large-scale scenarios.

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

TABLE 13. Maximum value of AR achieved by the heuristic strategies in very-large-scale scenarios.

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

\mathcal{P}_{GA} , although this modification would cause an increase in complexity, as well as of running time and computational capacity demanded. According to the simulation results

in Table 11, we observe that the low performance of GATs, in terms of AR, is overcome if the prepartitioning method is applied to the strategy (i.e., GATsPart produces better values of AR than GATs). This is because at the partition domain a greater search space is available for obtaining better combinations and consequently better quality solutions. The values of AR in Table 11 (for offline approaches) show that GATsPart outperforms GATs by an average of 35%. Whereas the partitioned versions of GreedyTs and DPTs produce values of AR very similar and even the same as their original versions (e.g., AR = 98.01 for DPTs and DPTsPart in Scenario V for $N = 10^4$).

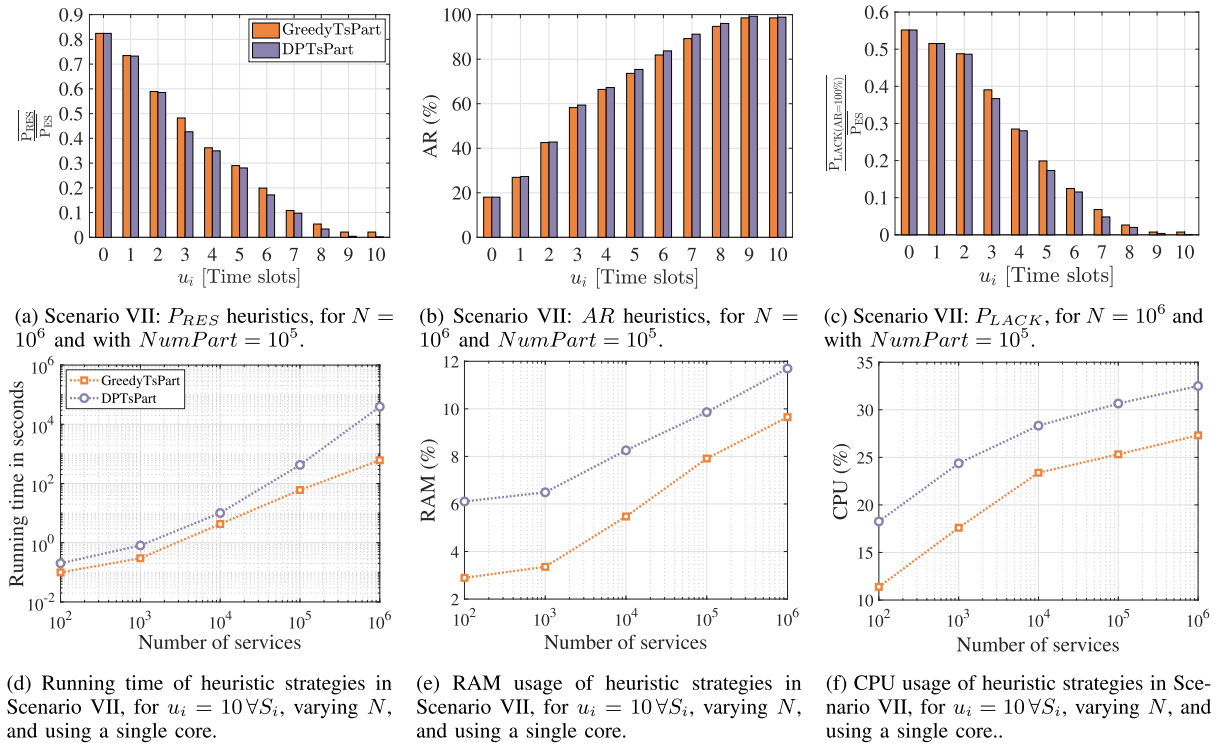


FIGURE 22. Example of performance evaluation of GreedyTsPart and DPTsPart for very-large-scale scenarios.

TABLE 14. Running time in seconds of prepartitioned heuristic strategies in very-large-scale scenarios.

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

The results of metrics P_{RES} (Fig. 21g), AR (Fig. 21h), and P_{LACK} (Fig. 21i) in Scenario VI reveal that heuristic strategies can be applied for adaptive energy management in online approaches. In this scenario, the improvements obtained, in terms of AR as shown in Table 11, are on average approximately 30% and these values, as expected, are lower than those obtained in the offline approach (approximately half), because the algorithms are limited to the use of forward time-shifting. In this scenario, the performance of all the strategies including the prepartitioned ones is similar, the best values of AR are obtained by GreedyTs and DPTs while the worst AR metric is generated by GATsPart, although the difference in the results is less than 1% (eg., $AR = 34.60\%$ for DPTs and $AR = 33.96\%$ for GATsPart). A feature that can be analyzed in future work is the incorporation of forecasting methods of energy supply in the algorithmic strategies to improve the service scheduling and, consequently, the AR metric.

Table 12 shows that the running time of the heuristic strategies and the G_R computed from the ratio between the heuristics and their prepartitioned versions. These results

report that the partitioned strategies are executed in less time than the original versions with a difference of up to three orders of magnitude (e.g., DPTsPart in Scenario V for $N = 10^4$). The lowest running time values are obtained by GreedyTsPart and DPTsPart, which reveals their potential applicability to larger scenarios (i.e., for $N > 10^4$). To better describe the difference in running time and computational resources used by heuristics strategies, we present the analysis for Scenario V considering only the maximum value of time-shifting (i.e., $u_i = 10 \forall S_i$) and varying the services (i.e., $10^2 \leq N \leq 10^4$). The evaluation results in Fig. 21d, Fig. 21e, and Fig. 21f for a single iteration report that: (i) the strategies with prepartitioning are executed in lower running time and use less RAM and CPU capacity than the original version of heuristics; (ii) the application of GATs, GATsPart, GreedyTs, and DPTs is limited to a maximum of $N = 10^3$ services, because for larger scenarios (e.g., for $N = 10^4$) the running time is around units of hours; (iii) the computational capacity used by the heuristics is between 3% and 11% for RAM (see Fig. 21e), and between 4% and 40% for CPU (see Fig. 21f); and (iv) the overall evaluation in terms of running time, RAM and CPU usage, demonstrate that the best strategies for large-scale scenarios are GreedyTsPart and DPTsPart.

Simulations in large-scale scenarios verify the validity of the proposed energy management model and heuristic strategies developed. Particularly, evaluation results demonstrate that the prepartitioning method improve the operation of heuristics in terms of running time, and RAM and CPU

TABLE 15. Summary of features, mean values of metrics, and applicability of service scheduling algorithmic strategies.

Strategy	Features	Version	Performance metrics small-scale scenarios			Application scope		
			AR gain (%)	RAM usage (%) single core	CPU usage (%) single core	Small-scale scenarios $N \leq 20$	Large-scale scenarios $20 < N \leq 10^4$	Very large-scale scenarios $10^4 < N \leq 10^6$
Optimal approach	<ul style="list-style-type: none"> - Exhaustive brute-force search-based algorithm. - Optimal service scheduling is based on the selection of the optimal combination of variations. - Easy implementation but high computational consumption mainly in term of RAM and CPU. - Exponential growth of complexity depending on the values of N and u_i. 	OPTTsCOST	65.86	29.34	35.04	Yes constrained to $N \leq 10$	No	No
		OPTTsCOSTPART	65.42	1.16	5.68	Yes	Yes constrained to $N \leq 10^3$	No
Greedy approach	<ul style="list-style-type: none"> - Iterative and constructive algorithm. - Suboptimal service scheduling based on the progressive selection of the best variation of services (one service analyzed at a time, which causes a reduction of search space). - Easy implementation and different feasible alternatives (some implementations will do better than others) depending on the criteria for choosing subsequent variations (e.g., selection of variation that produces the minimum value of P_{RES} in the analyzed time horizon). - Quality of solutions can vary widely on the input (i.e., based on the criteria for selecting the initial and subsequent variations). - Good running-time performance compared to the optimal solution and pseudo-polynomial growth of complexity. It requires less computational resources than the optimal approach. - This strategy can be used as a benchmark for the development of more sophisticated approaches. 	GREEDYTs	62.08	1.46	2.26	Yes	Yes but suggested up to $N = 10^3$	No
		GREEDYTsPART	61.97	1.28	1.38	Yes	Yes	Yes
Genetic algorithm	<ul style="list-style-type: none"> - Strategy inspired by the evolution of biological systems and the survival of the fittest. - Near-optimal (or even optimal) service scheduling based on the selection of the fittest combinations of variations (i.e., the individual or chromosomes that produces the minimum value of P_{RES}) that evolve over generations. - Medium complexity implementation because of mechanisms such as prioritization and time-shifting on services. - Nondeterministic generation of solutions due to the random values related to the creation of population and application of genetic operations (crossover and mutation). - Good running-time performance and less computational capacity used compared to the optimal approach. Still, the final solution may require many generations, especially if parameters for population creation or stop criteria are not properly established. In simulations, all solutions are obtained in a range lower than 10 generations. 	GATs	65.77	1.49	20.91	Yes	Yes but suggested up to $N = 10^3$	No
		GATsPART	63.92	1.32	18.03	Yes	Yes but suggested up to $N = 10^3$	No
Dynamic programming	<ul style="list-style-type: none"> - Strategy based on dynamic programming method. - Suboptimal service scheduling based on the progressive selection of the best variations of services. Using a bottom-up approach, systematically, the results of a previous set of variations are stored and used to solve a greater set of simultaneous services. - High quality (feasible) solutions for a variety of scenarios of different sizes. - Good running-time performance and less use of computational capacity compared to the optimal approach. - Difficult implementation due to the two-dimensional condition of adaptive energy management (power and time) and multiple-choice analysis due to the application of time-shifting (variations of services). Moreover, this strategy requires integer values regarding P_{ES} and energy demands. 	DPTs	61.50	1.56	10.42	Yes	Yes but suggested up to $N = 10^3$	No
		DPTsPART	60.72	1.40	7.11	Yes	Yes	Yes but suggested up to $N = 10^5$

usage, which makes these strategies (mainly GreedyTsPart and DPTsPart) have applicability to larger scenarios (i.e., for $N > 10^4$). Moreover, the performance metrics values indicate that the prepartitioning method not only can extend the scalability of heuristics but also it can improve the inner operation of the proposed algorithms as in the case of GATS for which its prepartitioned version (i.e., GATsPart) produce better values of AR .

For the evaluation in very-large-scale scenarios, GreedyTsPart and DPTsPart have been chosen, due to their computational capacity usage (less than 9% of RAM and 29% of CPU) and running time (less than 11 seconds) delivered in large-scale scenarios. Figure 22, Table 13, and Table 14 summarize the simulation results produced by these heuristics. Like the results obtained in smaller scenarios, for $10^5 \leq N \leq 10^6$, the values of metrics P_{RES} (Fig. 22a), AR (Fig. 22b), and P_{LACK} (Fig. 22c) verify the effectiveness of GreedyTsPart and DPTsPart to adapt consumption to the availability, which results in minimization of P_{RES} . As indicated in Table 13, the improvements, in terms of AR , achieved by the two heuristics are very similar to each other and are around 74%; although their running times differ as indicated in Table 14 and reveal that the application of DPTsPart is limited to scenarios with $N < 10^6$.

To better differentiate the performance of GreedyTsPart and DPTsPart in very-large-scale scenarios, we have analyzed Scenario VII considering only the maximum value of time-shifting (i.e., $u_i = 10 \forall S_i$) and varying N , as shown in the second row in Fig. 22. Simulations results in Fig. 22d, Fig. 22e, and Fig. 22f show that in terms of running time and use of RAM and CPU, GreedyTsPart presents a better performance than DPTsPart. Specifically, the running time of DPTsPart for $N = 10^6$ in order of tens of hours, confirm its applicability for scenarios with $N \leq 10^5$. Regarding the computational capacity usage, Fig. 22e and Fig. 22f report that the heuristic strategies consume between 3% and 12% of RAM, and between 11% and 32% of CPU, respectively. These values together with the results of the running-time demonstrate feasibility of adaptive energy management in scenarios with hundreds of thousands or even millions of services.

The simulations in large-scale and very large-scale demonstrate that the heuristics strategies, when deployed in the NFV domain (or a similar ICT infrastructure), can enable efficient adaptive energy management within reasonable running time and use of computational capacity (mainly in terms of RAM and CPU). Table 15 summarizes the main operating features, the mean values of AR and computation capacity used (considering the evaluation in small-scale scenarios for a fair comparison), and the applicability of the proposed service scheduling strategies (both optimal and heuristic).

VIII. CONCLUSION

This paper provides insight into optimal and heuristic strategies that can be deployed in the NFV domain (or a similar ICT infrastructure) for achieving adaptive energy management,

either renewable or not, in small, large, or very large-scale IoT-enabled scenarios. In this context, this paper starts describing the NFV-enabled energy management scheme, the stakeholders involved, and the management mechanisms, including energy provisioning prioritization, time-shifting application, and rejection of energy demands, to adapt consumption to available supply. Moreover, the proposal for energy management includes the mathematical model related to adaptive consumption and an algorithmic strategy based on brute-force search, denoted as OptTsCost, to solve the energy management model optimally. The optimal strategy allows us to identify all concerns related to the algorithmic implementation of adaptive energy management.

Given the NP-Hard nature of adaptive energy management and the exponential growth of OptTsCost (which depends on the values of N and u_i as shown in Eq. 33), we propose three heuristic strategies identified as GreedyTs, GATs, and DPTs, which are based on a greedy approach, an evolutionary algorithm, and dynamic programming method, respectively. To scale up the applicability of adaptive energy management to scenarios with thousands and hundreds of thousands of energy demands, we have incorporated a prepartitioning method for both the optimal strategy and the heuristics. As a result of the prepartitioning method, four additional heuristics were created and are denoted as OptTsCostPat, GreedyTsPart, GATsPart, and DPTsPart.

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

In terms of quality and complexity of solutions, simulation results indicate that the heuristic strategies, both the originals and the versions with prepartitioning, produce high-quality solutions while performing between two and six orders of magnitude faster than the optimal approach OptTsCost

(as shown in Table 9). Regarding the computational power demanded by the heuristics, the evaluation of heuristics indicates that these strategies only use a fraction of RAM and CPU capacity used by OptTsCost (as shown in Fig 20). In the worst case scenario for RAM usage, DPTs uses approximately 5% of capacity used by OptTsCost, and in the worst case scenario for CPU usage, GATs uses approximately 60% of capacity used by OptTsCost. Therefore, reduced running-time and computational capacity usage make possible the implementation of heuristics on advanced NFV-enabled infrastructures or on embedded systems in homes (e.g., on a Raspberry Pi platform) for adaptive energy management in IoT-enabled scenarios with hundreds or thousands of services. Likewise, the application of the prepartitioning method allows the energy model to extend the potentialities of the heuristics to IoT-enabled environments with hundreds of thousand of energy demands (as in the current and future communications infrastructures) or as future initiatives of the IoE.

Future work can address a variety of aspects related to the energy management model or the development of more sophisticated strategies. Possible improvements for the energy model include: (i) the incorporation of a parameter that enables the variation of consumption over time, so that a service can increase or decrease consumption on availability; (ii) the possible processing of partial services, in this case, the proposed model can be based a fractional knapsack problem; and (iii) the incorporation of a parameter that represents the possible storage of energy (i.e., the use of battery units in the model), so that in energy surplus the energy can be stored and used when needed. Concerning future algorithmic solutions, a first step might be the development of a hybrid strategy in which, depending on the size of the scenario, the proposed algorithm selects the appropriate strategy (e.g., use of GATs for small-scenarios, DPTs for large-scale scenarios, and use of GreedyTsPart for very large-scale IoT environments). The use of machine learning techniques can be explored for adaptive energy management. For instance, supervised learning can be applied to implement a prediction method of produced energy resource or consumption, which can be used as an input to the proposed energy model and improve its performance. The clustering of consumption patterns based on unsupervised learning can be applied to the energy model to guarantee the energy supply for CS. Moreover, a complete adaptive energy management model could be established using a reinforcement learning approach, as inspired by the multi-armed bandits problem.

REFERENCES

- [1] F. Birol, *World Energy Outlook 2013*. Paris, France: International Energy Agency, 2013.
- [2] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale, "The future renewable electric energy delivery and management (FREEDM) system: The energy internet," *Proc. IEEE*, vol. 99, no. 1, pp. 133–148, Jan. 2011.
- [3] Google. (2018). *Google Environmental Report*. [Online]. Available: <https://cloud.google.com/sustainability>
- [4] *Clicking Clean: Who is Winning the Race to Build a Green Internet?*, Greenpeace, Amsterdam, The Netherlands, 2017.
- [5] The New York Times. (Aug. 2016). *Apple Becomes a Green Energy Supplier, With Itself as Customer*. [Online]. Available: <https://www.nytimes.com/2016/08/24/business/energy-environment/apple-use-rises-corporations-turn-to-their-own-green-utility-sources.html>
- [6] V. Chamola and B. Sikdar, "Solar powered cellular base stations: Current scenario, issues and proposed solutions," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 108–114, May 2016.
- [7] J. Medina, N. Muller, and I. Roytelman, "Demand response and distribution grid operations: Opportunities and challenges," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 193–198, Sep. 2010.
- [8] C. Tipantuña and X. Hesselbach, "NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 357–380, 2020.
- [9] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, Cisco, San Jose, CA, USA, 2018, vol. 1.
- [10] R. Basmadjian, J. F. Botero, G. Giuliani, X. Hesselbach, S. Klingert, and H. De Meer, "Making data centers fit for demand response: Introducing GreenSDA and GreenSLA contracts," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3453–3464, Jul. 2018.
- [11] ETSI. (2012). *Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action, ETSI White Paper*. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf
- [12] C. Tipantuña and X. Hesselbach, "Demand-response power management strategy using time shifting capabilities," in *Proc. 9th Int. Conf. Future Energy Syst.*, Jun. 2018, pp. 480–485. [Online]. Available: <https://dl.acm.org/citation.cfm?id=3213519>
- [13] D. Pisinger, "Algorithms for knapsack problems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Copenhagen, Copenhagen, Denmark, 1995. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.9780> and <http://hjemmesider.diku.dk/~pisinger/95-1.pdf>
- [14] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.
- [15] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1098–1116, May 2019.
- [16] N. Huin, A. Tomassilli, F. Giroire, and B. Jaumard, "Energy-efficient service function chain provisioning," *J. Opt. Commun. Netw.*, vol. 10, no. 3, pp. 114–124, Mar. 2018.
- [17] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in NFV architectures," *IEEE Access*, vol. 5, pp. 4927–4938, 2017.
- [18] A. Wantamane, S. Watarakitpaisarn, G. Carella, C. Aswakul, and T. Magedanz, "Virtualising machine to machine (M2M) application using open baton as NFV-compliant framework for building energy management system," in *Proc. 11th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2016, pp. 199–204.
- [19] X. Li, R. Ferdous, C. F. Chiasserini, C. E. Casetti, F. Moscatelli, G. Landi, R. Casellas, K. Sakaguchi, S. B. Chundrigar, R. Vilalta, J. Mangues, A. Garcia-Saavedra, X. Costa-Perez, L. Goratti, and D. Siracusa, "Novel resource and energy management for 5G integrated backhaul/fronthaul (5G-crosshaul)," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 778–784.
- [20] A. N. Al-Quzweeni, A. Q. Lawey, T. E. H. Elgorashi, and J. M. H. Elmoghani, "Optimized energy aware 5G network function virtualization," *IEEE Access*, vol. 7, pp. 44939–44958, 2019.
- [21] C. Tipantuña, X. Hesselbach, V. Sánchez-Aguero, F. Valera, I. Vidal, and B. Nogales, "An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–20, Feb. 2019.
- [22] M. Niedermeier and H. de Meer, "Constructing dependable smart grid networks using network functions virtualization," *J. Netw. Syst. Manage.*, vol. 24, no. 3, pp. 449–469, Jul. 2016.
- [23] H. C. Leligou, T. Zahariadis, L. Sarakis, E. Tsampasis, A. Voulkidis, and T. E. Velivassaki, "Smart grid: A demanding use case for 5G technologies," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 215–220.
- [24] D. Yang, S. Meng, Z. Wang, S. Wu, X. Li, X. Cao, X. Liu, X. Wang, and Y. Chen, "E2E delay optimization for smart grids mission-critical slices in core networks," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6.

- [25] Ardiansyah, Y. Choi, M. R. K. Aziz, K. Cho, and D. Choi, "Latency-optimal network intelligence services in SDN/NFV-based energy internet cyberinfrastructure," *IEEE Access*, vol. 8, pp. 4485–4499, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8945310>, doi: 10.1109/ACCESS.2019.2963139.
- [26] B. Huang, X. Bai, Z. Zhou, Q. Cui, D. Zhu, and R. Hu, "Energy informatics: Fundamentals and standardization," *ICT Exp.*, vol. 3, no. 2, pp. 76–80, Jun. 2017.
- [27] B. Shakerighadi, A. Anvari-Moghaddam, J. Vasquez, and J. Guerrero, "Internet of Things for modern energy systems: State-of-the-art, challenges, and open issues," *Energies*, vol. 11, no. 5, p. 1252, May 2018.
- [28] L. Barbierato, A. Estebasari, E. Pons, M. Pau, F. Salassa, M. Ghirardi, and E. Patti, "A distributed IoT infrastructure to test and deploy real-time demand response in smart grids," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1136–1146, Feb. 2019.
- [29] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and challenges for data center demand response," in *Proc. Int. Green Comput. Conf.*, Nov. 2014, pp. 1–10.
- [30] N. Good, K. A. Ellis, and P. Mancarella, "Review and classification of barriers and enablers of demand response in the smart grid," *Renew. Sustain. Energy Rev.*, vol. 72, pp. 57–72, May 2017.
- [31] M. Wei, S. H. Hong, and M. Alam, "An IoT-based energy-management platform for industrial facilities," *Appl. Energy*, vol. 164, pp. 607–619, Feb. 2016.
- [32] F. AlFaris, A. Juaidi, and F. Manzano-Agugliaro, "Intelligent homes' technologies to optimize the energy performance for the net zero energy home," *Energy Buildings*, vol. 153, pp. 262–274, Oct. 2017.
- [33] H. K. Nguyen, J. B. Song, and Z. Han, "Demand side management to reduce peak-to-average ratio using game theory in smart grid," in *Proc. IEEE INFOCOM Workshops*, Mar. 2012, pp. 91–96.
- [34] Z. Zhao, W. C. Lee, Y. Shin, and K.-B. Song, "An optimal power scheduling method for demand response in home energy management system," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1391–1400, Sep. 2013.
- [35] P. Chavali, P. Yang, and A. Nehorai, "A distributed algorithm of appliance scheduling for home energy management system," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 282–290, Jan. 2014.
- [36] S. Kazmi, N. Javaid, M. J. Mughal, M. Akbar, S. H. Ahmed, and N. Alrajeh, "Towards optimization of metaheuristic algorithms for IoT enabled smart Homes targeting balanced demand and supply of energy," *IEEE Access*, vol. 7, pp. 24267–24281, 2019.
- [37] A. J. Conejo, J. M. Morales, and L. Baringo, "Real-time demand response model," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec. 2010.
- [38] B. Hussain, Q. U. Hasan, N. Javaid, M. Guizani, A. Almgren, and A. Alamri, "An innovative heuristic algorithm for IoT-enabled smart homes for developing countries," *IEEE Access*, vol. 6, pp. 15550–15575, 2018.
- [39] Y. Wang, S. Mao, and R. M. Nelms, "Online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 10–21, Jun. 2013.
- [40] ETSI. (2015). *Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework*. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFVEVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf
- [41] H. Kellerer, U. Pferschy, and D. Pisinger, "Multidimensional knapsack problems," in *Knapsack Problems*. Berlin, Germany: Springer, 2004, pp. 235–283. [Online]. Available: <https://link.springer.com/chapter/10.1007/978-3-540-24777-79>
- [42] P. P. Marino, *Optimization of Computer Networks: Modeling and Algorithms: A Hands-On Approach*. Hoboken, NJ, USA: Wiley, 2016.
- [43] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization: Methods and Case Studies*. Berlin, Germany: Springer-Verlag, 2006. [Online]. Available: <https://www.springer.com/gp/book/9783540230229>
- [44] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
- [45] E. Y. H. Lin and M. C. Chen, "A dynamic programming approach to the multiple-choice multi-period knapsack problem and the recursive APL2 code," *J. Inf. Optim. Sci.*, vol. 31, no. 2, pp. 289–303, Mar. 2010.
- [46] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," *Eur. J. Oper. Res.*, vol. 123, no. 2, pp. 394–407, 2000.
- [47] D. Bertsimas and R. Demir, "An approximate dynamic programming approach to multidimensional knapsack problems," *Manage. Sci.*, vol. 48, no. 4, pp. 550–565, Apr. 2002.
- [48] H. Kellerer, U. Pferschy, and D. Pisinger, "The multiple-choice knapsack problem," in *Knapsack Problems*. Berlin, Germany: Springer, 2004, pp. 317–347. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-540-24777-7_11
- [49] B. B. M. Shao and H. R. Rao, "A parallel hypercube algorithm for discrete resource allocation problems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 1, pp. 233–242, Jan. 2005.
- [50] C. Tipantuña and X. Hesselbach. (Aug. 2020). *Heuristic Strategies for NFV-Enabled Renewable and Non-Renewable Energy Management in the Future IoT World Repository*. [Online]. Available: <https://github.com/ChristianTipantuña/NFV-EnabledHeuristicStrategiesIoT>
- [51] E. Mudaheranwa, Y.-O. Udoakah, and L. Cipcigan, "Rwanda's energy profile and potential renewable energy resources mapping toward sustainable development goals," in *Proc. IEEE PES/IAS PowerAfrica*, Aug. 2019, pp. 533–538.
- [52] Y. M. Ding, S. H. Hong, and X. H. Li, "A demand response energy management scheme for industrial facilities in smart grid," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2257–2269, Nov. 2014.
- [53] O. Ayan and B. Turkay, "Energy management algorithm for peak demand reduction," in *Proc. 20th Int. Symp. Electr. App. Technol. (SIELA)*, Jun. 2018, pp. 1–4.
- [54] I. Richardson, M. Thomson, D. Infield, and C. Clifford, "Domestic electricity use: A high-resolution energy demand model," *Energy Buildings*, vol. 42, no. 10, pp. 1878–1887, 2010.
- [55] V. V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer-Verlag, 2013. [Online]. Available: <https://www.springer.com/gp/book/9783540653677>



CHRISTIAN TIPANTUÑA (Member, IEEE) received the bachelor's degree in telecommunications engineering from the Escuela Politécnica Nacional, Ecuador, in 2011, and the M.Sc. degree in wireless systems and related technologies from the Politecnico di Torino, Turin, Italy, in 2013. He is currently pursuing the Ph.D. degree in network engineering with the Universitat Politècnica de Catalunya, Barcelona, Spain. He is also a member of the Grupo de Investigación en Redes Inalámbricas, Escuela Politécnica Nacional. His research interests include network functions virtualization, green networking, and software defined networking.



XAVIER HESSELBACH (Senior Member, IEEE) received the M.S. degree (Hons.) in telecommunications engineering and the Ph.D. degree (Hons.) from the Universitat Politècnica de Catalunya (UPC), in 1994 and 1999, respectively. He is currently an Associate Professor with the Department of Network Engineering, UPC. He has taken part in several European and Spanish research projects, such as the EuroNGI/FGI/NF network of Excellence, COST293, Mantychore, and All4Green. He has been involved in several national and international projects, and has authored four books and more than 60 national and international publications in conferences and journals. His research interests include networks virtualization, resources management, broadband networks, quality of service, and green networking. In 1994, he received the award from the COIT/AEIT of Spain for the Best Master Thesis on Networks and Telecommunication Services.



WALTER UNGER received the Diploma and Ph.D. degrees (Hons.) from Paderborn University, in 1986 and 1990, respectively. He changed to RWTH Aachen University, in 1998, and received the postdoctoral lecture qualification (PD), in 2002. He is currently a PD (an Associate Professor) with the Department of Computer Science, RWTH Aachen University. He is a knight of Schlaraffia. He has taken part in several European research projects, such as the HCM and COST293.

He has authored more than 60 international publications in conferences and journals. His research interests include graph theory, algorithms, and complexity theory.

• • •