UNIVERSIDAD DE MURCIA

Escuela de Doctorado

TESIS DOCTORAL

Análisis y Diseño de soluciones para el despliegue de nodos IoT y Edge activos en el Computing Continuum

AUTOR/A D. Alberto Robles Enciso

DIRECTOR/ES Dr. Antonio Fernando Skarmeta Gómez

2025

UNIVERSIDAD DE MURCIA Escuela de Doctorado

TESIS DOCTORAL

Análisis y Diseño de soluciones para el despliegue de nodos IoT y Edge activos en el Computing Continuum

AUTOR/A Alberto Robles Enciso

DIRECTOR/ES Dr. Antonio Fernando Skarmeta Gómez

2025

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA EN MODALIDAD DE COMPENDIO O ARTÍCULOS PARA OBTENER EL TITULO DE DOCTOR/A

Aprobado por la Comisión General de Doctorado el 19 de octubre de 2022.

Yo, D. Alberto Robles Enciso, habiendo cursado el Programa de Doctorado en Informática de la Escuela Internacional de Doctorado de la Universidad de Murcia (EIDUM), como autor/a de la tesis presentada para la obtención del título de Doctor/a titulada:

Análisis y Diseño de soluciones para el despliegue de nodos IoT y Edge activos en el Computing Continuum

y dirigida por:

D.: Antonio F. Skarmeta Gómez D.: D.:

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Además, al haber sido autorizada como compendio de publicaciones, cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

Murcia, a 03 de febrero de 2025

Alberto Robles Enciso

Información básica sobre protección de sus datos personales aportados:	
Responsable	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad	Gestionar su declaración de autoría y originalidad
Destinatarios	No se prevén comunicaciones de datos
Derechos	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor/a.

Agradecimientos

En primer lugar, quiero agradecer a la Fundación Séneca por financiar mi proyecto de tesis. También a Viviane, que siempre estuvo ahí para ayudarme con todos los temas administrativos de la beca, haciéndome las cosas mucho más fáciles.

De igual forma quiero agradecer la labor de mi director de tesis, Antonio F. Skarmeta Gómez, por su papel fundamental en guiar mis "palos de ciego" hacia temas relacionados con mi tesis. También valoro mucho todo lo que he aprendido de él y las oportunidades que me ha brindado para explorar otras áreas dentro de los proyectos del grupo de investigación. Pero, sobre todo, agradezco su diligencia para resolver cualquier problema que surgiera durante el desarrollo de mi tesis. Sé que su agenda suele estar bastante ocupada, y yo tampoco se lo pongo fácil con mis dudas, pero al final siempre logramos solucionarlo todo con un generoso intercambio exprés de correos.

Por otro lado, querría hacer una mención especial a mis profesores de Física y Química del instituto, María Dolores y José Antonio, por ser una inspiración para mi vocación de la enseñanza y también por orientarme y centrarme en estudiar los temas que me apasionan. Además, en mis últimos años de instituto tuve la suerte de coincidir con el que en ese momento era vicedecano de estudios de la facultad, Juan Antonio Sánchez Laguna. Fue él quien me animó a participar en las olimpiadas de programación y respondió pacientemente todas mis dudas sobre la carrera y la programación. Muchas gracias por esa coincidencia que fue crucial para que me enfocase en la programación y los algoritmos.

Volviendo a mis años como doctorando, un detalle que agradezco es haber tenido la oportunidad de ser profesor durante cuatro años de una asignatura que he disfrutado enormemente: Fundamentos Físicos de la Informática (también conocida como Electrónica). Para mí, haber dado clase es, sin duda, lo más valioso que me llevo de esta etapa. Mi director bien sabe que, si tuviera que elegir entre investigar y enseñar, la respuesta sería más que obvia. Por esta misma razón agradezco a Miguel Ángel y a Alfonso por haberme hecho participe pleno de las prácticas de la asignatura, pudiendo proponer todo tipo de cambios y con la libertad de dar las clases a mi manera con algún que otro chispazo de condensadores o baterías de vez en cuando...

En lo personal, me gustaría agradecer a mis familiares y amigos por ser un apoyo constante, especialmente por soportar mis quejas continuas sobre el doctorado.

A mis padres Manuel y Pilar por habérmelo dado todo, mi madre por ser la alegría

de la casa y mi padre por haberme enseñado desde pequeño todos los entresijos de la informática.

A mi hermano Manolo por ser el que dio lugar a mi pasión por las matemáticas desde bien pequeño. Por supuesto a mi gemelo Ricardo por ser la otra mitad de todos mis proyectos e ideas, los dos por separado tenemos potencial pero juntos logramos todo lo que nos propongamos.

También a mi pareja Lorena, compañera continua de todas mis vivencias, mis altibajos y grandes entusiasmos, compartiendo conmigo todo aquello que me apasiona. Desdichadamente sufridora del mismo dolor que yo, hacer una tesis doctoral, pero que con gran fortaleza sigue adelante.

Por último me queda agradecer, y cabe decir de manera taxativa que es el mayor agradecimiento de todos, a mi gato *Hale*, o como yo lo llamo cariñosamente *El Verde*. Su compañía y cariño durante estos 15 años han sido incomparables, siendo testigo de prácticamente todas las etapas de mi vida desde que tenía 13 años. Me vio dar mis primeros pasos en el mundo de las redes, Linux y la programación, estuvo conmigo cuando escribí mis primeras líneas de código en la ESO, cuando elegí el tipo de bachillerato, me preparé para la selectividad, decidí mi carrera, estudié durante la universidad, obtuve mi máster y, ahora, al terminar mi doctorado. Podría decir, sin exagerar, que cada uno de mis títulos académicos también le pertenece a él, quien ahora, con mucho orgullo, se convierte en Doctor junto conmigo.

Índice

Listado de Figuras

Listado de Tablas

Lista de publicaciones

Esta Tesis Doctoral se presenta como un compendio de las siguientes publicaciones, siendo el doctorando el autor principal en todas ellas:

- Robles-Enciso, Alberto & Skarmeta, Antonio. (2022). A multi-layer guided reinforcement learning-based tasks offloading in edge computing. Computer Networks. 220. 109476. 10.1016/j.comnet.2022.109476.

- Robles-Enciso, Alberto & Robles-Enciso, Ricardo & Skarmeta, Antonio. (2023). Sim-PowerCS: An extensible and simplified open-source energy simulator. SoftwareX. 23. 101467. 10.1016/j.softx.2023.101467.

- Robles-Enciso, Alberto & Skarmeta, Antonio. (2024). Adapting Containerized Workloads for the Continuum Computing. IEEE Access. 12. 104102-104114. 10.1109/ACCESS.2024.3434585.

De igual forma, el doctorando también ha sido autor principal o colaborador en las siguientes publicaciones tanto de revistas como de congresos que han quedado fuera del compendio:

- Robles-Enciso, Alberto & Robles-Enciso, Ricardo & Skarmeta Gómez, Antonio. (2024). An Adaptive Energy Orchestrator for Cyberphysical Systems Using Multi-agent Reinforcement Learning. Smart Cities 2024, 7, 3210-3240. 10.3390/smartcities7060125

- Robles-Enciso, Alberto & Bernabé Murcia, José Manuel & Molina Zarca, Alejandro & Skarmeta Gomez, Antonio. (2024). Dynamic Multi-Method Allocation for Intent-based Security Orchestration. Journal of Network and Systems Management 33, 1 (Jan 2025). 10.1007/s10922-024-09896-8

- Gonzalez-Gil, Pedro & Robles-Enciso, Alberto & Martínez, Juan Antonio & Skarmeta Gomez, Antonio. (2024). Architecture for Orchestrating Dynamic DNN-Powered Image Processing Tasks in Edge and Cloud Devices. IEEE Access, vol. 9, pp. 107137-107148. 10.1109/ACCESS.2021.3101306.

- Robles-Enciso, Alberto, Skarmeta, Antonio. (2022). Task Offloading in Computing Continuum Using Collaborative Reinforcement Learning. Internet of Things. GIoTS 2022. Lecture Notes in Computer Science, vol 13533. Springer, Cham. 10.1007/978-3-031-20936-9_7

- Robles-Enciso, Alberto, Robles-Enciso, Ricardo., Skarmeta, Antonio. (2024). Multi-agent Reinforcement Learning-Based Energy Orchestrator for Cyber-Physical Systems. Algorithmic Aspects of Cloud Computing. ALGOCLOUD 2023. Lecture Notes in Computer Science, vol 14053. Springer, Cham. 10.1007/978-3-031-49361-4_6

Capítulo 1

Abstract

In today's information age, data is a fundamental resource, and its efficient collection and processing is essential. Thanks to advances in hardware technologies and the interest to digitise our society, connecting everything around us to the Internet is a promising development. In this way, the promising technology of the Internet of Things (IoT) arises, which aims to connect common everyday devices (*Things*) to the Internet and collect all kinds of data they produce. These connected devices make up what is called an IoT network, in which each of them has an internet connection and in some cases are able to communicate with each other, this allows new paradigms that facilitate scenarios such as the deployment of dozens of sensors and devices in large facilities (homes, factories, cities, etc.) to be consulted in real time from anywhere in the world.

Alongside the Internet of Things, other particularly useful technologies and techniques have been emerging that serve as a fundamental basis for the ambition of a fully connected future. For example, Big Data techniques have become indispensable for managing the enormous amount of data that IoT networks produce, while Machine Learning serves as a supporting tool that allows IoT networks to become "smart" enough to automatically process their data, draw conclusions from it and even take security measures if they detect a cyber-attack. However, the widespread adoption of IoT brings several challenges [1, 2], such as the latency problem [3], some IoT applications [4, 5] need to meet a maximum delay requirement, or the limited bandwidth usage produced by a large number of devices and data in today's networks. These difficulties are mainly due to the fact that IoT infrastructure is based on services that are centralised in a cloud hosted in large, compute-intensive data centres, but too far away from IoT devices, leading to bandwidth bottlenecks and prohibitive latencies in some cases [6]. To address the limitations of cloud computing in the IoT, a new computing approach is needed to reduce bandwidth consumption and end-user latency [7]. This is how the Edge Computing paradigm emerges, proposing a new topology in which computing tasks are moved as close as possible to the users, so that edge devices collaborate to perform part of the processing of tasks generated by nearby IoT devices, drastically reducing latency and network usage [8]. However, such networks focus on leveraging edge devices, leaving aside the obvious benefits offered by the widely known Cloud Computing. For this reason, a new approach has recently emerged that aims to combine the best of each architecture, so that the devices form part

of a pyramidal network where they all form a computing continuum and can distribute the work between devices according to their needs (latency, computation, space, need for hardware acceleration). This architecture union is called Computing Continuum, which aims to preserve the benefits of Edge Computing but without ignoring the fundamental role of Cloud Computing for very large workloads, also offering computing alternatives between the two approaches (Fog Computing).

All these innovations applied to the IoT concept following the principles of Edge Computing bring new possibilities, allowing the original concept of IoT devices to evolve [9]. Typically, an IoT device is considered to have a "passive" behaviour, as its tasks are limited to collecting data from various sources and sending it to the network [10]. In Edge Computing, it is proposed that IoT devices can also have an "active" behaviour, so that they participate intelligently in the processing of data and in the logic of the applications deployed in the network [11]. However, Edge Computing networks, unlike Cloud Computing networks, consist of a heterogeneous set of devices with very diverse hardware characteristics and geographical positions. Therefore, to make efficient use of edge devices, it is crucial to develop an efficient node orchestrator so that the allocation of resources and tasks take into account the heterogeneous characteristics and roles of edge devices [12].

In terms of resource management, one of the key components of the edge computing architecture is the process of allocating tasks to nodes [13], which decides where computing tasks should be processed in the most optimal way possible [14]. Theoretically, this process is a combinatorial optimisation problem, called a "tasking problem", defined as the process of determining where the computation of each task is performed in order to optimise certain parameters, such as latency [15], execution time, cost, energy consumption [16], geographical position [17], and even whether the node is moving [18] or powered by a battery. The Task Assignment Problem (TAP) can be formulated as a Generalised Assignment Problem (GAP) [19], which has been widely shown to be an NP-hard problem [20, 21].

The GAP can be solved with a multitude of very different methods and techniques [22–24], such as convex optimisation techniques, Lyapunov optimisation [25], Hungarian algorithm [26] and novel genetic algorithms [27]. In addition, promising new methods based on dynamic programming and machine learning techniques have emerged [28], such as reinforcement learning (RL) and neural network reinforcement learning (Deep RL) [29]. Despite that, it is difficult to find optimal solutions to the task allocation problem, especially given the prohibitive computational capacity of the devices commonly used in IoT and Edge Computing, so in practice heuristic-based techniques or methods that search for sub-optimal solutions are often used. One of the most common techniques is Greedy algorithms, which provide sub-optimal solutions but with a low computational cost, which compensates for the sub-optimal result. In some cases, if the problem is properly known, it is possible to achieve solutions very close to the optimal solution. On the other hand, another way of solving optimisation problems are algorithms based on metaheuristics, the most popular being Genetic Algorithms, which are inspired by the process of natural selection.

Finally, a novel technique called reinforcement learning is of special interest both in

the realisation of this thesis and in the most recent contributions in the literature on intelligent systems. This new method is based on machine learning (with hints of behavioural psychology) and is not part of the well-known supervised and unsupervised learning paradigms. The goal of reinforcement learning is to learn an optimal, or near-optimal, policy that maximises the reward function and provides an optimal set of actions for different agent states and environmental conditions. Reinforcement learning algorithms learn iteratively through the immediate rewards they receive each time they perform an action based on their state, thus learning in a trial-error-reward approach. In some cases, it is not possible to make use of a reinforcement learning algorithm directly, such as in scenarios where the agents' state is a large number of variables, which makes Q-Learning algorithms inefficient, or even when the state variables are not discrete.

In general, Reinforcement Learning algorithms face a common challenge: convergence time. These algorithms require multiple iterations to reach an optimal solution, and in some cases, the use of policies with random factors can further extend the convergence process [30, 31]. Nevertheless, in an IoT network environment where nodes are "active" and can collaborate with each other, this limitation can be mitigated. Nodes deployed at the edge can exchange information about the states they know or query more knowledgeable nodes to "guide" the initiation of learning, which accelerates convergence and avoids being stuck in suboptimal solutions.

1.1 Objectives and Methodology

All of the above provides the initial context for the general objective developed in this doctoral thesis: *The analysis and design of solutions for the management and deployment of IoT and Edge nodes capable of intelligently managing tasks within a Computing Continuum ecosystem to optimise resources in the best possible way, dynamically adapting the network to changes in the infrastructure.*

In order to achieve this goal, the following series of more specific objectives have guided the development stages of this thesis:

Objective 1: Analyse and study characteristics and constraints present in devices used in IoT and Edge Computing, to determine the limitations for their use in the Computing Continuum.

Objective 2: Analyse emerging IoT and Edge Computing scenarios related to Smart Cities, Smart Houses, Smart Grids, as well as the technologies involved.

Objective 3: Examine current Edge Computing approaches along with relevant use cases, evaluating the advantages and disadvantages of each of them in order to adapt them to the Computing Continuum.

Objective 4: To study the decision problem concerning the optimal distribution of computing tasks among the different nodes of a network, considering a set of constraints and parameters.

Objective 5: To analyse the state of the art regarding the methods to solve the task-node allocation problem, looking for improvement points for the most popular methods and the possibility of implementing several of them dynamically in an orchestrator.

Objective 6: Integrate the most relevant methods and techniques identified in Computing Continuum systems and test their effectiveness in different scenarios.

Objective 7: Validate the proposals and designs defined in real use cases, where the viability can be checked in a deployment in a real infrastructure.

The realisation of these objectives has followed a distributed approach, performing in parallel small increments in each one of them to later compile the partial results in order to prepare a publication. Having said that, the initial stage of the thesis was mainly focused on carrying out a deep study of the art of Edge-Fog Computing and the task-node assignment problem in order to contextualise all the work that would be done later on. After getting to know the Edge Computing architecture and the most common techniques to solve the TAP, the focus was put on designing test scenarios to check the performance of the most relevant technique we detected (reinforcement learning) and try to modify its design to increase its performance in scenarios where it is not very efficient. As a result of all the tests, several use cases have been defined, with specific network architectures (pyramidal Edge-fog-cloud, disconnected Edge, among others) and algorithms of all kinds have been developed to carry out the process of assigning tasks to the nodes. In addition, simulators, tools and modules have been developed to realistically test the proposals, and can even be integrated into popular orchestrators such as Kubernetes.

To guarantee the practical usefulness of the research conducted in this thesis, we have collaborated in different national and European projects in order to validate the solutions proposed in the framework of each project. Among them, it is remarkable the collaboration with the European project FLUIDOS (Grant. 101070473 [32]) regarding task-node management and Kubernetes, the European project PHOENIX (Grant. 893079 [33]) on the management of devices in intelligent buildings and the national project "Gemelo Digital para la Gestión Sostenible y Adaptación Al Cambio Global de la Agricultura Mediterránea" on the intelligent management of devices following the Digital Twin model.

1.2 Results

As a consequence of the objectives defined above, a set of results have been achieved that have led to various publications in indexed journals and congresses, which are detailed in the section *Publications* after the bibliography. The summary of the main results, as well as the objectives set and these publications are shown in Table 1.1. Furthermore, the following subsections will present a summary of the four main publications of the thesis, three of them comprising the compendium, relating them explicitly to the results achieved.

A detailed summary of the work carried out in each of the publications that make up this doctoral thesis is given below, indicating the results obtained for each one.

Table 1.1: Main results of the thesis

Results	Objectives	Publications
R1. Analysis of the weaknesses of Edge Computing orchestration systems, in particular with regard to the distribution of tasks between nodes.	1, 3, 4	[34] [35] [36] [37]
R2. Design of relevant scenarios for testing both Cloud and Edge Computing orchestration systems.	3, 6	[34] [35] [36] [37]
R3. Propose improvements to the most promising methods for solving the task-node assignment problem, in particular when nodes have limited information.	4, 5	[34] [35]
R4. Implement all the required software for testing, from algorithms to data generators and complete simulators to test the proposed solutions over extended periods of time.	5	[38]
R5. To explore the application of Edge Computing techniques in scenarios that are not specifically oriented to the management of computing services, but which show a special interest in the optimisation of resources.	2, 3	[36] [37] [38]
R6. Test machine learning-based methods to see if they are able to adapt in real time to unexpected changes in the computing system.	7	[35] [36] [37]
R7. Identify the most popular software tools for task management in Cloud Computing and study the necessary modifications to adapt it to the Computing Continuum.	6, 7	[39] [40] [41]
R8. Design modules that easily adapt Cloud Computing software solutions to be applicable in Edge and Continuum scenarios.	6, 7	[39] [40] [41]

1.2.1 A multi-layer guided reinforcement learning-based tasks offloading in edge computing

In this first publication [35] the basic idea of the problem of assigning tasks to nodes is presented and a study is made of the state of the art for solving it with traditional methods. After that, the Reinforcement Learning technique is identified as very promising but with certain limitations in Edge Computing scenarios, specifically the difficulty of converging to acceptable solutions in the initial stages of the algorithm (**R1**). To demonstrate the limitations, the allocation problem is mathematically defined exactly for a layer-based architecture (Edge-Fog-Cloud), where each layer hosts nodes of different performance (**R2**). This would then define the problem as the most optimal way to allocate the set of tasks in such a network of nodes. Using the RL technique, it is possible to iteratively make the edge nodes that decide where to send a task gradually discover the best destinations (neighbour nodes, fog nodes or the cloud). The problem lies in the fact that initially the algorithm erroneously sends tasks to unsuitable nodes because it has not yet converged towards the optimal policy, and even if the network situation changes abruptly, it again

has difficulties in reaching the optimal solution.

The initial reduced or limited performance in highly dynamic networks is what motivates the publication's contribution, modifying the RL agent so that it can "ask for help" in situations where its knowledge is limited (**R3**). The idea is that when the reinforcement learning agent of an edge node does not have enough knowledge or is in a situation where it is not clear what the best solution is (sudden changes in the network) it can delegate the decision to a higher level node (e.g. in the Fog layer) which has more knowledge and even a state that considers more parameters of the infrastructure. After that, the node learns from the decision of the higher level agent as if it was "guided" to the best solutions by more knowledgeable nodes.

To ensure the performance of the proposal, simulations are carried out with a modified version of PureEdgeSim to contemplate the developed algorithms. The results of the tests clearly show how the RL-based methods outperform the most basic approaches and also how the contribution of the multilayer agent improves the convergence speed of the method and even outperforms the basic RL solution in more complex scenarios by avoiding suboptimal solutions (**R6**).

1.2.2 Sim-PowerCS: An extensible and simplified open-source energy simulator

The second publication [38] derives from a transversal problem detected in the experiments of the first publication, which is the lack of simulators and test environments (**R4**) that easily adapt to Continuum Computing and reinforcement learning scenarios. In addition, for the next research objective it was necessary to consider as a key factor the energy consumed by the devices (**R5**), as well as the possibility of configuring operating schedules and being able to turn them off on time if necessary. These requirements arise from the desire to integrate the idea of Edge Computing within the Smart Building concept, being able to model the devices as services whose consumption is to be optimised. To this end, a simulator is developed capable of emulating the operation of a smart home with different types of energy sources, resulting in a set of data to identify the performance of each type of control algorithm implemented.

An adaptive energy orchestrator for cyberphysical systems using multiagent reinforcement learning

This third publication [37], which was presented in an initial version at a conference [36], deals with the application of multi-agent reinforcement learning as a control system for different types of devices and services deployed in a smart home, with the ultimate goal of reducing energy consumption while maximising the availability of services (**R1**). Although this publication is not part of the compendium of this thesis, it is considered relevant because it is part of the most relevant results obtained.

The concept of Smart City and Smart Home is quite new and promising. Among the benefits of these new paradigms is the reduction of the energy they require and, therefore, the emissions they generate, which is a high priority nowadays. However, a key

challenge in these systems is the proper energy management of devices, especially those whose operating schedules can be programmed (such as washing machines, ventilation and lighting systems) and shifted to more favourable times, such as when energy is cheaper at night. This challenge is essentially presented as an optimisation problem, similar to the one addressed in the first publication. Applying reinforcement learning to provide intelligence to each device is therefore very promising, allowing each device to make decisions autonomously.

For this purpose, the simulator developed in the previous publication is used and a use case is presented in which efficient energy management and proper planning are crucial (**R2**, **R5**). The scenario is an off-grid home that relies exclusively on energy generated by solar panels and stored in batteries. Services must efficiently manage the use of batteries and make the most of excess solar energy in the mornings. Inappropriate energy use could deplete the batteries, activating an emergency generator that would supply minimal power until the batteries are recharged. Therefore, the main objective of the control system is to avoid using the generator altogether (i.e. to avoid draining the batteries), followed by preventing low battery levels and maximising energy consumption when the batteries are full and solar generation is high. In addition, the system must adapt quickly to unforeseen situations, such as sudden consumption peaks, a temporary decrease in solar generation due to clouds, periodic energy events, among others.

The technique used to manage the system is based on modelling the devices as cyber-physical systems (CPS), where the physical components are abstracted into services and managed in a similar way to digital services. Subsequently, these CPSs are individually managed by reinforcement learning agents, which together seek to optimise energy use within a multi-agent model.

To test the effectiveness of the proposal, we use the simulator that evaluates the performance of various alternatives for the control system (**R6**). The results show that the reinforcement learning algorithm is not only superior in efficiency, but also in its ability to adapt quickly to changes. Furthermore, it highlights its ability to learn from past events and anticipate future situations, proactively adjusting the service uptime.

1.2.3 Adapting Containerized Workloads for the Continuum Computing

The fourth and last paper of the compendium [40] compiles what has been learnt during the PhD to apply it to a real Cloud Computing scenario using typical industry technology (Kubernetes) and adapt it to the Continuum Computing architecture (**R7**).

For this purpose, the multi-layer architecture (Edge-Fog-Cloud) is used again, but enhanced to make it possible to interconnect microservices between the layers, which are implemented as Kubernetes clusters connected by Liqo.io. Since the design of the K8s orchestrator is used as the basis for the assignment of tasks (Pods) to nodes, it is necessary to improve the mathematical model of the assignment problem (**R8**). This model should include not only the previously defined constraints, such as CPU, RAM and disk usage, but also other constraints such as anti-affinity, latency and specific capabilities. In this way, the inherent limitations of Kubernetes to adapt to the Continuum concept

become evident, as it is not possible to consider all these specifications together with the heterogeneous nature of the devices during the process of assigning pods to nodes.

To highlight the value of the proposal, a use case is designed with an application composed of several microservices, which must be deployed in the network nodes according to their specific requirements (such as country, maximum latency, anti-affinity and hardware needs). It is then compared with different orchestration alternatives. The main objective is to demonstrate the need to adapt the orchestration systems of the most popular container managers to be able to host applications that take advantage of the Continuum concept. This is demonstrated in the experiments conducted, where it is evaluated how, in high-demand situations (with numerous services deployed), traditional K8s methods tend to saturate the infrastructure and do not efficiently exploit the capabilities of edge devices to perform simple, low-latency tasks. In contrast, the proposed algorithms (greedy, genetic and optimal with backtracking) achieve a better utilisation of the nodes, avoiding infrastructure saturation.

On the other hand, in the publication [41] the use case was extended to a more complex intents-based orchestrator, capable of handling both service deployments and network security policies. This system also considers additional particularities, such as the fact that each service or policy can be resolved in different ways. Therefore, the solution must not only determine the best node for deployment, but also identify the most suitable implementation for the required deployment. In addition, to increase the flexibility of the solution, the method used to solve the problem is allowed to vary dynamically depending on its size and complexity. In this way, the results present several options of methods based on optimal and genetic algorithms, highlighting how each of them offers varying accuracy and different execution times.

1.3 Conclusions and future work

The increase in internet-connected devices, both those following the classic IoT model and those based on new Edge Computing techniques, and their growing adoption in our daily lives, raise significant challenges in several areas of IT, from security to efficient data management. This not only requires developing new security strategies to protect privacy and sensitive information, but also optimising the use of network resources and improving interoperability between different platforms and devices. The main objective of this thesis is to analyse the fundamentals of the workload distribution process between nodes, identify opportunities for improvement, and develop the necessary software for its implementation in real systems, adapting them to the concept of Computing Continuum.

As a result, the main conclusions drawn can be summarised in the following list:

- Although the concept of the Internet of Things is widely studied, there are still certain limitations when applying distributed computing approaches among a set of nodes.
- Currently IoT networks are not ready to adapt to the idea of Edge Computing, making it notoriously difficult to deploy systems based on Computing Continuum.

- There are parallel use cases and scenarios that can make use of Edge and Continuum technology with great potential, such as smart cities and smart homes.
- The problem of assigning tasks to nodes is well known but not adequately addressed in IoT, as well as not taking into account very important Continuum requirements such as the different hardware characteristics of each node.
- The potential of reinforcement learning-based techniques for the task-node assignment problem is recognised, as they are able to adapt in real time to abrupt changes and anticipate periodic events.
- However, the limitations of the traditional RL approach are demonstrated in an edge computing scenario and an improvement of the technique to exploit the layered architecture is proposed.
- Smart cities and smart homes can also benefit from the Continuum concept and apply reinforcement learning techniques to manage their services. This is demonstrated by a series of experiments in a smart home.
- Currently, machine learning techniques and architectures such as Edge Computing are not applied in the most popular solutions in the industry, such as Kubernetes, so it is necessary to develop a whole set of tools to be able to implement our proposals in these systems.
- All the solutions developed have been tested in different use cases to demonstrate their effectiveness, in addition to having used part of the results to solve similar problems in other projects.

The results and conclusions obtained in this thesis have contributed to consolidate the fundamental knowledge on the architecture of the Computing Continuum, as well as on related topics, among which areas of special interest such as the problem of assigning tasks to nodes, Smart Cities and energy optimisation stand out. However, throughout the development of this thesis, many issues have emerged that have remained outside its initial scope. Moreover, as the various aspects of the Computing Continuum were explored in more depth, new lines of research and challenges emerged that could be explored in the future.

The first of these would be the use of much more complex reinforcement learning techniques, such as those based on neural networks (DeepRL). This improved version of RL solves the limitation that the state space has to be discrete, but adds a memory and computational overhead that is difficult for the smaller Continuum devices to bear. Therefore, an interesting point for future work would be to study DeepRL techniques with very small networks and implement them in the smallest possible way to integrate them in devices with low computational capacity.

On the other hand, the orchestrator developed and implemented in Kubernetes shows great potential, not only to manage the Continuum and solve the task-node assignment problem, but also to address other issues, such as decision making on security issues, the selection of the software that will implement the required service, among many others.

In this context, it is considered that an interesting direction for future work would be to improve the capabilities of the orchestrator, both in terms of its functionalities and semantics, so that it can understand a wider variety of constraints and requirements in its deployments. In addition, it would be valuable to incorporate a wide range of algorithms that allow decision problems to be solved dynamically, adapting to more complex and variable environments. Precisely these aspects are already being explored as a basis for future work following the thesis. Among the important aspects is the extension of the analysis of the optimisation problem, incorporating conditions related to the security and privacy of each particular node, which would allow nodes to be excluded if they are untrustworthy according to these parameters. In addition, although some algorithms stand out from others, it has proven useful to have a variety of methods to solve the allocation problem with greater or lesser accuracy, in trade-offs of memory or time. Therefore, another future research direction that is being addressed is meta-orchestration, which will allow the appropriate algorithm to be dynamically selected for each request according to the size of the problem, time requirements, available memory and other relevant parameters.

Finally, in the problem of assigning tasks to nodes, research has mainly focused on methods that seek to optimise a single objective, even though this may integrate several factors through a weighted sum or a joint state. However, the use of purely multi-objective methods, which would allow multiple independent criteria to be addressed simultaneously, has not been explored in depth. This opens up a promising line of research, as multi-objective methods could offer more balanced and flexible solutions in complex scenarios.

Capítulo 2

Resumen

En la era de la información en la que vivimos, los datos son un recurso fundamental, y su recopilación y procesamiento eficiente resultan esenciales. Gracias a los avances en las tecnologías de hardware y al impulso por digitalizar nuestra sociedad, conectar todo a nuestro alrededor a Internet se presenta como un avance prometedor. De esta forma surge la prometedora tecnología del Internet de las Cosas (*Internet of Things*, IoT), la cual pretende conectar dispositivos comunes de nuestro día a día (*Things*) a Internet y recoger todo tipo de datos que produzcan. Estos dispositivos conectados conforman lo que se denomina una red IoT, en la cual cada uno de ellos dispone de conexión a internet y en algunos casos son capaces de comunicarse entre sí, esto permite nuevos paradigmas que facilitan escenarios como el despliegue de decenas de sensores y dispositivos en grandes instalaciones (hogares, fábricas, ciudades, etc..) para poder ser consultados en tiempo real desde cualquier parte del mundo.

Junto al Internet de las Cosas, han estado surgiendo otras tecnologías y técnicas especialmente útiles que sirven de base fundamental para la aspiración de un futuro totalmente conectado. Por ejemplo, las técnicas de Bigdata se han visto indispensables para la gestión de la enorme cantidad de datos que llegan a producir las redes IoT, paralelamente, el Machine Learning sirve como herramienta de apoyo que permite a las redes IoT dotarse de «inteligencia» para poder procesar automáticamente sus datos, sacar conclusiones de ellos e incluso tomar medidas de seguridad si detectan un ciberataque. Sin embargo, la adopción generalizada del IoT conlleva varios retos [1, 2] como el problema de la latencia [3], algunas aplicaciones [4, 5] IoT necesitan cumplir un requisito de retardo máximo, o el uso limitado del ancho de banda producido por un gran número de dispositivos y datos en las redes actuales. Estas dificultades se deben principalmente a que la infraestructura IoT se basa en servicios que están centralizados en una nube alojada en grandes centros de datos con mucha capacidad de cálculo, pero demasiado alejados de los dispositivos IoT, lo que provoca cuellos de botella en el ancho de banda y latencias prohibitivas en algunos casos [6]. Para superar las limitaciones de la computación en la nube en el IoT, es necesario un nuevo enfoque informático que reduzca el consumo de ancho de banda y la latencia del usuario final [7]. De esta manera surge el paradigma Edge Computing que propone una nueva topología en la que las tareas de computación se trasladan lo más cerca posible de los usuarios, de modo que los dispositivos de borde colaboran para realizar parte del

procesamiento de las tareas generadas por los dispositivos IoT cercanos, reduciendo drásticamente la latencia y el uso de la red [8]. No obstante, este tipo de redes se enfocan en aprovechar los dispositivos de borde, dejando de lado los evidentes beneficios que ofrece el ampliamente conocido Cloud Computing. Por esta razón surge recientemente un enfoque nuevo que pretende unir lo bueno de cada arquitectura, de manera que los dispositivos formen parte de una red piramidal donde todos formen un continuo de computo pudiendo distribuir el trabajo entre los dispositivos en función de las necesidades (latencia, computo, espacio, necesidad de hardware de aceleración). Esta unión de arquitectura es lo que se denomina Computing Continuum, el cual pretende respetar las bondades del Edge Computing pero sin relegar el papel fundamental que tiene el Cloud Computing para cargas de trabajo muy grandes, ofreciendo además alternativas de computo entre ambos enfoques (Fog Computing).

Todas estas novedades aplicadas al concepto de IoT siguiendo los principios de Edge Computing aporta nuevas posibilidades, permitiendo que el concepto original de dispositivos IoT evolucione [9]. Normalmente, se considera que un dispositivo IoT tiene un comportamiento "pasivo", ya que sus tareas se resumen en recoger datos de diversas fuentes y enviarlos a la red [10]. En Edge Computing, se propone que los dispositivos IoT puedan tener también un comportamiento "activo", de forma que participen de manera inteligente en el procesamiento de los datos y en la lógica de las aplicaciones desplegadas en la red [11]. Sin embargo, las redes de Edge Computing, a diferencia de las de Cloud Computing, están formadas por un conjunto heterogéneo de dispositivos con características de hardware y posiciones geográficas muy diversas. Por lo tanto, para hacer un uso eficiente de los dispositivos de borde, es crucial desarrollar un orquestador de los nodos eficiente para que la asignación de recursos y tareas tengan en cuenta las características heterogéneas y los roles de los dispositivos de borde [12].

En cuanto a la gestión de recursos, uno de los componentes clave de la arquitectura Edge Computing es el proceso de asignación de tareas a nodos [13], el cual decide dónde deben procesarse las tareas de computación de la forma más óptima posible [14]. Teóricamente, este proceso es un problema de optimización combinatoria, denominado "problema de asignación de tareas", definido como el proceso de determinar dónde se realiza el cómputo de cada tarea para optimizar ciertos parámetros, como la latencia [15], el tiempo de ejecución, el coste, el consumo de energía [16], la posición geográfica [17], e incluso si el nodo se está moviendo[18] o se alimenta con una batería. El problema de asignación de tareas (*Task Assignment Problem*, TAP) puede formularse como un Problema de Asignación Generalizado (*Generalized Assignment Problem*, GAP) [19], que se ha demostrado ampliamente que es un problema NP-hard [20, 21].

El GAP puede resolverse con multitud de métodos y técnicas muy diferentes [22–24], como técnicas de optimización convexa, optimización de Lyapunov [25], algoritmo Húngaro [26] y algoritmos genéticos novedosos [27]. Además, han surgido nuevos métodos muy prometedores basados en técnicas de programación dinámica y machine learning [28], como el aprendizaje por refuerzo (RL) y el aprendizaje por refuerzo de redes neuronales (Deep RL) [29]. A pesar de ello, es difícil encontrar soluciones óptimas al problema de asignación de tareas, especialmente dada la prohibitiva capacidad computacional de los dispositivos que habitualmente se usan en IoT y Edge Computing, por lo que en la

práctica se suelen utilizar técnicas basadas en heurísticas o métodos que buscan soluciones subóptimas. Una de las técnicas más comunes son los algoritmos Greedy, que proporcionan soluciones subóptimas pero con un bajo coste computacional, lo que compensa el resultado no óptimo. En algunos casos, si se conoce adecuadamente el problema es posible lograr soluciones muy próximas a la solución óptima. Por otro lado, otra manera de resolver problemas de optimización son los algoritmos basados en metaheurísticas, los más populares son los Algoritmos Genéticos que se inspiran en el proceso de selección natural.

Finalmente, una novedosa técnica denominada aprendizaje por refuerzo es de especial interés tanto en la realización de esta tesis como en las aportaciones más recientes en la literatura sobre sistemas inteligentes. Este nuevo método se basada en el aprendizaje automático (con matices de psicología conductista) y no forma parte de los conocidos paradigmas de aprendizaje supervisado y no supervisado. El objetivo del aprendizaje por refuerzo es aprender una política óptima, o casi óptima, que maximice la función de recompensa y proporcione un conjunto óptimo de acciones para diferentes estados del agente y condiciones ambientales. Los algoritmos de aprendizaje por refuerzo aprenden iterativamente a través de las recompensas inmediatas que reciben cada vez que realizan una acción en función de su estado, de esa manera aprenden siguiendo un enfoque de prueba-error-recompensa. En algunos casos, no es posible hacer uso de un algoritmo de aprendizaje por refuerzo directamente, como en escenarios en los que el estado de los agentes es un gran número de variables, lo que hace ineficientes los algoritmos de Q-Learning, o incluso cuando las variables de estado no son discretas.

En general, los algoritmos de Aprendizaje por Refuerzo enfrentan un desafío común: el tiempo de convergencia. Estos algoritmos requieren múltiples iteraciones para llegar a una solución óptima, y en algunos casos, el uso de políticas con factores aleatorios puede prolongar aún más ese proceso [30, 31]. No obstante, en un entorno de red IoT donde los nodos sean "activos" y puedan colaborar entre sí, esta limitación puede mitigarse. Los nodos desplegados en el Edge pueden intercambiar información sobre los estados que conocen o consultar a nodos con mayor conocimiento para "guiar" el inicio del aprendizaje, lo que acelera la convergencia y evita quedar atrapado en soluciones subóptimas.

2.1 Objetivos y Metodologías

Todo lo planteado anteriormente sirve de contexto inicial para el objetivo general que se desarrolla en esta tesis doctoral: *El análisis y el diseño de soluciones para la gestión y el despliegue de nodos IoT y Edge capaces de administrar tareas de manera inteligente dentro de un ecosistema de Computing Continuum para optimizar los recursos de la mejor manera posible, adaptando dinámicamente la red a cambios en la infraestructura.*

Para lograr esta meta, se han planteado la siguiente serie de objetivos más específicos que han guiado las etapas de desarrollo de la presente tesis:

Objetivo 1: Analizar y estudiar características y restricciones presentes en los dispositivos utilizados en IoT y Edge Computing, para determinar las limitaciones para su uso en el Computing Continuum.

Objetivo 2: Analizar escenarios de IoT y Edge Computing emergentes relacionados con Smart Cities, Smart Houses, Smart Grids, asi como de las tecnologías involucradas.

Objetivo 3: Examinar los enfoques actuales de Edge Computing junto con casos de uso destacados, valorando las ventajas y desventajas de cada uno de ellos para adaptarlos al Computing Continuum.

Objetivo 4: Estudiar el problema de decisión referente a la distribución optima de las tareas de computo entre los diferentes nodos de una red, considerando toda una serie de restricciones y parámetros.

Objetivo 5: Analizar el estado del arte referente a los métodos para resolver el problema de asignación tarea-nodo, buscando puntos de mejora para los métodos más populares y la posibilidad de implementar varios de manera dinámica en un orquestador.

Objetivo 6: Integrar los métodos y técnicas mas relevantes identificados en sistemas de Computing Continuum y probar la eficacia en diferentes escenarios.

Objetivo 7: Validar las propuestas y los diseños definidos en casos de uso reales, donde se pueda comprobar la viabilidad en un despliegue en una infraestructura real.

La realización de estos objetivos ha seguido un enfoque distribuido, realizando de manera paralela pequeños incrementos en cada uno de ellos para posteriormente recopilar los resultados parciales que se analizaban y preparaban para trabajarlo en una publicación. Dicho lo cual, la etapa inicial de la tesis se centró principalmente en realizar un profundo estudio del arte de los temas mas relevantes del Edge-Fog Computing y el problema de asignación tarea-nodo para contextualizar todo el trabajo que se haría posteriormente. Tras conocer adecuadamente la arquitectura del Edge Computing y las técnicas mas comunes para resolver el TAP se puso el foco en diseñar escenarios de pruebas donde comprobar el funcionamiento de la técnica mas relevante que detectamos (aprendizaje por refuerzo) e intentar modificar su diseño para aumentar su rendimiento en escenarios donde no es muy eficiente. Como resultado de todas las pruebas se han definido diversos casos de uso, con arquitecturas de redes concretas (piramidal Edge-fog-cloud, Edge desconectado, entre otras) y se han desarrollado algoritmos de todo tipo para realizar el proceso de asignación de las tareas a los nodos. Junto a eso, se han realizado simuladores, herramientas y módulos para poder probar de forma realista las propuestas, pudiendo integrarla incluso en orquestadores populares como Kubernetes.

Para asegurar la utilidad práctica del trabajo realizado en esta tesis, se ha colaborado en diferentes proyectos tanto nacionales como europeos para validar las soluciones propuestas en el marco de cada proyecto. Entre ellos cabe destacar la colaboración con el proyecto europeo FLUIDOS (Grant. 101070473 [32]) en lo referente a gestión de tareas-nodos y Kubernetes, el proyecto europeo PHOENIX (Grant. 893079 [33]) sobre la gestión de dispositivos en edificios inteligentes y el proyecto nacional “Gemelo Digital para la

Gestión Sostenible y Adaptación Al Cambio Global de la Agricultura Mediterránea" sobre la administración inteligente de dispositivos siguiendo el modelo de Gemelo Digital.

2.2 Resultados

Como consecuencia de los objetivos definidos anteriormente se han logrado una serie de resultados que han permitido la realización de diversas publicaciones en revistas indexadas y congresos, detalladas en el apartado *Publicaciones* tras la bibliografía. El resumen de los principales resultados, así como los objetivos planteados y dichas publicaciones se muestran en la Tabla 2.1. Además, en las siguientes subsecciones se presentará de manera resumida las cuatro publicaciones principales de la tesis, conformando tres de ella el compendio, relacionándolas de manera explicita con los resultados alcanzados.

Table 2.1: Principales resultados de la tesis

Resultado	**Objetivos**	**Publicaciones**
R1. Análisis de las deficiencias de los sistemas de orquestación de Edge Computing, en especial lo referente a la distribución de tareas entre los nodos.	1, 3, 4	[34] [35] [36] [37]
R2. Diseño de escenarios relevantes para realizar pruebas de sistemas de orquestación tanto de Cloud como de Edge Computing.	3, 6	[34] [35] [36] [37]
R3. Proponer mejoras a los métodos mas prometedores para resolver el problema de asignación tarea-nodo, en concreto cuando los nodos tienen poca información.	4, 5	[34] [35]
R4. Implementar todo el software necesario para realizar las pruebas, desde algoritmos hasta generadores de datos y simuladores completos para poner a prueba durante periodos extensos las soluciones propuestas.	5	[38]
R5. Explorar la aplicación de técnicas de Edge Computing en escenarios que no están específicamente orientados a la gestión de servicios de computación, pero que presentan un especial interés en la optimización de recursos.	2, 3	[36] [37] [38]
R6. Poner a pruebas los métodos basados en aprendizaje automático para ver si son capaces de adaptarse en tiempo real a los cambios inesperados en la plataforma de computación.	7	[35] [36] [37]
R7. Identificar las herramientas software mas populares para la administración de tareas en Cloud Computing y estudiar las modificaciones necesarias para adaptarlo al Computing Continuum.	6, 7	[39] [40] [41]
R8. Diseñar los módulos que adapten fácilmente soluciones software de Cloud Computing para que sean aplicables en escenarios de Edge y Continuum.	6, 7	[39] [40] [41]

A continuación, se encuentra un resumen detallado del trabajo realizado en cada una de las publicaciones que componen esta tesis doctoral, indicando para cada una los resultados obtenidos.

2.2.1 A multi-layer guided reinforcement learning-based tasks offloading in edge computing

En esta primera publicación [35] se presenta la idea básica del problema de asignación de tareas a nodos y se realiza un estudio del estado del arte para resolverlo con métodos tradicionales. Tras eso se identifica la técnica de Aprendizaje por Refuerzo como muy prometedora pero con ciertas limitaciones en escenarios de Edge Computing, en concreto la dificultad de converger a soluciones aceptables en las etapas iniciales del algoritmo (**R1**). Para demostrare las limitaciones se define matemáticamente con exactitud el problema de asignación para una arquitectura basada en capas (Edge-Fog-Cloud), donde cada una de ellas alberga nodos de diferentes prestaciones (**R2**). De esta manera se definiría entonces el problema como la forma más óptima de asignar el conjunto de tareas en esa red de nodos. Haciendo uso de la técnica de RL se puede lograr que, de manera iterativa, los nodos del Edge que deciden donde enviar una tarea descubran poco a poco cuales son los mejores destinos (nodos vecinos, nodos fog o la cloud). El problema radica en que inicialmente el algoritmo envía tareas erróneamente a nodos no adecuados ya que aún no ha convergido hacia la política óptima, e incluso si la situación de la red cambia bruscamente vuelve a tener dificultades para llegar a la solución óptima.

El rendimiento reducido inicial o limitado en redes muy dinámicas es lo que motiva la aportación de la publicación, modificar el agente RL para que pueda "pedir ayuda" en las situaciones donde su conocimiento es limitado (**R3**). La idea es que cuando el agente de aprendizaje por refuerzo de un nodo Edge no tenga suficiente conocimiento o se encuentre en una situación donde no tiene claro cual es la mejor solución (cambios bruscos en la red) pueda delegar la decisión a un nodo (por ejemplo, en la capa Fog) superior el cual tiene mas conocimiento e incluso un estado que tiene en cuenta más parámetros de la infraestructura. Tras eso, el nodo aprende de la decisión del agente de nivel superior como si fuera "guiado" hacia las mejores soluciones por nodos con más conocimiento.

Para asegurar el funcionamiento de la propuesta se realizan simulaciones con una versión modificada de PureEdgeSim para contemplar los algoritmos desarrollados. En los resultados de las pruebas se verifica claramente como los métodos basados en RL superan los enfoques más básicos y además como la aportación del agente multicapa mejora la velocidad de convergencia del método e incluso supera en escenarios mas complejos el rendimiento de la solución de RL básica al evitar soluciones subóptimas (**R6**).

2.2.2 Sim-PowerCS: An extensible and simplified open-source energy simulator

La segunda publicación [38] surge de un problema trasversal detectado en la realización de los experimentos de la primera publicación, este es la falta de simuladores y entornos de pruebas (**R4**) que se adapten fácilmente a escenarios de Computing Continuum y

aprendizaje por refuerzo. Además, para el siguiente objetivo de investigación era necesario considerar como un factor clave la energía consumida por los dispositivos (**R5**), así como la posibilidad de configurar horarios de funcionamiento y poder apagarlos puntualmente si fuera necesario. Estos requisitos surgen al querer integrar la idea de Edge Computing dentro del concepto de Smart Building, pudiendo modelar los dispositivos como servicios de los que se quiere optimizar su consumo. Para ello se desarrolla un simulador capaz de emular el funcionamiento de una casa inteligente con diferentes tipos de fuentes de energía, dando como resultado un conjunto de datos para identificar el rendimiento de cada tipo de algoritmo de control implementado.

An adaptive energy orchestrator for cyberphysical systems using multiagent reinforcement learning

Esta tercera publicación [37], de la que se presentó una versión inicial en un congreso [36], trata sobre la aplicación de aprendizaje por refuerzo multiagente como sistema de control de diferentes tipos de dispositivo y servicios desplegados en una casa inteligente, con el objetivo final de reducir el consumo energético a la vez que se máxima la disponibilidad de los servicios (**R1**). Aunque esta publicación finalmente no forma parte del compendio de esta tesis, se considera relevante abordarla por ser parte de los resultados más destacados obtenidos y una continuación de la publicación anterior.

El concepto de Ciudad Inteligente y Hogar Inteligente es bastante novedoso y prometedor. Entre las virtudes de estos nuevos paradigmas está la reducción de energía que requieren y, por tanto, las emisiones que generan, una prioridad bastante destacable en la actualidad. Sin embargo, un desafío clave en estos sistemas es la adecuada gestión energética de los dispositivos, especialmente aquellos cuyos horarios de funcionamiento pueden programarse (como lavadoras, sistemas de ventilación e iluminación) y desplazarse a momentos más favorables, como cuando la energía es más barata por la noche. Este desafío se presenta esencialmente como un problema de optimización, similar al que se abordó en la primera publicación. Por ello, aplicar aprendizaje por refuerzo para dotar de inteligencia a cada dispositivo resulta muy prometedor, permitiendo que cada uno tome decisiones de manera autónoma.

Para ello, se utiliza el simulador desarrollado en la publicación anterior y se plantea un caso de uso en el que la gestión eficiente de la energía y una correcta planificación resultan cruciales (**R2**, **R5**). El escenario es una vivienda sin conexión a la red eléctrica, que depende exclusivamente de la energía generada por paneles solares y almacenada en baterías. Los servicios deben gestionar de manera eficiente el uso de las baterías y aprovechar al máximo el exceso de energía solar durante las mañanas. Un uso inadecuado de la energía podría agotar las baterías, activando un generador de emergencia que suministraría energía mínima hasta que las baterías se recarguen. Por lo tanto, el objetivo principal del sistema de control es evitar por completo el uso del generador (es decir, evitar agotar las baterías), seguido de prevenir niveles de batería bajos y maximizar el consumo energético cuando las baterías estén llenas y la generación solar sea alta. Además, el sistema debe adaptarse rápidamente a situaciones imprevistas, como picos de consumo repentinos, una disminución temporal en la generación solar debido a

nubes, eventos energéticos periódicos, entre otros.

La técnica utilizada para gestionar el sistema se basa en modelar los dispositivos como sistemas ciberfísicos (CPS), donde los componentes físicos se abstraen en servicios y se gestionan de forma similar a los servicios digitales. Posteriormente, estos CPS son administrados individualmente mediante agentes de aprendizaje por refuerzo, que en conjunto buscan optimizar el uso energético dentro de un modelo multiagente.

Para comprobar la eficacia de la propuesta, se utiliza el simulador que evalúa el rendimiento de diversas alternativas para el sistema de control (**R6**). Los resultados demuestran que el algoritmo de aprendizaje por refuerzo no solo es superior en eficiencia, sino también en su capacidad para adaptarse rápidamente a los cambios. Además, destaca su habilidad para aprender de eventos pasados y anticipar situaciones futuras, ajustando de manera proactiva el tiempo de funcionamiento del servicio.

2.2.3 Adapting Containerized Workloads for the Continuum Computing

El cuarto y último trabajo del compendio [40] recoge lo aprendido durante el doctorado para aplicarlo a un escenario real de Cloud Computing que usa la tecnología típica de la industria (Kubernetes) y adaptarlo a la arquitectura de Continuum Computing (**R7**).

Para ello se vuelve a usar la arquitectura de varias capas (Edge-Fog-Cloud) pero mejorada para que sea posible interconectar microservicios entre las capas, las cuales se implementan como clústeres Kubernetes conectados por Liqo.io. Dado que el diseño del orquestador de K8s se utiliza como base para la asignación de tareas (Pods) a los nodos, es necesario mejorar el modelo matemático del problema de asignación (**R8**). Este modelo debe incluir no solo las restricciones previamente definidas, como el uso de CPU, RAM y disco, sino también otras como antiafinidad, latencia y capacidades específicas (capabilities). De este modo, se evidencian las limitaciones inherentes de Kubernetes para adaptarse al concepto del Continuum, ya que no es posible considerar todas estas especificaciones junto con la naturaleza heterogénea de los dispositivos durante el proceso de asignación de pods a nodos.

Para destacar el valor de la propuesta, se diseña un caso de uso con una aplicación compuesta por varios microservicios, los cuales deben ser desplegados en los nodos de la red según sus requisitos específicos (como país, latencia máxima, anti-afinidades y necesidades de hardware). Luego, se compara con distintas alternativas de orquestación. El objetivo principal es demostrar la necesidad de adaptar los sistemas de orquestación de los gestores de contenedores más populares para que puedan alojar aplicaciones que aprovechen el concepto del Continuum. Esto se demuestra en los experimentos realizados, donde se evalúa cómo, en situaciones de alta demanda (con numerosos servicios desplegados), los métodos tradicionales de K8s tienden a saturar la infraestructura y no aprovechan eficientemente las capacidades de los dispositivos de borde para realizar tareas simples y de baja latencia. En contraste, los algoritmos propuestos (greedy, genéticos y óptimo con backtracking) logran un mejor aprovechamiento de los nodos, evitando la saturación de la infraestructura.

Por otro lado, en la publicación [41] se amplió el caso de uso a un orquestador más

complejo basado en Intents, capaz de gestionar tanto los despliegues de servicios como las políticas de seguridad de red. Este sistema también considera particularidades adicionales, como el hecho de que cada servicio o política puede resolverse de distintas maneras. Por ello, la solución no solo debe determinar el mejor nodo para el despliegue, sino también identificar la implementación más adecuada para el despliegue requerido. Además, para incrementar la flexibilidad de la solución, se permite que el método empleado para resolver el problema varíe dinámicamente en función de su tamaño y complejidad. De este modo, los resultados presentan diversas opciones de métodos basados en algoritmos óptimos y genéticos, destacando cómo cada uno de ellos ofrece una precisión variable y tiempos de ejecución diferentes.

2.3 Conclusiones y trabajo futuro

El incremento de dispositivos conectados a internet, tanto los que siguen el modelo clásico de IoT como aquellos basados en las nuevas técnicas de Edge Computing, y su creciente adopción en nuestra vida cotidiana, plantean desafíos significativos en diversas áreas de la informática, que abarcan desde la seguridad hasta la gestión eficiente de los datos. Esto no solo requiere desarrollar nuevas estrategias de seguridad para proteger la privacidad y la información sensible, sino también optimizar el uso de los recursos de red y mejorar la interoperabilidad entre diferentes plataformas y dispositivos. El objetivo principal de esta tesis es analizar los fundamentos del proceso de distribución de cargas de trabajo entre nodos, identificar oportunidades de mejora, y desarrollar el software necesario para su implementación en sistemas reales, adaptándolos al concepto de Computing Continuum.

En consecuencia, se pueden resumir las principales conclusiones obtenidas en la siguiente lista:

- Aunque el concepto de internet de las cosas esté ampliamente estudiado, existen aun ciertas limitaciones cuando se aplican enfoques de computación distribuida entre un conjunto de nodos.
- En la actualidad las redes IoT no están preparadas para adaptarse a la idea de Edge Computing, dificultando notoriamente el despliegue de sistemas basados en Computing Continuum.
- Existen paralelamente casos de uso y escenarios que pueden hacer uso de la tecnología de Edge y Continuum con un gran potencial, como pueden ser ciudades y hogares inteligentes.
- El problema de asignación de tareas a nodos es bien conocido pero no se trata de manera adecuada en IoT y Edge Computing, además de no contemplar requisitos muy importantes del Continuum como pueden ser las diferentes características hardware de cada nodo.
- Se reconoce el potencial para el problema de asignación tarea-nodo de las técnicas basadas en aprendizaje por refuerzo, dado que son capaces de adaptase en tiempo real a cambios bruscos y anticiparse a eventos periódicos.

- Aun así, se demuestra en un escenario de Edge Computing las limitaciones del enfoque tradicional de RL y se propone una mejora de la técnica para que explote la arquitectura basada en capas.

- Las ciudades y casas inteligentes también pueden beneficiarse del concepto de Continuum y aplicar técnicas de aprendizaje por refuerzo para gestionar sus servicios. Esto queda demostrado con una serie de experimentos realizados en la simulación de una casa inteligente.

- Actualmente no se aplican técnicas de Machine Learning ni arquitecturas como Edge Computing en las soluciones mas populares de la industria, como Kubernetes, por lo que es necesario desarrollar todo un conjunto de herramientas para poder implementar nuestras propuestas en estos sistemas.

- Todas las soluciones desarrolladas han sido probadas en diferentes casos de uso para demostrar su eficacia, además de haber usado parte de los resultados para resolver problemas semejantes en otros proyectos.

Los resultados y conclusiones obtenidos en esta tesis han contribuido a consolidar el conocimiento fundamental sobre la arquitectura del Computing Continuum, así como de temas afines, entre los que destacan áreas de especial interés como las el problema de asignación de tareas a nodos, aprendidaje por refuerzo, Smart Cities y la optimización energética. No obstante, a lo largo del desarrollo de esta tesis, han surgido numerosos temas que han quedado fuera de su alcance inicial. Además, a medida que se profundizaba en los distintos aspectos del Computing Continuum, aparecieron nuevas líneas de investigación y retos que podrían explorarse en el futuro.

La primera de ellas sería la del uso de técnicas mucho mas complejas de aprendizaje por refuerzo, como las basadas en redes neuronales (DeepRL). Esta versión mejorada de RL soluciona la limitación de que el espacio de estado tiene que ser discreto, pero añade un sobrecoste de memoria y computo difícilmente asumible por los dispositivos mas pequeños del Continuum. Por lo tanto, un punto interesante para futuros trabajos seria el de estudiar técnicas de DeepRL con redes muy pequeñas e implementarlas de la manera mas reducida posible para integrarlas en dispositivos con poca capacidad de cómputo.

Por otro lado, el orquestador desarrollado e implementado en Kubernetes demuestra un gran potencial, no solo para gestionar el Continuum y resolver el problema de asignación tarea-nodo, sino también para abordar otras cuestiones, como la toma de decisiones en temas de seguridad, la selección del software que implementará el servicio requerido, entre muchas otras. En este contexto, se considera que un camino interesante para futuros trabajos sería mejorar las capacidades del orquestador, tanto en cuanto a sus funcionalidades como a su semántica, de manera que pueda comprender una mayor variedad de restricciones y requerimientos en sus despliegues. Además, sería valioso incorporar una amplia gama de algoritmos que permitan resolver dinámicamente los problemas de decisión, adaptándose a entornos más complejos y variables. Precisamente, estos temas ya están siendo explorados como base para futuros trabajos tras la tesis. Entre los aspectos

destacados se encuentra la ampliación del análisis del problema de optimización, incorporando condiciones relacionadas con la seguridad y privacidad de cada nodo particular, lo que permitiría excluir nodos si resultan poco confiables según estos parámetros. Además, aunque algunos algoritmos sobresalgan frente a otros, se ha demostrado la utilidad de disponer de una variedad de métodos para resolver el problema de asignación con mayor o menor precisión, a cambio de memoria o tiempo. Por consiguiente, otra dirección de investigación futura que se está abordando es la meta-orquestación, que permitirá seleccionar dinámicamente el algoritmo adecuado en cada petición según el tamaño del problema, los requisitos de tiempo, la memoria disponible y otros parámetros relevantes.

Por último, en el problema de asignación de tareas a nodos, la investigación se ha centrado principalmente en métodos que buscan optimizar un único objetivo, aunque este pueda integrar varios factores mediante una suma ponderada o un estado conjunto. Sin embargo, no se ha explorado en profundidad el uso de métodos puramente multiobjetivo, que permitirían abordar de manera simultánea múltiples criterios independientes. Esto abre una línea de investigación prometedora, ya que los métodos multiobjetivo podrían ofrecer soluciones más equilibradas y flexibles en escenarios complejos.

Capítulo 3

Introducción

En la actualidad, la tecnología avanza a pasos agigantados y transforma constantemente la manera en que vivimos y trabajamos. Uno de los cambios más significativos es la creciente dependencia de los datos, que se han convertido en el núcleo de la toma de decisiones y en una herramienta esencial para optimizar procesos en todos los sectores. Esos datos se han convertido en un recurso fundamental, haciendo que su recopilación y procesamiento eficiente resulten imprescindibles. Gracias a los avances en tecnologías de hardware y al impulso hacia la digitalización de nuestra sociedad, conectar cada aspecto de nuestro entorno a Internet se presenta como un desarrollo muy prometedor. En este contexto surge el Internet de las Cosas (IoT, por sus siglas en inglés), una tecnología innovadora que busca conectar dispositivos cotidianos (las "cosas") a Internet para recoger y analizar los datos que generan. Estos dispositivos conectados conforman una red IoT, donde cada uno cuenta con conexión a Internet y, en algunos casos, con capacidad para comunicarse entre sí. Esto abre la puerta a nuevos paradigmas, facilitando escenarios como el despliegue de numerosos sensores y dispositivos en grandes instalaciones (como hogares, fábricas, o ciudades) que pueden ser monitoreados en tiempo real desde cualquier lugar del mundo.

Paralelamente al desarrollo del IoT, han surgido otras tecnologías y metodologías que se perfilan como pilares fundamentales en la construcción de un futuro completamente interconectado. Entre estas, destacan las técnicas de Big Data, que son cruciales para gestionar y analizar el enorme volumen de datos que generan las redes IoT. La capacidad de recopilar, almacenar y procesar esta gran cantidad de información permite descubrir patrones globales, predecir tendencias y tomar decisiones en tiempo real. Por otro lado, el Machine Learning se ha convertido en una herramienta indispensable para otorgar "inteligencia" a los dispositivos conectados a la red. Gracias al aprendizaje automático, los sistemas IoT pueden procesar sus datos de manera autónoma, detectar patrones, identificar anomalías e incluso responder a situaciones críticas. En el ámbito de la ciberseguridad, por ejemplo, el Machine Learning permite que estas redes reconozcan posibles amenazas y reaccionen de forma rápida y eficiente ante ciberataques, aumentando la resiliencia de los sistemas conectados.

Sin embargo, la adopción generalizada del IoT plantea diversos desafíos [1, 2], como pueden ser el problema de la latencia [3], dado que ciertas aplicaciones [4, 5] IoT requieren

cumplir con estrictos límites de retardo máximo para funcionar correctamente. Además, el uso del ancho de banda se ve limitado debido al gran volumen de dispositivos y datos que transitan por las redes actuales, lo cual afecta su eficiencia y rendimiento. Estas dificultades se deben en gran medida a que la infraestructura de IoT depende de servicios centralizados en la nube, alojados en grandes centros de datos con alta capacidad de procesamiento. Sin embargo, al encontrarse estos centros demasiado lejos de los dispositivos IoT, se generan cuellos de botella en el ancho de banda y, en algunos casos, latencias prohibitivas que limitan el rendimiento de las aplicaciones [6].

Para superar las limitaciones de la computación en la nube en entornos IoT, se necesita un nuevo enfoque que disminuya el consumo de ancho de banda y la latencia para el usuario final [7]. Así surge el paradigma de Edge Computing, una topología innovadora que acerca las tareas de procesamiento lo máximo posible a los usuarios. En este modelo, los dispositivos de borde colaboran para llevar a cabo parte del procesamiento de las tareas generadas por los dispositivos IoT cercanos, lo cual reduce de forma significativa tanto la latencia como la carga sobre la red [8]. No obstante, este tipo de redes se enfocan en aprovechar los dispositivos de borde, dejando de lado los evidentes beneficios que ofrece el ampliamente conocido Cloud Computing. Por esta razón surge recientemente un enfoque nuevo que pretende unir lo bueno de cada arquitectura, de manera que los dispositivos formen parte de una red piramidal donde todos formen un continuo de cómputo pudiendo distribuir el trabajo entre los dispositivos en función de las necesidades (latencia, computo, espacio, necesidad de hardware de aceleración). Esta unión de arquitectura es lo que se denomina Computing Continuum, el cual pretende respetar las bondades del Edge Computing pero sin relegar el papel fundamental que tiene el Cloud Computing para cargas de trabajo muy grandes, ofreciendo además alternativas de cómputo entre ambos enfoques (Fog Computing).

La integración de estos avances en el concepto de IoT, bajo los principios de Edge Computing, abre nuevas posibilidades que permiten una evolución del concepto original de los dispositivos IoT [9]. Normalmente, se considera que un dispositivo IoT tiene un comportamiento "pasivo", ya que sus tareas se resumen en recoger datos de diversas fuentes y enviarlos a la red [10]. En Edge Computing, se propone que los dispositivos IoT puedan tener también un comportamiento "activo", de forma que participen de manera inteligente en el procesamiento de los datos y en la lógica de las aplicaciones desplegadas en la red [11]. Un ejemplo transversal de la aplicación de estos avances son las Smart Houses o casas inteligentes, en las que la tecnología IoT combinada con Edge Computing permite una mayor autonomía y eficiencia en el funcionamiento del hogar. En una Smart House, los dispositivos IoT no solo recopilan información sobre el entorno (temperatura, iluminación o consumo de energía) sino que, al contar con capacidades de procesamiento local, pueden analizar esta información y tomar decisiones en tiempo real. Por ejemplo, un sistema de climatización puede ajustar la temperatura automáticamente en función de las preferencias del usuario y de los datos obtenidos de sensores de proximidad, o puede apagar dispositivos con alto consumo cuando el precio de la luz es elevado. Gracias al Edge Computing, estos dispositivos pueden ejecutar funciones avanzadas sin tener que depender completamente de un servidor central, lo cual reduce la latencia y permite una respuesta casi inmediata. Además, la implementación de estos sistemas en las

Smart Houses también mejora la seguridad, ya que los dispositivos no tienen que enviar información privada a servidores externos, y el uso de ancho de banda.

Sin embargo, las redes de Edge Computing, a diferencia de las de Cloud Computing, están formadas por un conjunto heterogéneo de dispositivos con características de hardware y posiciones geográficas muy diversas. Por lo tanto, para hacer un uso eficiente de los dispositivos de borde, es crucial desarrollar un orquestador de los nodos eficiente para que la asignación de recursos y tareas tengan en cuenta las características heterogéneas y los roles de los dispositivos de borde [12].

En cuanto a la gestión de recursos, uno de los componentes clave de la arquitectura Edge Computing es el proceso de asignación de tareas a nodos [13], el cual decide dónde deben procesarse las tareas de computación de la forma más óptima posible [14]. Teóricamente, este proceso es un problema de optimización combinatoria, denominado "problema de asignación de tareas", definido como el proceso de determinar dónde se realiza el cómputo de cada tarea para optimizar ciertos parámetros, como la latencia [15], el tiempo de ejecución, el coste, el consumo de energía [16], la posición geográfica [17], e incluso si el nodo se está moviendo[18] o se alimenta con una batería. El problema de asignación de tareas (*Task Assignment Problem*, TAP) puede formularse como un Problema de Asignación Generalizado (*Generalized Assignment Problem*, GAP) [19], que se ha demostrado ampliamente que es un problema NP-hard [20, 21].

El GAP puede resolverse con multitud de métodos y técnicas muy diferentes [22–24], como técnicas de optimización convexa, optimización de Lyapunov [25], algoritmo Húngaro [26] y algoritmos genéticos novedosos [27]. Además, han surgido nuevos métodos muy prometedores basados en técnicas de programación dinámica y machine learning [28], como el aprendizaje por refuerzo (RL) y el aprendizaje por refuerzo de redes neuronales (Deep RL) [29]. A pesar de ello, es difícil encontrar soluciones óptimas al problema de asignación de tareas, especialmente dada la prohibitiva capacidad computacional de los dispositivos que habitualmente se usan en IoT y Edge Computing, por lo que en la práctica se suelen utilizar técnicas basadas en heurísticas o métodos que buscan soluciones subóptimas. Una de las técnicas más comunes son los algoritmos Greedy, que proporcionan soluciones subóptimas pero con un bajo coste computacional, lo que compensa el resultado no óptimo. En algunos casos, si se conoce adecuadamente el problema es posible lograr soluciones muy próximas a la solución óptima. Por otro lado, otra manera de resolver problemas de optimización son los algoritmos basados en metaheurísticas, los más populares son los Algoritmos Genéticos que se inspiran en el proceso de selección natural.

Finalmente, una novedosa técnica denominada aprendizaje por refuerzo es de especial interés tanto en la realización de esta tesis como en las aportaciones más recientes en la literatura sobre sistemas inteligentes. Este nuevo método se basada en el aprendizaje automático (con matices de psicología conductista) y no forma parte de los conocidos paradigmas de aprendizaje supervisado y no supervisado. El objetivo del aprendizaje por refuerzo es aprender una política óptima, o casi óptima, que maximice la función de recompensa y proporcione un conjunto óptimo de acciones para diferentes estados del agente y condiciones ambientales. Los algoritmos de aprendizaje por refuerzo aprenden iterativamente a través de las recompensas inmediatas que reciben cada vez que realizan una acción en función de su estado, de esa manera aprenden siguiendo un enfoque de

prueba-error-recompensa. En algunos casos, no es posible hacer uso de un algoritmo de aprendizaje por refuerzo directamente, como en escenarios en los que el estado de los agentes es un gran número de variables, lo que hace ineficientes los algoritmos de Q-Learning, o incluso cuando las variables de estado no son discretas. Para aplicar RL en estos escenarios es necesario analizar adecuadamente el entorno del agente para adaptarlo a los requisitos de Q-Learning, por ejemplo, reduciendo el numero de variables del estado o discretizando las variables continuas en un conjunto finito de valores. Otra alternativa sería usar redes neuronales para modelar la Q-Table, permitiendo así las variables continuas, además de aumentar el rendimiento en la gestión de estados con un gran número de variables.

Aun así, los algoritmos de Aprendizaje por Refuerzo enfrentan un obstáculo común: el tiempo necesario para alcanzar la convergencia. Para llegar a una solución óptima, estos algoritmos suelen requerir múltiples iteraciones, y el uso de políticas con elementos aleatorios puede prolongar aún más este proceso [30, 31]. Sin embargo, en redes IoT donde los nodos son "activos" y pueden colaborar, esta limitación puede aliviarse. Los nodos en el Edge pueden compartir información sobre los estados que conocen o recurrir a otros nodos con mayor conocimiento para facilitar el inicio del aprendizaje, lo cual acelera la convergencia y reduce el riesgo de quedarse en soluciones subóptimas.

3.1 Objetivos

El desarrollo de esta tesis ha seguido un objetivo principal, del cual han derivado una serie de subobjetivos específicos. El objetivo principal se detalla a continuación:

> Análisis y diseño de soluciones para la gestión y el despliegue de nodos IoT y Edge capaces de administrar tareas de manera inteligente dentro de un ecosistema de Computing Continuum para optimizar los recursos de la mejor manera posible, adaptando dinámicamente la red a cambios en la infraestructura

Para lograr esta meta, se han planteado la siguiente serie de objetivos más específicos que han guiado las etapas de desarrollo de la presente tesis:

Objetivo 1: Analizar y estudiar características y restricciones presentes en los dispositivos utilizados en IoT y Edge Computing, para determinar las limitaciones para su uso en el Computing Continuum.

Objetivo 2: Analizar escenarios de IoT y Edge Computing emergentes relacionados con Smart Cities, Smart Houses, Smart Grids, asi como de las tecnologías involucradas.

Objetivo 3: Examinar los enfoques actuales de Edge Computing junto con casos de uso destacados, valorando las ventajas y desventajas de cada uno de ellos para adaptarlos al Computing Continuum.

Objetivo 4: Estudiar el problema de decisión referente a la distribución optima de las tareas de computo entre los diferentes nodos de una red, considerando toda una serie de restricciones y parámetros.

Objetivo 5: Analizar el estado del arte referente a los métodos para resolver el problema de asignación tarea-nodo, buscando puntos de mejora para los métodos más populares y la posibilidad de implementar varios de manera dinámica en un orquestador.

Objetivo 6: Integrar los métodos y técnicas mas relevantes identificados en sistemas de Computing Continuum y probar la eficacia en diferentes escenarios.

Objetivo 7: Validar las propuestas y los diseños definidos en casos de uso reales, donde se pueda comprobar la viabilidad en un despliegue en una infraestructura real.

3.2 Metodología

Para alcanzar estos objetivos, se ha optado por un enfoque distribuido, realizando pequeños incrementos de manera paralela en cada uno de ellos. Luego, se recopilaron y analizaron los resultados parciales, que sirven como fundamento para la elaboración de cada una de las publicaciones del compendio. Con lo anterior en mente, la etapa inicial de la tesis se centró principalmente en llevar a cabo un exhaustivo estudio del arte sobre los temas más relevantes del Edge-Fog Computing y el problema de asignación de tareas a nodos, con el fin de contextualizar todo el trabajo posterior. Una vez comprendida adecuadamente la arquitectura del Edge Computing y las técnicas más comunes para resolver el problema de asignación de tareas a nodos, se enfocó en diseñar escenarios de prueba para evaluar el funcionamiento de la técnica más destacada identificada (aprendizaje por refuerzo) e intentar modificar su diseño para mejorar su rendimiento en escenarios donde resulta menos eficiente.

Como resultado de todas las pruebas, se han definido varios casos de uso con arquitecturas de redes específicas (como la piramidal Edge-Fog-Cloud y el Edge desconectado, entre otras) y se han desarrollado diversos algoritmos para llevar a cabo el proceso de asignación de tareas a los nodos. Además, se han creado simuladores, herramientas y módulos que permiten probar de manera realista las propuestas, lo que incluso posibilita su integración en orquestadores populares como Kubernetes.

Para asegurar la utilidad práctica del trabajo realizado en esta tesis, se ha colaborado en diferentes proyectos tanto nacionales como europeos para validar las soluciones propuestas en el marco de cada proyecto. Entre ellos cabe destacar la colaboración con el proyecto europeo FLUIDOS (Grant. 101070473 [32]) en lo referente a gestión de tareas-nodos y Kubernetes, el proyecto europeo PHOENIX (Grant. 893079 [33]) sobre la gestión de dispositivos en edificios inteligentes y el proyecto nacional “Gemelo Digital para la Gestión Sostenible y Adaptación Al Cambio Global de la Agricultura Mediterránea” sobre la administración inteligente de dispositivos siguiendo el modelo de Gemelo Digital.

El resto del documento se organiza del siguiente modo. La primera parte del capítulo 4 ofrece una breve contextualización de los principales temas teóricos abordados en esta tesis. La segunda parte del capítulo 4 detalla los resultados derivados dentro de esta tesis, describiendo el análisis de los trabajos relacionados, los gaps identificados y cómo la tesis los aborda a través de sus resultados, incluido el compendio de publicaciones. El capítulo 5 presenta las conclusiones derivadas del desarrollo de la tesis y las futuras vías de trabajo que permiten los resultados producidos. El capítulo 6 incluye el resumen y el texto completo de las tres publicaciones que forman parte de la tesis por compendio.

Capítulo 4

Resumen de resultados

Este capítulo presenta los principales resultados obtenidos durante el desarrollo de esta tesis. Para contextualizarlos de manera adecuada, primero se introducen los conceptos fundamentales necesarios y, posteriormente, se ofrece una visión general del estado del arte, abarcando los trabajos relacionados encontrados en la literatura para cada uno de los problemas analizados.

4.1 Conceptos fundamentales

Antes de abordar los trabajos relacionados y presentar las aportaciones de esta tesis, resulta relevante proporcionar al lector una visión general de algunos conceptos fundamentales de este documento, permitiendo contextualizar de manera más clara las contribuciones realizadas. Los dos temas principales que se van a explicar son el problema de optimización con el que se ha estado trabajando, denominado problema de asignación de tareas, y los fundamentos en los que se basa la técnica de aprendizaje por refuerzo.

4.1.1 Problema de asignación de tareas

El principal problema que se enfrenta los sistemas de computación formados por un conjunto de dispositivos es la distribución idónea de las cargas de trabajo entre los diferentes dispositivos, teniendo en cuenta todo tipo de criterios para poder optimar el uso general de los recursos. Este problema se conoce como el Problema de Asignación de Tareas ("*Task Assignment Problem*", TAP), el cual se considera un problema de optimización combinatoria comúnmente conocido como el Problema de Asignación [42].

En términos generales, el problema de asignación se define como: Dado un conjunto de tareas y un conjunto de nodos, cada tarea puede asignarse a cualquiera de los nodos incurriendo en un coste y proporcionando un beneficio, siempre que no se excedan las capacidades del nodo. El objetivo es encontrar la mejor asignación tarea-nodo de todas las tareas que maximice el beneficio final sin exceder las capacidades de cada nodo. Las capacidades de un nodo pueden incluir, además de las tradicionales CPU y RAM, otras características como su ubicación, velocidad de CPU, latencia promedio, ancho de banda disponible, GPU o disponibilidad de RAM con corrección de errores (RAM-ECC).

En este contexto, lo ideal es que los orquestadores de sistemas de nodos intenten resolver el TAP para cada conjunto de aplicaciones que quieren desplegar, pero como resolver este problema suele ser costoso, intentan utilizar otra metodología, como las asignaciones uno a uno basadas en heurísticas u otros métodos más simplificados. En el caso del Computing Continuum, este problema se complica aún más por la naturaleza heterogénea de los dispositivos que forman parte de la infraestructura (desde teléfonos móviles hasta grandes centros de datos), la dispersión de las ubicaciones físicas entre nodos, el ancho de banda limitado en algunos nodos y muchos otros detalles. Sin embargo, las distintas capacidades de los elementos de este tipo de arquitectura pueden aprovecharse para distribuir los servicios y aplicaciones en el continuo en función de su prioridad y requisitos computacionales, minimizando así el coste de las aplicaciones al tiempo que se maximiza su rendimiento. En resumen, el TAP es un concepto crítico en los sistemas que distribuyen el cómputo entre múltiples dispositivos, y adquiere especial relevancia en escenarios basados en el Computing Continuum. Esto se debe a que las restricciones y consideraciones adicionales en este contexto hacen que el problema de optimización sea mucho más complejo de resolver.

Elementos

Como ya se ha mencionado, los elementos que constituyen la infraestructura del distribuidos, como Edge Computing o el Continuum, son muy diferentes tanto en capacidad como en diseño. En las zonas más cercanas al usuario tenemos dispositivos más pequeños con potencia limitada pero capaces de responder rápidamente a las peticiones (baja latencia), al otro lado tenemos servidores Cloud con gran capacidad de computación pero con mayor latencia y menor ancho de banda disponible. Entre ambos lados se despliegan diferentes dispositivos con capacidades intermedias para dar soporte a las necesidades de usuarios y desarrolladores. La figura 4.1 resume la distribución típica de dispositivos en el Continuum identificando los servicios que podrían desplegarse en cada punto en función de la latencia, el ancho de banda y la capacidad de cálculo.

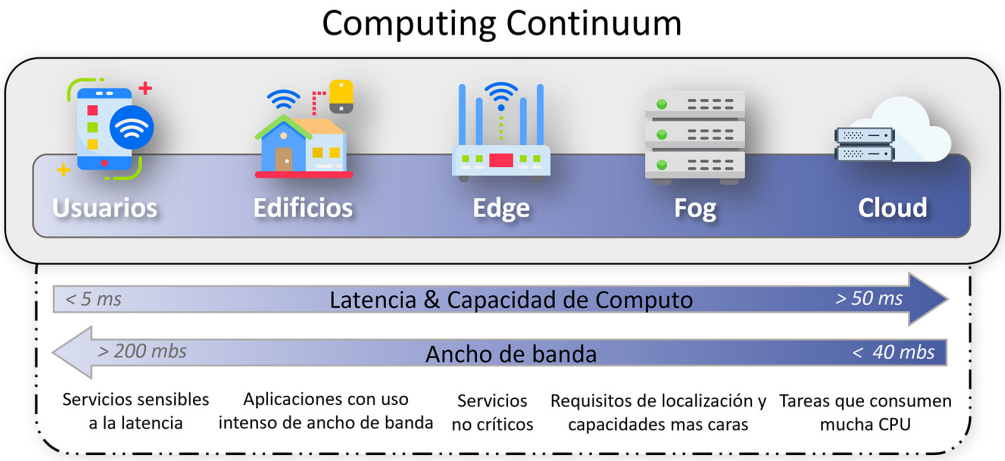

Figure 4.1: Computing Continuum.

Además, dependiendo de la capa, algunos dispositivos pueden proporcionar capacidades adicionales. Por ejemplo, algunos nodos Fog pueden tener GPU para acelerar las tareas de Deep Learning, mientras que algunos nodos Edge pueden no tener ciertas instrucciones necesarias para acelerar las tareas de cifrado (por ejemplo, AES-NI). También es importante tener en cuenta que la velocidad computacional de cada dispositivo es diferente, por lo que las mismas tareas pueden resolverse más rápido en un nodo más potente aunque tenga mayor latencia. Esta variedad de características debe ser tenida muy en cuenta por el orquestador a la hora de decidir a qué nodo asignar un servicio y se deben definir adecuadamente en el problema de optimización.

Definición Formal

El problema de asignación de tareas es un problema bien conocido y se puede formalizar como un Problema de Asignación Generalizado (GAP) [43], el cual es NP-hard [44]. El objetivo del problema es definir la mejor asignación de tareas a los nodos, por lo que la salida del algoritmo consistirá en una lista de tamaño S (número de tareas a desplegar), donde cada elemento denotará en qué nodo (del conjunto N) se despliega esa tarea. Por consiguiente, la solución de salida al problema se define como sigue:

$$tw = \{tw_0, tw_1, ..., tw_{n-1}, tw_s\} \tag{4.1}$$

where:

$$tw_x = \{w \mid w \in N\}$$

Por tanto, el problema de optimización de la asignación de S tareas a N nodos se formula inicialmente del siguiente modo:

$$\min \sum_{w=0}^{n} \sum_{t=0}^{s} a(w,t)\, c(w,t) \tag{4.2}$$

El objetivo es minimizar el coste, definido por la función de coste nodo-tarea $c(w,t)$, que representa el costo de asignar una tarea t a un nodo w. Esta asignación se expresa mediante la función $a(w,t)$, la cual indica si la tarea t ha sido finalmente asignada al nodo w, tal que:

$$a(w,t) = \begin{cases} 1 & \text{Si } tw_t \textit{ [eq 4.1]} \text{ es igual a } w \\ 0 & \text{En otro caso} \end{cases} \tag{4.3}$$

Sin embargo, la definición anterior del problema de asignación es insuficiente, ya que no contempla ninguna restricción en el proceso de asignación. Esto es claramente incorrecto, ya que es necesario considerar al menos las limitaciones de recursos de los dispositivos (como RAM, CPU, disco, etc.) para evitar sobrecargarlos. De esa forma una definición mas adecuada que incluya las restricciones de las capacidades de los dispositivos sería la siguiente:

$$\min \sum_{w=0}^{n} \sum_{t=0}^{s} a(w,t)\, c(w,t) \tag{4.4}$$

sujeto a:

$$\sum_{t=0}^{s} a(w,t) S^t_{cpu} \leqslant N^w_{cpu} \quad \forall w$$

$$\sum_{t=0}^{s} a(w,t) S^t_{ram} \leqslant N^w_{ram} \quad \forall w$$

$$\sum_{w=0}^{n} a(w,t) = 1 \qquad \forall t$$

En este caso el problema ya incluía las siguientes restricciones: Las asignaciones de nodos no pueden superar la capacidad de CPU de cada nodo; Del mismo modo, nunca se puede superar la capacidad de RAM de los nodos; No se puede asignar más de un nodo a cada tarea. Aun así, esta definición podría considerarse la más básica, ya que incluye solo los elementos mínimos para un despliegue inicial de tareas en un conjunto de nodos. En escenarios de Edge Computing, sería necesario tener en cuenta otras limitaciones de los nodos, como su ubicación geográfica, capacidades de hardware (como la presencia de GPU o TPU), latencia promedio, afinidades y anti-afinidades con otras tareas, entre otros factores adicionales. La siguiente formulación sería la mas adecuada para los problemas tratados durante esta tesis:

$$\min \sum_{w=0}^{n} \sum_{t=0}^{s} a(w,t)\, c(w,t) \tag{4.5}$$

sujeto a:

$$\sum_{t=0}^{s} a(w,t) S^t_{cpu} \leqslant N^w_{cpu} \qquad \forall w$$

$$\sum_{t=0}^{s} a(w,t) S^t_{ram} \leqslant N^w_{ram} \qquad \forall w$$

$$\sum_{w=0}^{n} a(w,t) = 1 \qquad \forall t$$

$$\sum_{w=0}^{n} a(w,t) \sum_{t'=0}^{s} a(w,t') \cdot \mathrm{I}(t' \in t_{\text{anti}}) = 0 \qquad \forall t$$

$$a(w,t) = 0 \qquad if S^t_{loc} \neq N^w_{loc}$$

$$a(w,t) = 0 \qquad if S^t_{reqCap} \subset N^w_{cap}$$

$$a(w,t) = 0 \qquad if S^t_{maxLat} > N^w_{lat}$$

$$a(w,t) \in \{0,1\}$$

En esta nueva definición se considerarían requisitos como la antiafinidad entre tareas, la ubicación geográfica deseada para la tarea, que el nodo posea el conjunto de capacidades requerido por la tarea (por ejemplo, GPU, AES-NI, RAM-ECC, TLS, etc.), y que la latencia promedio del nodo no exceda la latencia máxima permitida por la tarea.

Con todo lo mencionado, se configuraría un problema de asignación completo para los escenarios abordados en esta tesis. Como se ha explicado, este problema puede adaptarse fácilmente a otros contextos mediante la adición o modificación de sus restricciones. Así, se podrían considerar otros parámetros, como un límite de consumo energético, la presencia o ausencia de batería en el dispositivo o incluso características de seguridad, como la conexión cifrada o un factor de confianza mínima en las tareas, para excluir nodos con mala reputación.

4.1.2 Aprendizaje por refuerzo

Otro concepto clave destacado en esta tesis es el aprendizaje por refuerzo. Por esta razón, se dedicará esta sección a ofrecer una breve introducción a la teoría que lo sustenta para clarificar su funcionamiento. En primer lugar, explicaremos el Proceso de Decisión de Markov, después el aprendizaje por refuerzo y, por último, un ejemplo de uso de RL en un problema concreto de Edge Computing.

Uno de los enfoques de aprendizaje automático más utilizados es el aprendizaje supervisado, que utiliza ejemplos de entrada-salida para predecir la salida de nuevas entradas. Sin embargo, en algunos escenarios sólo es posible obtener información como recompensa por acciones específicas. Lo único que se conoce es el efecto de las acciones que se realizan y no se tiene un conjunto de datos de prueba etiquetados. La retroalimentación que se obtiene al realizar cada acción guía al sistema hacia las mejores acciones en función de su utilidad. El aprendizaje por refuerzo pretende aprender de forma iterativa la mejor acción a realizar en cada situación (estado) según una función de recompensa dada.

En los algoritmos de aprendizaje por refuerzo no es necesario definir pares de entrada y salida como en el aprendizaje supervisado, ya que el agente de RL aprende de forma activa durante la ejecución mediante prueba y error, tomando acciones y recibiendo recompensas en función de sus decisiones. El agente tiene una percepción (estado) del entorno (mundo real) al que está conectado y un conjunto de posibles acciones a realizar para cambiar el entorno. Así, en cada iteración el agente recibe como entrada el estado actual del entorno y luego selecciona una acción a realizar del conjunto de acciones. La acción cambiará el estado y producirá una recompensa que el agente utilizará para aprender si la acción fue la correcta. La versión clásica del comportamiento del aprendizaje por refuerzo se muestra en la figura 4.2.

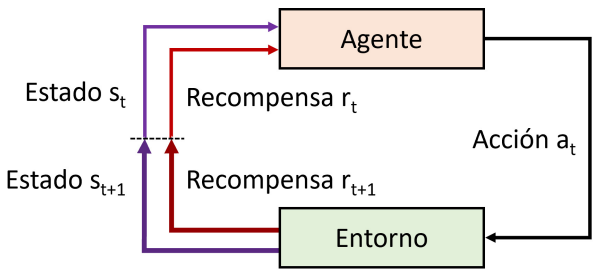

Figure 4.2: Proceso de aprendizaje por refuerzo.

Por tanto, el comportamiento de un agente puede resumirse como un sistema que debe elegir acciones que tiendan a aumentar las recompensas totales a largo plazo. El objetivo del agente será encontrar una política (π) que asigne estados a acciones y maximice una función de recompensa dada.

En las siguientes subsecciones introduciremos la teoría del aprendizaje por refuerzo y a continuación presentaremos nuestra propuesta de algoritmo basado en Q-Learning.

Proceso de decisión de Markov

Un Proceso de Decisión de Markov (MDP) es un framework matemático para describir un proceso de control estocástico. Proporciona una teoría para modelar la toma de decisiones secuenciales en un entorno en evolución determinado por una función de transición de probabilidad que asigna estados y acciones a otros estados. Los procesos de decisión de Markov no tienen memoria: la probabilidad de transición de un estado a otro sólo depende del estado actual y de la acción realizada, y es independiente de todos los estados y acciones anteriores. Este comportamiento de los procesos estocásticos se denomina propiedad de Markov y casi todos los problemas de refuerzo pueden formalizarse mediante procesos de decisión de Markov que satisfacen la propiedad de Markov.

Matemáticamente, un proceso de decisión de Markov se define de manera general como una 4-tupla $M =< S, A, \mathcal{P}, \mathcal{R} >$, donde:

- S es el espacio de estados, un conjunto finito de todos los estados posibles del sistema.
- A es el espacio de acciones, un conjunto finito de acciones que el agente puede realizar. Alternativamente, A_s es el conjunto de acciones disponibles desde el estado s.
- $\mathcal{P}$ es un conjunto de probabilidades de transición de un estado a otro para una acción determinada. $\mathcal{P}_a(s, s')$ denota la probabilidad de ir al estado s' desde el estado s mediante la acción a.
- $\mathcal{R}$ es la función de recompensa que determina el valor de la recompensa inmediata obtenida tras pasar del estado s al estado s' mediante la acción a, definida por $\mathcal{R}_a(s, s')$.

Dado un estado s_t en un intervalo de tiempo t, el agente seleccionará la acción a_t para cambiar al estado s_{t+1} según una política π, y luego recibirá una recompensa r_{t+1}. La política π es un mapeo del espacio de estados a las probabilidades de elegir una acción en particular. El objetivo del MDP es encontrar una política óptima π^* que maximice las recompensas acumuladas esperadas R. La recompensa acumulada esperada de un estado s_t puede definirse como la suma de las recompensas de estados futuros descontadas geométricamente utilizando el factor γ ($0 \leq \gamma \leq 1$) de la siguiente manera:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \tag{4.6}$$

En consecuencia, el valor de un estado puede definirse en términos de las recompensas futuras r_t que pueden esperarse. Este valor dará una estimación de "qué tan bueno" es un estado concreto para el agente, y dependerá de las acciones realizadas al utilizar la política π. La función de valor del estado s bajo una política π se define como sigue:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[R_t \mid s_t = s\right] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \;\middle|\; s_t = s\right] \tag{4.7}$$

Del mismo modo, la calidad de tomar la acción a bajo una política π se expresa como el valor esperado de la recompensa acumulada partiendo del estado s, tomando la acción a, y luego siguiendo la política π:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[R_t \;\middle|\; \begin{matrix} s_t = s, \\ a_t = a \end{matrix}\right] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \;\middle|\; \begin{matrix} s_t = s, \\ a_t = a \end{matrix}\right] \tag{4.8}$$

Como ya se ha mencionado, el agente MDP intenta encontrar la mejor política posible comparando funciones de valor. Se considera que una política π' es mejor que otra política π si para cada estado la función de valor esperado tiene un valor mayor. En otras palabras, una política π' es mejor o igual que π ($\pi' \geq \pi$) si y sólo si el valor de la función estado-valor de π' es mayor que la de π ($V^{\pi'}(s) \geq V^{\pi}(s)$) para todos los estados ($\forall s \in S$).

De hecho, siempre hay al menos una política que es mejor o igual a todas las demás, ya que la comparación de las funciones de valor proporciona una ordenación parcial sobre el espacio de políticas y el teorema del punto fijo de Banach demuestra la existencia y unicidad de una función de valor óptima y múltiples políticas óptimas [45]. Debido al teorema de Banach [46] podemos encontrar V^{π} y V^{*} aplicando iterativamente una aplicación contractiva a alguna función de valor inicial V para derivar una política óptima.

La mejor política posible, o el mejor conjunto de políticas, se denomina política óptima (π^{*}) y es el objetivo a alcanzar por el agente MDP. Así, las funciones de valor y calidad pueden reescribirse en términos de política óptima como sigue:

$$V^{*}(s) = \max_{\pi} V^{\pi}(s)$$

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a)$$

Las funciones de valor y calidad también pueden escribirse recursivamente según las ecuaciones de optimalidad de Bellman [47] de la siguiente manera:

$$V^{*}(s) = \max_{a} \sum_{s'} \mathcal{P}_a(s,s')[\mathcal{R}_a(s,s') + \gamma V^{*}(s')] \tag{4.9}$$

$$Q^{*}(s,a) = \sum_{s'} \mathcal{P}_a(s,s')[\mathcal{R}_a(s,s') + \gamma \max_{a'} Q^{*}(s',a')] \tag{4.10}$$

Aprendizaje por refuerzo

Existen diferentes formas de resolver un proceso de decisión de Markov, en particular soluciones basadas en algoritmos de Programación Dinámica, métodos de Monte Carlo y Diferencias Temporales.

Los métodos de programación dinámica (DP) son importantes en teoría, ya que sirven de base para todos los demás métodos, pero en la práctica su utilidad es limitada porque son costosos desde el punto de vista informático y requieren un conocimiento completo del modelo del problema (las probabilidades de transición y los valores de recompensa). Sin embargo, en un problema real el modelo no suele conocerse, ya que se descubre iterativamente mientras se busca la política óptima para el problema. Las soluciones basadas en Monte Carlo (MC) sólo requieren la "experiencia", secuencias de estados, acciones y recompensas de interactuar con el sistema, de cada episodio a aprender.

Dado que no se conoce el modelo del entorno, es necesario asegurarse de que el agente explora lo suficiente, no basta con seleccionar la acción que se estima que es la mejor ya que no podremos averiguar si hay alguna que sea aún mejor, por lo que se añade un componente aleatorio a la selección de acciones por parte del agente. Esto divide a los algoritmos en dos categorías en cuanto a la forma en que seleccionan las acciones, los métodos on-policy y los métodos off-policy.

En los métodos on-policy, el agente intenta encontrar la mejor política y la utiliza para decidir qué acciones tomar. En cambio, en los métodos off-policy el agente también explora, pero separa la política a buscar de la utilizada para tomar decisiones. De este modo, la estimación de la política puede seguir un enfoque greedy, mientras que la política de control puede ser ε-greedy, se selecciona una acción aleatoria con probabilidad ε o bien se selecciona la mejor acción. Los métodos off-policy garantizan el control de exploración y explotación equilibrando ambos tipos de selección de acciones. La exploración permite al agente mejorar su conocimiento actual sobre las acciones, con la esperanza de obtener beneficios a largo plazo, realizando acciones aleatoriamente para mejorar la precisión de las estimaciones del valor de las acciones. La explotación, por su parte, pretende elegir la acción con la mejor recompensa basándose en el conocimiento actual para garantizar la obtención de la mejor solución posible.

Los métodos de Programación Dinámica y Monte Carlo tienen sus propias ventajas e inconvenientes, como la necesidad de conocer el modelo del problema (DP) o tener que esperar hasta el final del episodio para calcular las estimaciones del valor de la acción (MC). La combinación de ambos métodos proporciona un método muy relevante de aprendizaje por refuerzo, el Aprendizaje por Diferencia Temporal (TD). La Diferencia Temporal aprende de la experiencia sin un modelo, como Monte Carlo, y los valores estimados se actualizan utilizando otras estimaciones sin esperar al final del episodio, hacen bootstrap como la Programación Dinámica. El método TD más sencillo, conocido como TD(0), es el siguiente:

$$V(s_t) = V(s_t) + \alpha\Big[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)\Big] \tag{4.11}$$

Donde α es la tasa de aprendizaje, entre 0 y 1, que determina la proporción entre el nuevo valor aprendido y el valor anterior.

Uno de los algoritmos más importantes de aprendizaje por refuerzo basado en la diferencia temporal es Q-Learning. El proceso de aprendizaje se centra en la optimización de la función acción-valor (Q) mediante una actualización iterativa basada en valores anteriores y en la diferencia temporal. El proceso general de actualización de la función Q se define por:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\Big] \tag{4.12}$$

Donde s_t es el estado actual, a_t es la acción actual, α es la tasa de aprendizaje y γ es el factor de descuento. Los Q-Values comienzan con valores predefinidos o aleatorios y se actualizan mientras el agente aprende hasta que converge en la política casi óptima.

Soluciones basadas en el aprendizaje por refuerzo

Tras la explicación los fundamentos del problema de asignación de tareas y el aprendizaje por refuerzo, en esta sección se presentará un ejemplo de aplicación de RL para resolver el TAP. El caso de uso que usaremos será el que se definió en la primera publicación del compendio, que se explicará en detalle en el siguiente capitulo. De manera resumida el escenario consiste en una serie de dispositivos Edge que tienen la capacidad de elegir donde realizar (*offload*) la tarea de computo que han recibido, si la procesan localmente, la envían a otro nodo Edge, la envían al nivel Fog o directamente a la Cloud. La decisión dependerá de las condiciones de la infraestructura que los dispositivos sean capaces de conocer, y está aspirará a optimizar el tiempo de procesamiento de la tarea y su consumo energético.

Está claro que en esencia es un problema de asignación de tareas típico, por lo tanto lo definimos de igual forma que el presentado en la ecuación 4.4 y proponemos resolverlo usando un enfoque de aprendizaje por refuerzo basado en Tabular Q-Learning. Cada dispositivo Edge ejecuta su propio algoritmo de aprendizaje por refuerzo para explorar individualmente la política óptima de offloading de tareas minimizando el coste acumulativo a largo plazo. La figura 4.3 muestra una vista general de un agente del sistema RL multiagente propuesto.

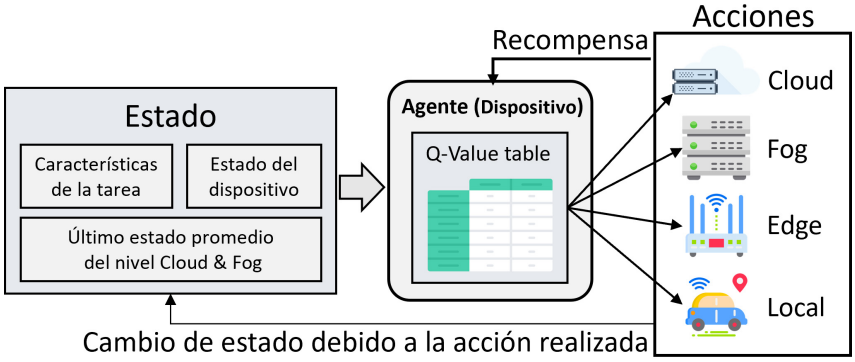

Figure 4.3: Agente Q-Learning agent

Cuando se recibe una tarea, el agente tendrá que decidir qué acción realizar, siempre que sea posible, entre las siguientes:

- Procesamiento local (*acción 0*): Ejecuta la tarea por sí mismo.
- Edge Offloading (*acción 1*): Envía la tarea a un nodo cercano.
- Fog Offloading (*acción 2*): Enviar la tarea a un servidor Fog.
- Cloud Offloading (*acción 3*): Enviar la tarea a un servidor Cloud.

Que puede resumirse en el siguiente vector:

$$a_t \in A = \{0, 1, 2, 3\}$$

La decisión dependerá del estado del entorno, que se basa en las características de la tarea, el estado del dispositivo y el último estado medio de Fog y Cloud (actualizado periódicamente). Los algoritmos clásicos de Q-Learning requieren un espacio de estado finito, por lo que es necesario discretizar valores continuos como el uso de la CPU y la latencia máxima. Un ejemplo de posible discretización de los parámetros continuos del problema son los siguientes:

- **MIPS requeridos por la tarea:** Hasta 20000 MIPS el valor es "*Low*", de 20000 a 100000 es "*Medium*" y por encima de 100000 es "*High*".
- **Máxima latencia permitida:** Hasta 6 segundos el valor es "*Low*", de 6 a 15 es "*Medium*" y por encima de 15 es "*High*".
- **Uso actual de MIPS del dispositivo:** Hasta 30000 MIPS el valor es "*Low*", de 30000 a 130000 es "*Medium*" y por encima de 130000 es "*High*".
- **Uso actual de la CPU del dispositivo:** De 0% a 25% de uso de CPU el valor es "*Low*", de 25% a 50% es "*Medium*", de 50% a 75% es "*Busy*" y de 75% a 100% es "*High*".
- **Último uso medio de CPU de los servidores Fog:** El porcentaje se divide en cuatro partes iguales del mismo modo que el uso de CPU del dispositivo.
- **Último uso medio de CPU del servidor Cloud:** El porcentaje se divide en cuatro partes iguales de la misma manera que el uso de la CPU del dispositivo.

El estado del agente puede formularse como el siguiente vector.

$$s_t \in S = \{taskMIPS, taskMaxLat, localMIPS, localCPU, AvgFogCPU, AvgCloudCPU\}$$

En escenarios reales es difícil conocer el estado en tiempo real de los servidores desplegados en las capas Fog y Cloud, ya que tendría un alto impacto en el uso de la red e incluso podría ser imposible de implementar sin afectar a la latencia del sistema. Por lo tanto, en este ejemplo sólo es posible obtener información estadística sobre el uso medio

de CPU de los últimos minutos (por ejemplo 5-10m), que se actualiza y envía a la red periódicamente.

El proceso de aprendizaje utiliza una tabla, la tabla Q-Value, para almacenar y consultar el valor de la función Q para cada estado y acción. Cuando se realiza una acción, el nuevo valor Q de la tabla se actualiza según la siguiente fórmula de actualización Q de un paso basada en 4.12:

$$Q(s_t, a_t) = (1 - \alpha)\, Q(s_t, a_t) \; + \; \alpha \left(C_t + \gamma \min_a Q(s_{t+1}, a)\right) \tag{4.13}$$

Esta fórmula actualiza el Q-Value utilizando un porcentaje $(1-\alpha)$ del Q-Value antiguo y un porcentaje (α) del valor aprendido, que está formado por la recompensa obtenida C_t y el Q-Value descontado (γ) de la mejor acción del nuevo estado. El comportamiento de la función de actualización indica que el algoritmo es off-policy, ya que para la elección de la acción se sigue una política ε-greedy mientras que para la actualización de los Q-Values se sigue un enfoque greedy.

Para este ejemplo, la recompensa obtenida tras la ejecución de una acción es una función definida a trozos de dos elementos que depende del tiempo de ejecución de la tarea. Si el tiempo de ejecución de la tarea $(T^t_{end} - T^t_{start})$ es inferior a su deadline (T^t_{dl}), la recompensa obtenida es una suma ponderada entre el tiempo de ejecución y el consumo de energía de la tarea (T^t_{energy}). En caso contrario, la recompensa es la misma pero multiplicada por un factor de penalización δ.

$$C_t = \begin{cases} (T^t_{end} - T^t_{start}) + \beta T^t_{energy} & T^t_{end} - T^t_{start} < T^t_{dl} \\ \delta \cdot ((T^t_{end} - T^t_{start}) + \beta T^t_{energy}) & \text{en otro caso} \end{cases} \tag{4.14}$$

La solución usando aprendizaje por refuerzo basado en Q-Learning que se propone en este ejemplo se detalla en el Algoritmo 1. El proceso comienza con la definición de los parámetros del algoritmo, como el factor de descuento γ de las recompensas, la tasa de aprendizaje α de la función Q y la tasa de exploración ε de la política de control. A continuación, el algoritmo entra en un bucle que itera sobre cada paso del proceso de aprendizaje. Cada paso t corresponde al proceso de decisión de offloading de una tarea T^t, y comienza obteniendo el estado actual del agente y determinando el conjunto de acciones posibles que el agente puede realizar (por ejemplo, los sensores no pueden hacer procesamiento local). Una vez que se dispone del estado y del conjunto de acciones, debe utilizarse la tabla Q-Value para determinar la mejor acción. Dado que el algoritmo es off-policy y sigue una política de control ε-greedy, la mejor solución (la de menor Q-Value) sólo se seleccionará si un número aleatorio e es mayor o igual que ε, en caso contrario la acción se elige aleatoriamente del conjunto de acciones factibles. La acción elegida establecerá dónde tiene lugar el offloading de la tarea, por lo que es necesario enviar la tarea y sus datos al nodo apropiado y esperar el resultado. Por último, se determina el nuevo estado del agente s_{t+1}, se calcula la recompensa C_t mediante la función definida a trozos 4.14 y se actualiza el Q-Value $Q(s_t, a_t)$ mediante la fórmula 4.13.

Algorithm 1: Algoritmo ε-greedy Q-Learning

Parameters: factor de descuento γ, ratio de aprendizaje α, ratio de exploración ε y factor de penalización δ

begin
  for *cada paso* t **do**
    Observar el estado actual s_t
    Determinar el conjunto de estados factibles A' de A
    $e \leftarrow$ número aleatorio de [0,1]
    if $e < \varepsilon$ **then**
      $a_t \leftarrow$ elegir una acción aleatoriamente de A'
    else
      $a_t \leftarrow \arg\min_{a \in A'} Q(s_t, a)$
    end
    Ejecutar la acción de offloading a_t
    Esperar a que se complete la tarea
    Observar el nuevo estado s_{t+1}
    Calcular la recompensa C_t mediante eq. 4.14
    Actualizar $Q(s_t, a_t)$ de acuerdo a eq. 4.13
  end
end

4.2 Estado del arte

En esta sección se abordarán, de forma general, los trabajos relacionados analizados para cada uno de los problemas estudiados a lo largo del desarrollo de la tesis. Cada tema constituirá la base bibliográfica utilizada en la elaboración de los artículos presentados, señalando en cada caso el principal "*gap*" identificado en la literatura y la solución que se ha propuesto en la realización de esta tesis.

4.2.1 Problema de asignación de tareas

Como se ha mencionado en varias ocasiones, el problema de asignación de tareas es un concepto clave en esta tesis, por lo que es importante conocer las distintas maneras de abordarlo según diferentes autores. Por lo tanto, en esta sección se presentarán algunos de los métodos más relevantes obtenidos al realizar una investigación de la bibliografía más reciente. Dado que encontrar soluciones óptimas para el problema de asignación de tareas es complicado, especialmente debido a la alta complejidad computacional en dispositivos IoT y Edge, muchos enfoques recurren a técnicas heurísticas que buscan soluciones subóptimas.

Los algoritmos Greedy son técnicas habituales que proporcionan soluciones subóptimas pero con un bajo coste computacional y, en algunos casos, logran soluciones muy próximas a la óptima. En [48] Rahabri et al. presentan un enfoque Greedy para resolver el problema de asignación de tareas utilizando un algoritmo de programación knapsack-based (GKS). En su trabajo proponen un algoritmo que ordena las tareas por su beneficio y peso y

asigna las de mayor valor para llenar la mochila. Evalúan la solución utilizando iFogSim, uno de los simuladores de fog computing más populares.

Otra forma de resolver problemas de optimización son los algoritmos basados en metaheurísticas, los más populares son los Algoritmos Genéticos que se inspiran en el proceso de selección natural. En [49] ZhenyueJia et al. formulan un problema de asignación cooperativa de tareas múltiples para vehículos aéreos no tripulados y lo resuelven mediante un algoritmo genético modificado. Concluyen, mediante simulaciones numéricas, que su algoritmo proporciona soluciones eficientes y factibles desde el punto de vista computacional.

Por otro lado, el aprendizaje por refuerzo es una técnica novedosa basada en el aprendizaje automático que no forma parte de los conocidos paradigmas de aprendizaje supervisado y no supervisado. El objetivo del aprendizaje por refuerzo es aprender una política óptima, o casi óptima, que maximice la función de recompensa y proporcione un conjunto óptimo de acciones para diferentes estados del agente y condiciones ambientales. Los algoritmos de aprendizaje por refuerzo aprenden iterativamente a través de las recompensas inmediatas que reciben cada vez que realizan una acción en función de su estado.

En [50] Tanmoy Sen et al. proponen utilizar RL para resolver el TAP teniendo en cuenta la puntualidad y el consumo de energía, y a continuación proponen una heurística para resolver el problema y diseñan el modelo de refuerzo basándose en el resultado de la heurística propuesta. Implementan el sistema de RL propuesto (RILTA) como un algoritmo Q-Learning y lo evalúan en el simulador iFogSim. Sus resultados de simulación muestran que RILTA puede reducir el tiempo de procesamiento de las tareas y el consumo de energía en torno a un 10-20% en comparación con otros métodos.

En algunos casos no es posible utilizar directamente el algoritmo RL, como en escenarios en los que el estado de los agentes es un gran número de variables, lo que hace ineficaces los algoritmos Q-Learning, o incluso cuando las variables de estado no son discretas. Para hacer frente a estas limitaciones, los investigadores proponen el uso de redes neuronales para modelar el proceso de aprendizaje del agente.

Jiadai Wang et al. proponen en [51] un esquema de asignación de recursos basado en Deep Reinforcement Learning (DRLRA), que puede asignar recursos de computación y de red de forma adaptativa, reducir el tiempo medio de servicio y equilibrar el uso de recursos en un entorno MEC variable. Su solución utiliza una red neuronal para calcular el valor Q del estado de cada agente y seleccionar una acción de acuerdo con la estrategia ε-greedy. Dado que utilizan redes neuronales, es necesario realizar un estado de entrenamiento antes de utilizar el algoritmo. Evalúan la solución utilizando una topología de red real de Topology Zoo y desarrollan un simulador en Python para probar el algoritmo DRLRA. Concluyen, mediante varias simulaciones, que en comparación con el algoritmo OSPF clásico en múltiples condiciones, su DRLRA propuesto consigue un rendimiento superior.

Del mismo modo, Xiaoyan Huang et al. proponen en [52] un esquema DRL multiagente distribuido con el objetivo de minimizar el consumo total de energía garantizando al mismo tiempo los requisitos de latencia y Liang Xiao et al. en [53] presentan un algoritmo de offloading RL y DRL para hacer frente a las smart jamming y heavy interference en MEC teniendo en cuenta la velocidad de transmisión y la potencia.

En general, los algoritmos de RL tienen un problema común, el tiempo de convergencia.

Este tipo de algoritmos requiere una serie de iteraciones para alcanzar la solución óptima y, en algunos casos, el uso de políticas de factores aleatorios puede tardar aún más tiempo en alcanzar la solución óptima. Mauricio Fadel Argerich et al. de NEC Laboratories proponen en [54] utilizar conocimiento externo para guiar las decisiones y la experiencia de los agentes. El método mejora el rendimiento durante el entrenamiento utilizando conocimiento experto o de dominio de una base de datos de conocimiento en forma de funciones programables. El sistema que proponen consiste en un agente RL junto con un "tutor" que accede a una base de datos de conocimiento definida por el experto del dominio. La base de conocimiento se compone de varias funciones de tipo restricción, que limitan el comportamiento del agente, y funciones de tipo guía, que expresan heurísticas del dominio que el agente utilizará para guiar sus decisiones, especialmente en momentos de gran incertidumbre. Durante sus primeros pasos, el agente utiliza estas funciones de conocimiento para decidir la mejor acción, guiando su exploración y proporcionando un mejor rendimiento desde el principio. En sus pruebas, Tutor4RL consigue una recompensa más de tres veces superior al principio de su entrenamiento que un agente sin conocimiento externo.

Por último, existen otro enfoques en la literatura que utilizan otro tipo de técnicas novedosas para resolver el TAP. En [55] Dadmehr Rahbari et al. utilizan árboles de clasificación y regresión para resolver el problema y utilizan el simulador Cloudsim para evaluarlo. En [56] Mainak Adhikari et al. diseñan una estrategia de offloading dependiente del retardo (DPTO) asignando una prioridad a cada tarea en función de su deadline, priorizando las tareas sensibles al retardo y minimizando el problema de inanición de las tareas de baja prioridad. En [57] Lindong Liu et al. proponen un enfoque de aprendizaje automático supervisado para resolver el problema de programación de tareas basado en la técnica de minería de datos de clasificación. Desarrollan un nuevo algoritmo de clasificación (I-Apriori) basado en el algoritmo Apriori e implementan un modelo de programación de tareas basado en él. Haibin Zhang et al. proponen en [58] el uso de una blockchain junto con el algoritmo DRL para mejorar la seguridad entre los servidores de borde en una red ultradensa.

En cuanto a la arquitectura edge computing, existen ejemplos en la literatura de diferentes enfoques tanto para la distribución de dispositivos como para el proceso de offloading. Liang Tong et al. proponen en [59] una típica arquitectura jerárquica edge-cloud con dispositivos móviles en la capa inferior (edge), cloudlet en la capa intermedia y centro de datos en la capa superior (cloud). Definen y evalúan un algoritmo de distribución de cargas de trabajo para garantizar una utilización eficiente de los recursos de la nube. En [60] Wen Sun et al. diseñan una red Edge con Digital Twin en 6G y proponen un esquema de offloading móvil basado en DRL para reducir la latencia en aplicaciones MEC.

Gap y Solución Propuesta

Como se explicó en el apartado anterior, existen diversas arquitecturas y técnicas para resolver problemas de asignación, cada una con sus propias ventajas y desventajas. Entre ellas, las técnicas basadas en inteligencia artificial son particularmente prometedoras, destacando el aprendizaje por refuerzo por su simplicidad y su capacidad de aprendizaje

autónomo a través de recompensas. No obstante, la convergencia hacia una solución óptima en métodos basados en Q-Learning representa un aspecto crítico, especialmente en las etapas iniciales del proceso de aprendizaje, cuando la estabilidad de la solución aún es limitada.

Para la primera contribución científica de esta tesis, se propone abordar las limitaciones del aprendizaje por refuerzo en la resolución del problema de asignación de tareas en una arquitectura Edge-Cloud similar a las anteriormente descritas. Para ello, se plantea un sistema multiagente colaborativo de múltiples capas, donde cada capa cuenta con un nivel de conocimiento de la infraestructura que depende de su nivel en la jerarquía. Así, los agentes de nivel inferior (comúnmente los ubicados en el Edge) podrán realizar una acción especial: delegar la toma de decisiones a un agente de nivel superior y luego aprender de los resultados. Esto permite que el agente con mayor conocimiento oriente y capacite al agente en el Edge, logrando que, en las primeras etapas de aprendizaje, este pueda alcanzar mejores soluciones y converger rápidamente hacia una solución óptima, reduciendo además el impacto de acciones inadecuadas sobre los usuarios.

4.2.2 Simulador de energía

Durante el desarrollo de la tesis surgió la necesidad de probar algunas de las propuestas en un entorno donde la gestión y el consumo de energía fueran aspectos cruciales. A partir de esta necesidad, se planteó la creación de un simulador propio que permitiera evaluar el rendimiento de los métodos propuestos bajo condiciones específicas de consumo energético. Antes de esta decisión, se analizaron las soluciones existentes y se evaluó su viabilidad para los casos de uso diseñados.

El interés en los simuladores de energía surge del hecho de que el consumo energético es un aspecto crítico en el mundo actual, ya que nos enfrentamos a los retos del cambio climático y el agotamiento de los recursos naturales. Como resultado, existe un creciente interés [61–64] en optimizar el consumo de energía para reducir las emisiones de carbono y crear un futuro más sostenible. El uso de fuentes de energía renovables, como los paneles solares, y el control inteligente del edificio [65] y los dispositivos domésticos se han vuelto cada vez más populares como una forma de lograr este objetivo.

El control inteligente de los dispositivos de un edificio y una Smart House [66] es un planteamiento prometedor que puede utilizarse para optimizar el consumo de energía. Mediante el uso de sensores avanzados y sistemas de control automatizados, el uso de la energía puede supervisarse y gestionarse con mayor precisión. Los dispositivos inteligentes pueden programarse para apagarse o reducir el consumo de energía cuando no están en uso, ahorrando energía y reduciendo las emisiones de carbono. Además, los dispositivos inteligentes pueden aprender los patrones de consumo de energía y hacer ajustes en consecuencia, lo que los convierte en una poderosa herramienta para optimizar la energía [66].

Aunque probar algoritmos en entornos reales es esencial para validar la eficacia de las propuestas, puede ser una tarea difícil [67]. La complejidad de las pruebas en entornos reales no se limita únicamente a los componentes físicos implicados, sino que también abarca los riesgos inherentes y las posibles consecuencias económicas de la exper-

imentación, así como las dificultades para realizar pruebas a largo plazo que permitan recopilar datos suficientes para el análisis. Esto subraya la importancia de emplear métodos de prueba basados en simuladores que puedan evaluar exhaustivamente los algoritmos en un contexto simulado realista a un coste muy bajo, permitiendo la repetibilidad y la realización de estudios a largo plazo en poco tiempo.

Existen varios estudios [68] y software desarrollados para simular el consumo de energía en edificios residenciales. El objetivo de estos programas y herramientas es optimizar el uso de la energía y reducir los costes energéticos, garantizando al mismo tiempo el confort y la seguridad de los ocupantes. Uno de los programas más utilizados para la simulación energética de edificios es EnergyPlus [69, 70], un programa de simulación energética de edificios desarrollado por el Departamento de Energía de Estados Unidos. EnergyPlus utiliza un enfoque modular para la simulación energética de edificios y puede simular una amplia gama de sistemas de calefacción, ventilación y aire acondicionado, incluidas bombas de calor geotérmicas y aerotérmicas, calderas, hornos, etc. Otro software popular es TRNSYS (Transient System Simulation Tool) [71]. TRNSYS es un programa de simulación dinámica que simula el comportamiento transitorio de los sistemas a lo largo del tiempo. Puede modelizar una serie de sistemas de edificios, como sistemas de calefacción, ventilación y aire acondicionado, sistemas de energías renovables y sistemas de almacenamiento térmico.

Además, existen otras herramientas de simulación para el análisis de la energía y los edificios. EnergyPLAN [72, 73], por ejemplo, es un programa que modela los sistemas energéticos de regiones enteras, incluidos los hogares, y permite simular distintos escenarios de producción y consumo de energía, así como evaluar el impacto de diversas políticas y tecnologías. Simergy [74] es otra herramienta de simulación que predice el rendimiento de diseños de edificios o remodelaciones antes de que se construyan, mientras que OpenStudio [75] es un conjunto de aplicaciones gratuitas y de código abierto para el análisis energético de edificios que funciona con SketchUp y EnergyPlus. Y, por último, Power TAC [76], una plataforma de competición de código abierto que simula los mercados eléctricos y permite a los investigadores desarrollar agentes de software inteligentes que representan a distintas entidades del mercado para optimizar los beneficios o satisfacer las necesidades energéticas. Power TAC proporciona un entorno realista con dinámica de precios, fluctuaciones de la oferta y la demanda, y factores externos como las condiciones meteorológicas. La plataforma evalúa y compara el rendimiento de los agentes, fomentando el desarrollo de estrategias que maximicen la utilidad y aborden los retos de los mercados eléctricos, como la integración de las energías renovables y las repercusiones políticas.

Estos simuladores son sólo algunos ejemplos de las muchas herramientas que existen para simular el consumo de energía en los hogares, incluso utilizando datos del mundo real [77]. Cada herramienta tiene sus puntos fuertes y débiles, y puede ser más adecuada para determinados tipos de simulaciones o para determinados tipos de edificios, pero en general suelen ser herramientas excesivamente complejas que dificultan su uso.

Gap y Solución Propuesta

Los simuladores de consumo energético para viviendas son una herramienta importante para comprender y optimizar el uso de la energía en edificios residenciales. Resuelven el problema científico de modelar con precisión el uso de la energía en las casas y contribuyen al avance científico al proporcionar una plataforma para que los investigadores prueben diferentes algoritmos en todo tipo de escenarios. Sin embargo, los simuladores más conocidos suelen ser demasiado complejos de utilizar y modificar, lo que dificulta a los investigadores la creación de prototipos y la experimentación con nuevos algoritmos sin dedicar mucho tiempo a comprender las características del simulador. Por esta razón, proponemos la creación de un simulador que sea lo más simplificado posible para que sea fácil de entender y permita a los investigadores centrarse en aplicar y comparar las nuevas soluciones y métodos que están desarrollando.

En este contexto, se ha desarrollado un simulador basado en Java (Sim-PowerCS) que sirve como herramienta simplificada para probar y desplegar diferentes fuentes y consumidores de energía en diversos escenarios. Sim-PowerCS nos permite definir múltiples fuentes de energía, como paneles solares y baterías, y los servicios que generan consumo, que luego podemos utilizar para probar la eficacia de distintos algoritmos de gestión. En concreto, Sim-PowerCS está diseñado para ayudarnos a gestionar por separado las fuentes de energía comprando a la red, utilizando baterías o generadores, así como los servicios de control apagando dispositivos, retrasando el encendido, reduciendo la potencia, etc. Una de las principales ventajas de utilizar el simulador es que podemos evaluar individualmente el rendimiento de distintos algoritmos de gestión energética y mecanismos de control de servicios inteligentes en un entorno controlado, sin necesidad de desplegarlos en el mundo real. Esto nos permite iterar rápidamente y afinar nuestro enfoque, reduciendo la probabilidad de cometer errores costosos y que requieren mucho tiempo. Sim-PowerCS permite a los usuarios definir fuentes de energía y servicios, y probar distintos algoritmos para gestionar las fuentes de energía y controlar los servicios. A continuación, los usuarios pueden ejecutar simulaciones procedimentales basadas en sus selecciones, evaluar los datos generados y ajustar su enfoque según sea necesario para optimizar la gestión de la energía y el control de los servicios.

4.2.3 Gestión de energía en una casa inteligente

La gestión de la energía en edificios y casas inteligentes desempeña un papel crucial en la optimización del consumo de energía y la mejora de la eficiencia general, como demuestra la variedad de soluciones propuestas por los investigadores. En esta sección, presentamos una visión general del estado del arte de los algoritmos de gestión de la energía en edificios/casas inteligentes, clasificándolos en función de la metodología subyacente.

Algoritmos basados en el análisis de datos

Los algoritmos basados en datos aprovechan técnicas numéricas y de minería de datos bien conocidas, como el análisis de series temporales, el aprendizaje automático y el reconocimiento de patrones, para extraer ideas significativas de la información recopilada.

Mediante el análisis de datos históricos y en tiempo real, estos algoritmos son capaces de detectar patrones sencillos, lo que permite realizar previsiones precisas, detectar anomalías y optimizar la energía.

El análisis de series temporales es un método clásico utilizado para analizar patrones dentro de una secuencia de puntos de datos recogidos a lo largo de un intervalo de tiempo. Técnicas como los modelos de medias móviles autorregresivas integradas (ARIMA), la descomposición estacional de series temporales (STL) y el análisis de Fourier se emplean habitualmente para identificar tendencias, estacionalidad y otros patrones temporales. En [78] Nepal et al. proponen un modelo híbrido que comprende ARIMA y una técnica de agrupación (K-means) para predecir la carga eléctrica de los edificios de una universidad. Afirman que la combinación de clustering y ARIMA aumenta el rendimiento de la previsión en comparación con el uso de ARIMA únicamente. Asimismo, Panapongpakorn y Banjerdpongchai [79] utilizan ARIMA junto con redes neuronales para implementar un complejo sistema de previsión de carga a largo plazo, y comparan el rendimiento utilizando criterios estadísticos de tres versiones de sus modelos híbridos para determinar cuál es el mejor para un sistema de microgestión. Además, Di Silvestre y Riva Sanseverino [80] diseñan un modelo de optimización del almacenamiento de energía utilizando el análisis de Fourier para problemas de suministro óptimo de energía en redes inteligentes y concluyen que los resultados de diferentes aplicaciones en un sistema de tamaño medio muestran que la representación de variables propuesta es bastante eficaz y constituye el primer paso hacia una nueva forma de formular variables en problemas de suministro (dispatch).

El control predictivo por modelos (MPC) es un método bien establecido para el control con restricciones. Utiliza un modelo matemático del edificio y sus sistemas energéticos para predecir el consumo futuro y optimizar las acciones de control. El MPC puede tener en cuenta diversos parámetros, como el confort de los ocupantes, las condiciones meteorológicas, los precios de la electricidad y las limitaciones de los equipos, para lograr un rendimiento óptimo. Un trabajo notable en esta área es [81], que proporciona una visión en profundidad de los MPC y presenta un marco claro y robusto que resume los pasos necesarios para formular el problema de control, especialmente para el control y la gestión de la térmica necesaria en los edificios. Además, en [82] utilizan la lógica MPC, basada en previsiones meteorológicas, para el análisis de cargas en un sistema doméstico sin conexión a red. Su estrategia de control tiene como objetivo minimizar los costes finales, manteniendo el confort ambiental óptimo en la vivienda, optimizando así el uso de fuentes renovables. También comparan su sistema con un sistema estándar basado en reglas para concluir que su propuesta reduce el uso de energía de origen fósil y supera al control basado en reglas. Asimismo, Zhou et al. [83] presentan un enfoque similar al anterior, destinado a una estrategia de gestión energética predictiva de una comunidad inteligente, pero en este escenario utilizan la información obtenida de los sistemas de transporte inteligentes para estimar la próxima ocupación del edificio y así anticiparse a los picos de consumo debidos a la carga de vehículos eléctricos (VE).

La regresión lineal basada en maquinas de vectores de soporte (SVR) es un método de regresión basado en el aprendizaje automático especialmente útil para la previsión de la carga. Resulta especialmente eficaz para tratar relaciones no lineales entre las variables de entrada y el consumo eléctrico. SVR utiliza un subconjunto de datos de

entrenamiento llamados vectores de soporte para construir un modelo de regresión, lo que permite una predicción precisa de la demanda futura. Ejemplo de ello son los trabajos [84] y [85], que proponen el uso de SVR para estimar el consumo energético en el primer caso de un laboratorio y en el segundo de edificios en una región concreta de China. Ambos trabajos concluyen que, con suficiente información histórica, SVR es capaz de identificar patrones y predecir con precisión las demandas futuras, lo que permite una gestión y optimización eficaces de la electricidad. En determinados casos, como el de un espacio muestral altamente no lineal, el uso de SVR estándar no basta para construir modelos precisos. Por esa razón, Zhong et al. [86] proponen modificar la estrategia de modelado común con el fin de utilizar múltiples mapeos para transformar entre espacios de características y finalmente acercarse al espacio de características óptimo, ya que un solo mapeo no puede garantizar que el espacio de características correspondiente sea óptimo.

Las redes bayesianas (BN) son modelos gráficos probabilísticos que utilizan grafos acíclicos dirigidos (DAG) para representar y razonar sobre conocimientos y dependencias inciertos. En [87], presentan una técnica BN aplicada para seleccionar la unidad primaria de HVAC más eficiente energéticamente, en lugar de utilizar un sistema tradicional basado en el conocimiento y la experiencia de un diseñador. Demuestran que, tras entrenar adecuadamente la BN, el resultado es similar al de los sistemas tradicionales pero con mayor eficiencia, y el estudio también demuestra la viabilidad y capacidad del diseño de edificios basado en datos. Del mismo modo, Shoji et al. [88] describen brevemente cómo utilizar BNs para aprender patrones de comportamiento de los usuarios con el fin de priorizar el funcionamiento de algunos electrodomésticos bajo restricciones de consumo de energía. Además, Amayri et al. [89] proponen un enfoque basado en redes bayesianas para determinar la detección de ocupación en edificios. Aprovechando los datos de los sensores y los patrones históricos de ocupación, el algoritmo ajusta de forma inteligente los sistemas de calefacción (producción de agua caliente), lo que se traduce en un ahorro sustancial de energía.

Los métodos de Clustering, como la técnica de agrupación k-vecinos y el agrupamiento jerárquica, se utilizan para agrupar patrones similares. Al identificar clusters o segmentos en los datos, estas técnicas facilitan la creación de perfiles energéticos y permiten el desarrollo de estrategias de gestión personalizadas para diferentes grupos de usuarios o tipos de edificios. Así, Ayenew et al. [90] utilizan K-mean para comprender y predecir la carga agregada de las subestaciones de una zona urbana de Etiopía, donde se producen frecuentes cortes de electricidad debido a la sobrecarga de los sistemas de transmisión y distribución. Con la solución que proponen, son capaces de identificar las horas de demanda intermedia y pico, y los clientes potenciales para la gestión del cambio de carga de la demanda basada en los precios, demostrando que un aumento de los precios de la electricidad en las horas punta conduce a una reducción de la demanda de electricidad y, por tanto, se puede reducir la carga, mejorando la fiabilidad y la estabilidad de la red. Del mismo modo, Ayub et al. [91] implementan un algoritmo de planificación de carga basado en la agrupación K-mean y programación lineal entera para programar los electrodomésticos de los consumidores para el día siguiente. Evalúan la propuesta utilizando un conjunto de datos reales disponibles públicamente y los resultados de la simulación muestran que el enfoque funciona mejor, aumentando así la confianza y la

continuidad del sistema.

Algoritmos basados en optimización

Las técnicas de optimización se utilizan ampliamente en muchos problemas de decisión, incluida la gestión de la energía, en los que se busca el mejor resultado posible para un problema determinado. La investigación actual se ha centrado en diversos enfoques de optimización, como la programación lineal, los MILP, la programación dinámica y los algoritmos genéticos.

La programación lineal (LP) es una técnica muy utilizada para resolver problemas de optimización. Consiste en formular una función objetivo y un conjunto de restricciones lineales para optimizar la asignación de recursos y el uso de la energía. Umetani et al. [92] desarrollan un algoritmo basado en LP sobre un modelo de red espacio-temporal para programar la carga y descarga de vehículos eléctricos en un tiempo de computación limitado. Tras comparar su propuesta con los resultados obtenidos con un solucionador MILP exacto, demuestran que es posible utilizar métodos heurísticos basados en LP para resolver problemas de optimización complejos de forma rápida y con resultados muy cercanos al óptimo. De forma similar, Bio Gassi y Baysal [93] formulan un modelo LP complejo que combina electricidad, calefacción y refrigeración para controlar un sistema de gestión energética de microrredes, con especial consideración de los sistemas de almacenamiento, como las baterías, ya que son un aspecto importante [94] de este tipo de problemas.

La programación lineal de enteros mixtos (MILP) amplía la programación lineal permitiendo que las variables de decisión tomen valores discretos. MILP es particularmente útil para los problemas que implican decisiones binarias o elecciones discretas, y es mucho más ampliamente utilizado en los sistemas de optimización de LP. Un trabajo notable que hace uso de MILP en la gestión de la energía es [95], que presenta el modelado y diseño de un sistema modular de energía y su integración a una red fotovoltaica híbrida conectada a la red basada en un microgrid eólica-batería. El modelo de planificación es una estrategia que prioriza la generación de energía, definida como un MILP teniendo en cuenta dos etapas para la carga adecuada de las unidades de almacenamiento basadas en datos de previsión a 24 horas vista. Siguiendo el mismo enfoque, Javadi et al. [96] diseñan un Sistema de Gestión de Energía Doméstica (HEMSs) basado en MILP para la auto-programación óptima en presencia de generación de energía fotovoltaica y baterías.

La programación dinámica (DP) es un método para resolver problemas de optimización dividiéndolos en subproblemas más pequeños y resolviéndolos recursivamente. Tischer y Verbic [97] utilizan la programación dinámica para el control de una casa inteligente, que está equipada con una pila de combustible utilizada para la cogeneración de calor y electricidad, un sistema fotovoltaico, un coche eléctrico, una batería y una unidad de almacenamiento de energía térmica. Mediante simulaciones demuestran que su algoritmo obtiene buenos resultados aunque su velocidad de cálculo es lenta, también discuten cómo disminuir el tiempo de cálculo aproximando la influencia de variables aleatorias. Además, DP es bastante útil para resolver problemas de carga de baterías, en escenarios domésticos [98] y vehículos eléctricos [99], ya que proporciona una forma de elegir la forma más adecuada de cargar las baterías minimizando los costes y maximizando la vida útil.

Los algoritmos genéticos (GA) son técnicas de optimización inspiradas en la naturaleza, estos algoritmos emplean principios de selección natural, cruce y mutación para buscar soluciones óptimas o casi óptimas en problemas de optimización complejos con bajos requerimientos computacionales. El uso de GA es más común en la planificación y el reajuste del consumo en viviendas o edificios, como puede verse en los trabajos de [100] y [101]. Además, Arabali et al. [102] proponen una estrategia basada en GA para controlar de manera óptima la calefacción, ventilación y aire acondicionado (HVAC) con un sistema híbrido de generación renovable y almacenamiento de electricidad.

Algoritmos basados en inteligencia artificial

Las técnicas de Inteligencia Artificial han revolucionado las técnicas de gestión de la energía aprovechando grandes conjuntos de datos y algoritmos complejos. Estas técnicas permiten la toma de decisiones inteligentes, la detección de anomalías y el control adaptativo de forma directa y precisa.

Una de las técnicas mas simples de IA son los árboles de decisión, algoritmos de aprendizaje automático sencillos pero eficaces que se utilizan en la gestión para tareas como la predicción del consumo, el diagnóstico de averías y la clasificación de cargas. Se basan en la partición recursiva de datos según distintos atributos, lo que permite generar reglas que guían el sistema de gestión de decisiones. Ferrández-Pastor et al. [103] proponen un sistema web para la previsión automatizada del consumo eléctrico para múltiples instancias (edificios) utilizando un enfoque de árbol de decisión que incluye condiciones basadas en variables disponibles (consumo eléctrico, estado del edificio, temperatura exterior, etc.). Indican una disminución del error de previsión (hasta 1,75 veces) debido al uso del árbol de decisión para la selección del modelo de previsión. Adicionalmente, en [104] se presenta un trabajo similar pero con un enfoque especial en sistemas IoT donde se pueden aprovechar completamente los beneficios de estas propuestas. Incluyen, además de la estimación del consumo futuro, la previsión meteorológica ya que el esquema incluye la generación por paneles solares y aerogeneradores, y concluyen que la arquitectura de infraestructuras IoT debería ser un requisito para este tipo de sistemas para poder rendir al máximo.

Por otro lado, los algoritmos de Gradient Boosting, como XGBoost (eXtreme Gradient Boosting) y LightGBM (Light Gradient-Boosting Machine), son algoritmos que construyen iterativamente un conjunto de modelos predictivos débiles, cada uno centrado en minimizar los errores cometidos por los modelos anteriores. Los algoritmos de Gradient Boosting destacan en la captura de patrones complejos y el logro de una alta precisión de predicción. Como puede verse en [105], en el contexto de la predicción del consumo, existen varios modelos diferentes (por ejemplo, LightGBM, CatBoost y XGBoost) que pueden utilizarse para desarrollar soluciones. En el trabajo antes mencionado, cada uno de ellos es examinado y probado con diferentes hiperparámetros para determinar cuál es el más adecuado o, al menos, cuál funciona mejor para un tipo concreto de datos, concluyendo que XGBoost obtiene los mejores resultados cuando se entrena en el conjunto de datos seleccionado. Además, Lu et al. [106] diseñan una predicción a corto plazo del consumo en una torre bocatoma empleando un modelo mejorado de boosting de gradiente extremo, y comparan el rendimiento del método con cinco modelos de referencia, mostrando que el

error porcentual absoluto medio del modelo propuesto es del 4,85%, el más bajo de todos los modelos. Además, en [107] se utilizan los tres métodos comunes de gradient boosting mencionados anteriormente con el fin de implementar una solución de control y previsión en unas plantas de energía solar en una red inteligente, y tras un adecuado entrenamiento y comparación utilizando varias metodologías, demuestran cómo los métodos de gradient boosting ofrecen ventajas considerables como una alta precisión y un rápido aprendizaje.

Entre todas las técnicas ya citadas, las Redes Neuronales Artificiales (ANN) son probablemente una de las más importantes en la investigación actual, no solo en la gestión de la energía, sino también en muchas otras áreas [108]. En el contexto de la gestión de la electricidad, las ANN son útiles porque pueden entrenarse utilizando datos históricos para predecir la demanda futura con gran precisión. Hay un gran número de arquitecturas y clases de redes neuronales artificiales, incluyendo fully connected networks, radial basis function networks, feedforward networks, convolutional networks, recurrent networks y muchas más. Un ejemplo de red fully connected (arquitectura back-propagation de tres capas) se muestra en [109], donde los autores desarrollan un sistema para regular de forma óptima el factor equivalente, un factor de escala crítico que determina la proporción de energía consumida del combustible y la batería, de los vehículos eléctricos híbridos enchufables. Por otro lado, en [110] se presenta una red neuronal recurrente profunda (RNN) para la predicción energética de edificios a corto plazo, sirviendo de base para cualquier otro algoritmo de control que utilice estas predicciones en la toma de decisiones para optimizar el consumo eléctrico o reducir el uso de fuentes no renovables. Otro enfoque prometedor es el uso de redes convolucionales (CNN) para extraer características de los datos, en lugar de imágenes, para implementar sistemas de gestión. Por ejemplo, en [111], los autores proponen utilizar una CNN para extraer características espaciales de los datos de electricidad con el fin de predecir cargas futuras. También implementaron una red de Memoria a Corto Plazo (LSTM) para aprender la información temporal y una Unidad Recurrente Cerrada (GRU), realizando varios experimentos con cada modelo y evaluándolos con métricas adicionales de Coeficiente de Variación del RMSE (CV-RMSE).

El Aprendizaje por Refuerzo ha ganado una atención significativa en la gestión de la energía, especialmente aquellos que implican sistemas HVAC (Calefacción, Ventilación y Aire Acondicionado), debido a su capacidad para aprender fácilmente las políticas de control óptimo a través de la interacción con el medio ambiente [112–114]. De hecho, RL es una técnica muy popular en la investigación actual y cada vez hay más trabajos que la utilizan para resolver dinámicamente todo tipo de problemas de control [115, 116]. Generalmente se ha utilizado para resolver problemas de control de vehículos y sistemas basados en movimiento, pero poco a poco se ha ido descubriendo su potencial en otros ámbitos en los que es necesario optimizar las acciones de un sistema inteligente, destacando por ser un método muy dinámico que se adapta fácilmente. En particular, los enfoques utilizados para la gestión de la electricidad aprovechan las capacidades adaptativas de la RL, basadas en las recompensas recibidas por cada acción, para aprender iterativamente las mejores decisiones para cualquier estado posible. Por ejemplo, en [117] se utiliza un algoritmo de RL para la programación en tiempo real de un microgrid, teniendo en cuenta la incertidumbre de la demanda de carga, la energía renovable, y el precio de la electricidad, con el objetivo de encontrar una estrategia de programación óptima para

minimizar el coste de funcionamiento diario del microgrid. Asimismo, en [66] se utiliza del mismo modo para programar de forma inteligente el uso de la energía de una vivienda, pero en este caso se añade además una red neuronal feedforward (FNN) para aumentar el rendimiento de la propuesta al permitir que el agente RL utilice la FNN para predecir los próximos datos de consumo energético. Por otro lado, el aprendizaje por refuerzo también puede utilizarse para la gestión eléctrica de vehículos eléctricos (EV/HEVs) como se ha hecho en otros trabajos mencionados anteriormente. Weihan Li et al. proponen en [118] una estrategia de gestión de la energía basada en Deep RL para un sistema híbrido de baterías en vehículos eléctricos, caracterizando las celdas de la batería eléctrica y térmicamente con el objetivo de minimizar la pérdida de energía y aumentar el nivel de seguridad eléctrica y térmica de todo el sistema. Además, comparan su solución basada en el Deep Q-Learning (utilizando redes neuronales) con un algoritmo clásico de Q-Learning e informan de que en el escenario HEV el método basado en el Deep Q-Learning obtiene mejores resultados.

Algoritmos híbridos

Los enfoques integrados combinan múltiples técnicas algorítmicas para aprovechar las ventajas al combinarlas. Por ejemplo, Fan et al. [119] combinan redes bayesianas y Deep RL para optimizar la fiabilidad del suministro de gas en redes de gasoductos de gas natural, Bisen et al. [120] proponen un modelo híbrido basado en redes neuronales convolucionales y un método de agrupación k-mean modificado para el Edge Computing, y Kuppusamy et al. [121] implementan una solución de sistema de gestión de la energía basada en un algoritmo de árbol de decisión J48 modificado con principios de lógica difusa.

Gap y Solución Propuesta

Los algoritmos de gestión de energía en edificios y casas inteligentes han evolucionado significativamente con los avances en IoT, aprendizaje automático y análisis de datos. Los artículos analizados destacan varias tecnologías, cada una con sus propias ventajas y desventajas, empleadas en algoritmos de gestión energética, incluyendo IoT, aprendizaje automático, análisis de datos y enfoques híbridos. Estos algoritmos contribuyen a optimizar el consumo, reducir costes y mejorar la sostenibilidad en edificios y casas inteligentes.

Entre todos los métodos mencionados, los basados en el aprendizaje por refuerzo son especialmente prometedores y fáciles de aplicar en entornos reales. Como se ha señalado anteriormente, existe una gran variedad de técnicas para abordar problemas de optimización aplicados a sistemas de decisión, pero la mayoría de ellas deben diseñarse con gran precisión y un conocimiento detallado del escenario en el que se van a aplicar, lo que limita su flexibilidad y portabilidad. Sólo las basadas en inteligencia artificial o aprendizaje automático pueden aplicarse en escenarios dinámicos en los que no se conocen con detalle los posibles estados, y entre ellos destaca el aprendizaje por refuerzo por su flexibilidad y capacidad de aprender sobre la marcha sin necesidad de definir previamente el comportamiento del agente.

Por esta razón, nos centramos en usar RL y aplicarlo a un entorno específico para la gestión de dispositivos, CPSs y servicios en una vivienda inteligente aislada con baterías, paneles solares, un generador y sin conexión a la red. Así, el algoritmo detectará dinámicamente, utilizando las recompensas de cada acción realizada, cuáles son las mejores decisiones para cada momento teniendo en cuenta el estado de la producción energética y con el objetivo de optimizar el uso de la electricidad y evitar el uso del generador de gasolina. Uno de los aspectos clave de la solución propuesta es la integración de los CPS en el sistema de decisión, de forma que tanto los servicios virtuales como los dispositivos físicos sean considerados por igual y gestionados por un sistema multiagente de aprendizaje por refuerzo. Además, también es importante considerar un escenario sin conexión a la red eléctrica, ya que ninguna de las propuestas analizadas contemplaba la posibilidad de que un sistema estuviera aislado completamente de la red. En consecuencia, no contemplaban situaciones que implicasen demandas críticas de energía y la posibilidad de un apagón.

4.2.4 Infraestructura para el Computing Continuum

El proceso de decidir qué recursos usar para maximizar la eficiencia de una infraestructura ha sido detallado a lo largo de este documento, citando a diversos autores que han propuesto soluciones variadas. Sin embargo, con la aparición del concepto de Continuum Computing surgen nuevas ideas y desafíos sobre cómo gestionar este tipo de red de manera óptima y eficiente [12].

Varios autores han exploran las posibilidades del Continuum con experimentos y simulaciones para descubrir el posible potencial futuro de este tipo de infraestructura distribuida, como por ejemplo los que describen Daniel Rosendo et al. en [122]. Sin embargo, la gestión del Continuum requiere considerar sus características heterogéneas, algo que no es común en los métodos aplicados en Cloud Computing. En [123] Angelo Marchese et al. presentan en su trabajo cómo es necesario modificar los planificadores de tareas estándar para considerar nuevas características que son muy relevantes en el proceso de decisión del mejor nodo para realizar un trabajo, como el uso del ancho de banda o el coste económico de su utilización. De forma similar, Kaihua Fu et al. explotan el Continuum para desplegar microservicios y para ello desarrollan en [124] un sistema (Nautilus) encargado de desplegar los servicios de forma óptima, asegurando la QoS y considerando siempre el uso de la red, ya que es un factor importante en el continuum computacional debido a su impacto directo en el rendimiento de los servicios.

Aunque el concepto de Continuum Computing es relativamente nuevo, se beneficia del hecho de que el problema de gestionar eficientemente los recursos de computación en una red de nodos es un área de investigación que ha sido ampliamente estudiada por Cloud Computing y recientemente mejorada significativamente por Edge Computing. Por ello, existe un gran número de trabajos que proponen todo tipo de algoritmos, metodologías y frameworks para la gestión de tareas y nodos [23, 24], siendo de especial interés las nuevas aproximaciones basadas en Edge Computing [125], que son fácilmente adaptables al paradigma Continuum Computing. Como se ha mencionado, el Continuum se diferencia del Cloud Computing por tener que hacer frente a nuevas restricciones en el proceso de

decisión de los nodos, por ejemplo, es importante tener en cuenta el consumo de energía [16], la geolocalización del nodo [17], las latencias [15], e incluso si el nodo se está moviendo [18]. Para alcanzar este objetivo se pueden emplear diversos algoritmos; sin embargo, debido a la alta complejidad computacional que suele presentar este problema, a menudo se recurre a técnicas alternativas menos costosas, como por ejemplo el uso de métodos evolutivos.

En cuanto a las tecnologías utilizadas, Kubernetes destaca como el estándar de facto en sistemas multinodo donde se despliegan servicios. Este orquestador de contenedores ha ganado gran popularidad en el ámbito de la computación en la nube y la gestión de microservicios, gracias a su solidez y fiabilidad. Sin embargo, el rendimiento por defecto de K8s es limitado en arquitecturas que no sean clústeres de nodos relativamente homogéneos y, por lo tanto, no es típico para su uso en sistemas distribuidos heterogéneos. Aun así, estas limitaciones son bien conocidas y discutidas por varios trabajos [126][127], que proporcionan metodologías de mejora apoyadas en las herramientas que Kubernetes ofrece para extender la funcionalidad de sus componentes, por ejemplo el Framework de scheduling. Sin embargo, estos trabajos tienden a centrarse principalmente en arquitecturas Cloud computing, como por ejemplo en el articulo de Torgeir Lebesbye et al. [128] donde se identifica el limitado rendimiento del scheduler por defecto de K8s y proponen un servicio (Boreas) que proporciona una mejor forma de seleccionar los mejores nodos para cada uno de los servicios pero dentro de un mismo cluster Cloud. De forma similar, Angelo Marchese et al [129], proponen mejorar el planificador K8s por su limitado rendimiento teniendo en cuenta parámetros relacionados con la red, extendiendo la arquitectura a un entorno de edge computing donde es crítico considerar la calidad de la conexión entre nodos. Además, otros trabajos también abordan esta cuestión, como [130] donde también se introduce el concepto de Continuum Computing, mientras que otros también informan de un creciente interés en añadir nuevas características al proceso de planificación, como la geolocalización de nodos [131], que son muy importantes para el continuum, sirviendo así de referencia para futuras investigaciones. Por otro lado, el articulo de Sebastian Böhm et al.[132] discute de manera general las dificultades de orquestar arquitecturas Cloud-Edge en Kubernetes y propone soluciones, siendo uno de los pocos estudios en discutir explícitamente muchas de las limitaciones críticas del planificador K8s, por ejemplo, una de las más notables es el enfoque de mapeo uno a uno del proceso de asignación de pods a nodos.

Gap y Solución Propuesta

Dado que existen muchos trabajos que tratan sobre la asignación de tareas en nodos, gestión inteligente de contenedores y similares, en este caso nuestro interés no va enfocado en proponer diferentes métodos para la gestión óptima de recursos. El *gap* y objetivo principal de nuestra siguiente propuesta es lo que se ha detectado al realizar la revisión bibliográfica, existe un número limitado de trabajos y herramientas que se centran en mejorar el orquestador más importante del Cloud Computing para que sea eficiente en los nuevos paradigmas de computación. Aunque el área de Edge Computing está siendo abordada en trabajos recientes, el nuevo enfoque del Continuum que combina el modelo Cloud

clásico con Edge y Fog no es todavía un área de estudio muy avanzada, especialmente en la necesidad de incorporar las características diferenciales de cada nodo (potencia de cómputo, aceleradores hardware, capacidades específicas, etc.) en el proceso de gestión de recursos del orquestador. Por ello, se considera relevante tratar de integrar las propiedades individuales de los nodos en la gestión de recursos, en particular los aspectos relacionados con la asignación de cargas de trabajo a los nodos. De este modo, cuando el orquestador deba decidir a qué nodo asignar un trabajo, podrá evaluar las capacidades y características de cada nodo. Así, podrá elegir, por ejemplo, el nodo que resuelva el problema más rápidamente gracias a su mayor potencia de cómputo, aunque presente más latencia, o seleccionar un nodo ubicado en un país específico y/o que cuente con un acelerador de hardware concreto, como una GPU. En definitiva, el orquestador tendrá en cuenta la heterogeneidad de sus nodos a la hora de elegir dónde enviar las tareas de computación, con el fin de optimizar el uso de la infraestructura y mejorar la QoS.

Por lo tanto, nuestra propuesta consiste en diseñar todas las herramientas de soporte y algoritmos para que K8s sea capaz de integrar los parámetros habituales del Continuum en el proceso de distribución de tareas entre sus nodos. De esta forma, el orquestador será capaz de considerar adecuadamente los diferentes aspectos de cada nodo para realizar la mejor distribución tareas usando los módulos desarrollados.

4.2.5 Conclusiones

En conclusión, la arquitectura del Computing Continuum representa un enfoque prometedor al combinar los beneficios del Cloud Computing y el Edge Computing. Sin embargo, debido a su reciente desarrollo, aún enfrenta múltiples desafíos no resueltos y su aplicación se mantiene limitada o con un grado de complejidad reducido. Durante el análisis del estado del arte en Edge Computing y el Continuum, se identificaron diversas limitaciones y gaps de investigación. El primero de ellos fue la necesidad de mejorar la técnica de aprendizaje por refuerzo Q-Learning, destacada por su potencial. Además, se observó la utilidad de aplicar técnicas de Edge Computing en escenarios de Smart House para optimizar el consumo energético, lo que adicionalmente evidenció la necesidad de desarrollar un simulador adecuado para realizar pruebas. Finalmente, se reconoció Kubernetes como una de las herramientas clave para la gestión de contenedores en sistemas basados en microservicios. No obstante, dado que K8s está diseñado principalmente para entornos de Cloud Computing, su adaptación al Computing Continuum plantea un desafío significativo, el cual también ha sido abordado en este trabajo.

4.3 Resultados

En este apartado se presentan los principales resultados obtenidos durante el desarrollo de la tesis. Estos resultados abordan los gaps identificados, cumplen con los objetivos planteados y sirven para implementar las soluciones propuestas. A continuación, se proporciona una visión general de cada resultado, los cuales corresponden a cada una de las publicaciones principales, conformando tres de ellas el compendio.

4.3.1 A multi-layer guided reinforcement learning-based tasks offloading in edge computing

El objetivo principal del primer resultado es abordar el problema de la convergencia en los algoritmos clásicos de Q-Learning. Para ello, se propone un escenario de Edge Computing con una arquitectura piramidal: en la base se encuentran los dispositivos de borde, en el nivel intermedio los dispositivos Fog, y en la parte superior, la nube. La figura 4.4 ilustra la arquitectura propuesta, mostrando los dispositivos y las conexiones que la componen.

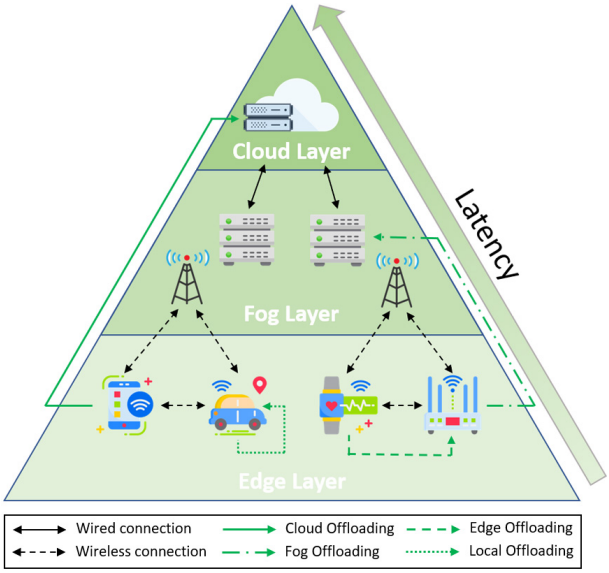

Figure 4.4: Arquitectura Edge Computing.

En este escenario, los dispositivos Edge son responsables de generar tareas de cómputo, las cuales pueden procesar localmente, enviar a otro nodo Edge vecino, a un nodo Fog o a la nube. La elección de dónde procesar las tareas dependerá del nivel de saturación de los nodos, de la red y la deadline de las tareas, lo que constituye el principal problema a optimizar. De hecho, el problema de optimización planteado es muy similar al descrito en la sección 2.1: un conjunto de nodos genera tareas, y el objetivo es asignar dichas tareas de manera que se minimice la suma de las funciones de costo asociadas a cada una, cumpliendo restricciones como evitar la saturación de la RAM o la CPU de los nodos.

Para garantizar el realismo en los experimentos, se ha utilizado el simulador PureEdgeSim para implementar los métodos propuestos. Este simulador permite modelar una red compuesta por un gran número de nodos de diversos tipos y características, distribuidos físicamente en un área de 200x200 metros, como se ilustra en la figura 4.5.

El problema planteado se aborda mediante la aplicación de aprendizaje por refuerzo, con el objetivo de evaluar su idoneidad en este tipo de escenarios. Aunque inicialmente se percibe como un enfoque innovador y prometedor, las pruebas realizadas en el simulador revelan un problema significativo: los nodos tardan demasiado en alcanzar una solución aceptable o, en algunos casos, ni siquiera lo logran. Esto provoca, especialmente en las etapas iniciales de la simulación, un elevado número de tareas sin completar. Este resultado es completamente inaceptable en sistemas reales, donde el impacto negativo sobre

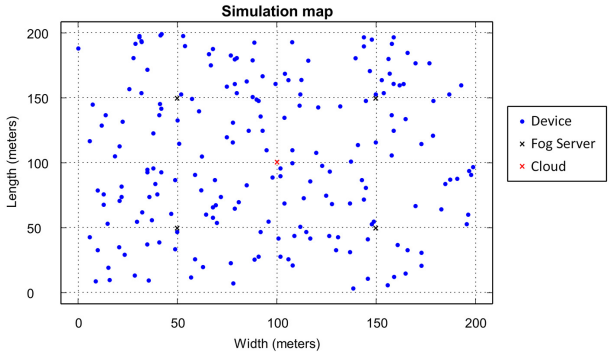

Figure 4.5: Mapa de pruebas.

los usuarios sería excesivo. Tras un análisis detallado del comportamiento de los agentes, se identifica un patrón en las decisiones erróneas. Estas se producen principalmente al inicio, debido al desconocimiento total del impacto de los errores en el proceso de prueba y error característico de las políticas épsilon-greedy, y en situaciones complejas donde muchos nodos están saturados.

El problema radica en que el estado y el entorno del agente son demasiado limitados para permitirle tomar decisiones óptimas, careciendo inicialmente de la capacidad para "intuir" qué opciones podrían ser mejores. Como solución, se propone mejorar el comportamiento de los agentes, inspirándonos en el aprendizaje humano durante la infancia. La propuesta consiste en que, cuando un agente detecte que tiene poco conocimiento del entorno o se enfrente a una situación donde le resulte difícil identificar la mejor decisión, pueda delegar o consultar a un agente de nivel superior con mayor conocimiento. Este agente más experimentado tomaría la decisión con mucha más experiencia e información disponible, sirviendo como modelo para que el primero aprenda a partir de su ejemplo.

La metodología utilizada para diseñar el algoritmo de aprendizaje por refuerzo es la descrita en la sección 2.2, donde cada nodo Edge actúa como un agente, tal como se ilustraba en la figura 4.3. En el algoritmo original, cada dispositivo operaba de forma independiente, basándose en su información local y en los datos globales agregados. Por otro lado, en la propuesta multicapa el agente tiene la posibilidad de delegar la decisión de offloading de tareas a un agente de nivel superior cuando carezca de suficiente información para tomar una decisión óptima. La decisión final de offloading que determina el agente de nivel superior se enviará de vuelta al dispositivo que hizo la consulta para que la ejecute, y tanto el agente local como el de nivel superior aprenderán de la recompensa obtenida tras finalizar la acción. La Figura 4.6 muestra el proceso de la consulta de offloading de la propuesta RL multicapa.

En esta arquitectura mejorada, tanto los dispositivos de edge como los servidores fog ejecutan un mismo algoritmo RL independiente que puede colaborar entre capas. La consulta de offloading permite una transferencia pasiva de conocimientos de los agentes fog a los agentes de edge, especialmente útil cuando un agente de edge comienza sin conocimientos y realiza consultas para aprender de las decisiones del agente de nivel superior. Para ello, hemos ampliado la funcionalidad del algoritmo de offloading RL 1 para incluir la nueva acción de delegar la decisión de offloading en el conjunto de acciones

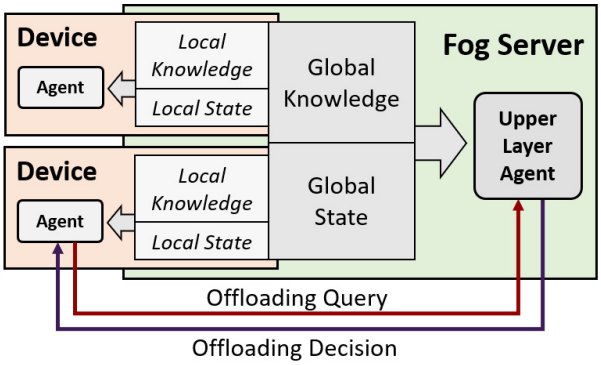

Figure 4.6: Agentes RL multicapa.

factibles de los dispositivos de edge y sección de código para gestionar las consultas en los servidores fog. Los dispositivos de edge y los servidores de fog ejecutan el mismo algoritmo pero con un comportamiento diferente, ya que tienen una visión distinta del entorno.

El agente del dispositivo edge funciona de forma similar al anterior, con la diferencia de que ahora tiene la acción $(A \cup \{4\})$ de realizar la consulta de offloading. Cuando se recibe una recompensa por una acción consultada a un servidor de fog, se considera que tiene un valor más alto ya que se supone que el agente superior tomará mejores decisiones. Además, se penaliza el valor Q de la acción de consultar a un agente superior para disminuir la probabilidad de ser hacerlo de nuevo a la vez que aumenta el conocimiento local del agente. Por otro lado, el agente de la capa superior espera a recibir consultas de offloading de otros agentes y utiliza su propio conocimiento y visión del estado actual, incluidos los usos de CPU en tiempo real de la nube y la capa fog, para tomar una decisión de offloading y enviarla de vuelta. Como resultado, se obtendrá una recompensa de la ejecución de la acción de offloading que mejorará el conocimiento tanto del agente que realizó la petición como del agente de la capa superior.

La solución multicapa propuesta para el problema de offloading basada en ε-greedy Q-Learning se muestra en el Algoritmo 2. Como ya se ha mencionado, todos los dispositivos ejecutan el mismo algoritmo pero con su propia visión del estado y sus propios conocimientos. Esto significa que cada dispositivo gestionará una Q-Table independiente que será entrenada localmente. Además, el agente de la capa superior aprovechará todas las interacciones con los dispositivos para actualizar su estado global.

El algoritmo comienza definiendo los parámetros del proceso de aprendizaje, como el factor de descuento γ de las recompensas, la tasa de aprendizaje α de la función Q, la tasa de exploración ε de la política de control, el parámetro de compensación latencia-energía β, el factor de penalización δ en caso de error de offloading, el multiplicador de recompensa ρ de las acciones de consulta y el factor de penalización ω de utilizar la acción de consulta. En general, todos los dispositivos tendrán los mismos valores de parámetros para mantener la coherencia de los resultados. El algoritmo entra en un bucle que itera cada vez que se recibe una tarea. En el caso de un agente edge, las tareas se reciben a medida que se generan, mientras que un agente fog sólo recibirá tareas cuando se realice una consulta de offloading. Tras eso, se determina el estado actual y el conjunto de

Algorithm 2: ε-greedy Multilayer Q-Learning Algorithm

Parameters: factor de descuento γ, ratio de aprendizaje α, ratio de exploración ε y factor de penalización δ, factor de recompensa de consulta ρ and penalización de consulta ω

```
1  begin
2      for cada paso t do
3          Observar el estado actual s_t
4          Determinar el conjunto de estados factibles A' de A
5          esConsulta ← false
6          e ← número aleatorio de [0,1]
7          if e < ε then
8              a_t ← elegir una acción aleatoriamente de A'
9          else
10             a_t ← arg min_{a∈A'} Q(s_t, a)
11         end
12         if a_t es consultar a un servidor fog then
13             esConsulta ← true
14             Enviar la solicitud de offloading a un servidor fog
15             a_t ← obtener la decisión del servidor fog
16         end
17         Ejecutar o enviar la acción de offloading a_t
18         Esperar a que se complete la tarea
19         Observar el nuevo estado s_{t+1}
20         Calcular la recompensa C_t mediante (4.14)
21         if esConsulta then
22             C_t ← ρ · C_t
23             C_t^q ← ω · t · C_t
24             Actualizar Q(s_t, 4) usando (4.13) con C_t^q
25         end
26         Actualizar Q(s_t, a_t) de acuerdo a (4.13) con C_t
27     end
28 end
```

acciones viables para la tarea T^t y se utiliza para decidir la acción de offloading a tomar siguiendo una política ε-greedy. Si la acción es delegar la decisión en un agente de nivel superior, entonces es necesario enviar la petición y esperar la respuesta. Una vez conocida la acción offloading, se puede ejecutar o, en el caso de un agente fog, enviar al agente que realizó la petición y esperar a que finalice la tarea para calcular la recompensa obtenida. Si la acción offloading es decidida por el agente de nivel superior, el valor de la recompensa se incrementa con el multiplicador ρ y el valor Q de la acción de consulta se disminuye con el factor de penalización ωt. Por último, el valor Q de la acción realizada se actualiza en función de la recompensa calculada mediante la fórmula 4.13.

La figura 4.7 ilustra el comportamiento del algoritmo en tiempo real en un ejemplo con 70 y 170 dispositivos, destacando su desempeño en comparación con otras alternativas.

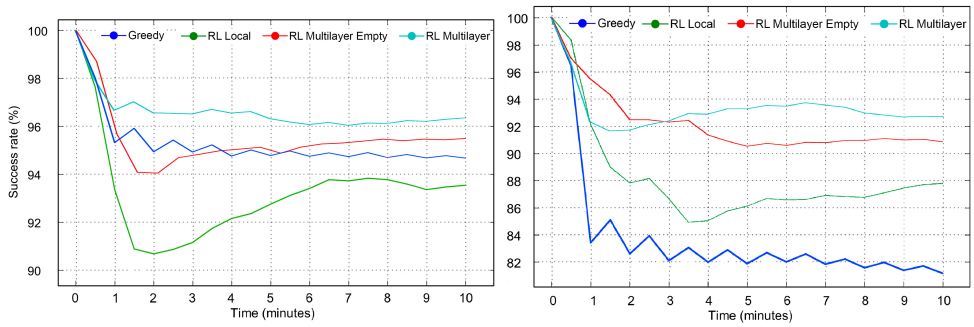

Figure 4.7: Agentes RL multicapa en un ejemplo con 70 y 170 dispositivos.

En ella se presenta la evolución de la tasa de éxito en la ejecución de tareas (cumplen su deadline) a lo largo de 10 minutos. La comparación incluye la propuesta planteada frente a un método Greedy (que asigna cada tarea al nodo con el menor coste), un método basado en aprendizaje por refuerzo sin la acción de consulta, así como la propuesta aplicada con Q-Tables inicializadas en vacío y con las Q-Tables de nodos fog preentrenadas con los valores de una simulación previa. En la figura se observa que el método propuesto es el que logra estabilizarse más rápidamente, gracias a que, durante los primeros instantes de la simulación, realiza un gran número de consultas sobre la acción óptima a tomar. Esto le permite aprender de manera acelerada las mejores decisiones. Además, en ambos ejemplos, la propuesta converge hacia una solución superior al resto de métodos, reduciendo significativamente la tasa de fallos.

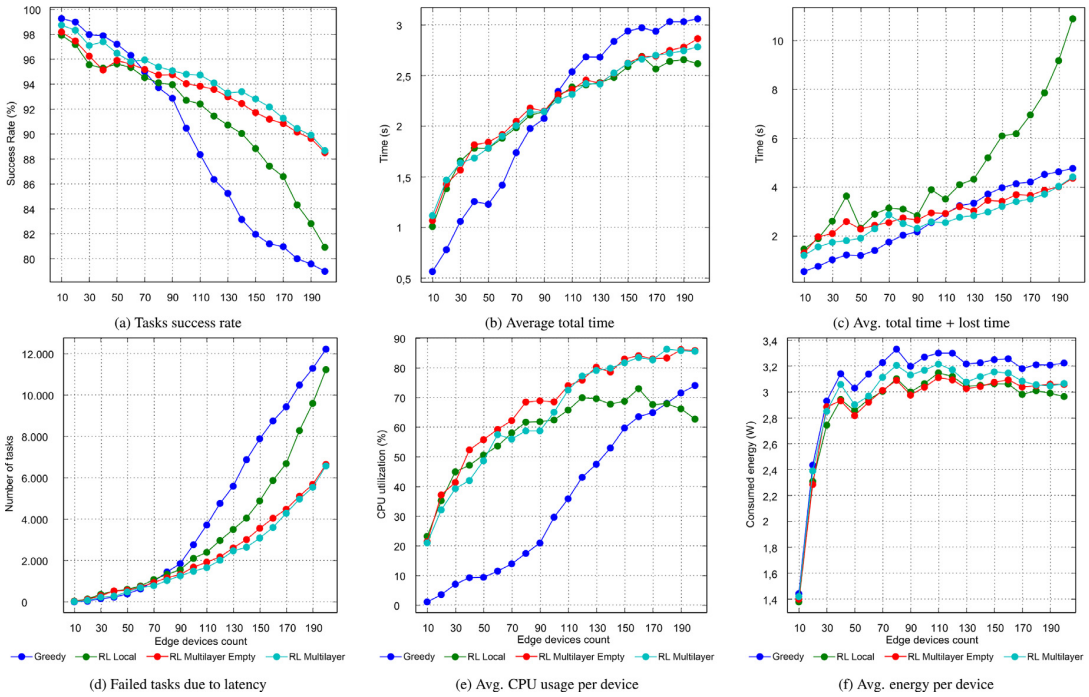

Figure 4.8: Agentes RL multicapa.

Finalmente, en la figura 4.8 se resumen los resultados según seis métricas diferentes

de las simulaciones realizadas sobre escenarios con un numero incrementar de dispositivos (desde 10 hasta 200). Entre los resultados obtenidos, destaca la métrica A, que evidencia cómo la propuesta multicapa logra la mejor tasa de éxito en la ejecución de tareas, abarcando escenarios que van desde baja hasta alta densidad de dispositivos. Por otro lado, la métrica E resulta especialmente interesante, ya que muestra que los métodos basados en aprendizaje por refuerzo distribuyen las tareas de manera eficiente entre los nodos, evitando la saturación de las CPUs y maximizando así la capacidad de procesamiento distribuido en este tipo de redes.

4.3.2 Sim-PowerCS: An extensible and simplified open-source energy simulator

El segundo trabajo surge de la necesidad de diseñar un entorno de pruebas que permita simular dispositivos inteligentes integrados en sistemas de Smart House. Este entorno está destinado a medir tanto el consumo energético como el tiempo de funcionamiento de los servicios instanciados.

Los servicios simulados pueden incluir componentes de software, como servidores web o bases de datos, así como dispositivos físicos, como bombillas, frigoríficos o sistemas de ventilación. El objetivo del simulador no se limita a proporcionar un entorno de pruebas realista, sino que también busca facilitar la implementación de algoritmos de control, tanto para la gestión de los servicios desplegados como para la optimización del consumo energético.

Por ello, en esta publicación nos enfocamos en el desarrollo de un software de simulación open-source que sea lo más sencillo posible. Esto responde a la necesidad identificada en el estudio del estado del arte, donde se observó que las alternativas existentes eran excesivamente complejas. Además, el software debe permitir la implementación de un plano de control para los dispositivos inteligentes y otro orientado a la gestión energética, incluyendo el uso de fuentes de energía renovables y la compra-venta de energía. La figura 4.9 resume los componentes principales del simulador Sim-PowerCS.

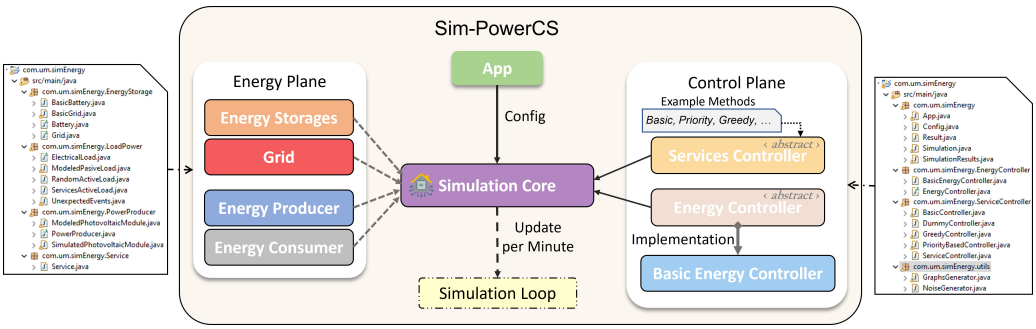

Figure 4.9: Componentes principales del simulador Sim-PowerCS.

Sim-PowerCS está compuesto por dos planos principales: el plano energético y el plano de control. En el plano energético, el sistema agrupa diversas fuentes, almacenes y consumidores de energía. Por su parte, el plano de control ofrece dos puntos clave para la

personalización por parte de los desarrolladores: la gestión de las decisiones energéticas y el control del estado de los consumidores de energía.

Antes de comenzar una simulación, es fundamental configurar los elementos que utilizará el sistema. Esto incluye definir las fuentes de energía disponibles (como solar y eólica), la información relacionada con la red eléctrica, la capacidad de las baterías, la lista de consumidores de energía (dispositivos, servicios y cargas pasivas) y los algoritmos de control para la gestión de la energía y los servicios.

Posteriormente, Sim-PowerCS comenzará a funcionar durante el número de días especificados por el usuario, inicializando todos los componentes mencionados e iterando a través del bucle principal de simulación por cada minutos. Por lo tanto, la unidad de tiempo indivisible de lasimulación es un minuto, y todos las acciones y resultados se determinaran minuto a minuto.

A continuación, el bucle de la simulación se repetirá tantas veces como sea necesario para completar el número de minutos requerido según el conjunto de días a simular. La figura 4.10 muestra el procedimiento general para cada iteración, incluyendo un ejemplo básico de algoritmos de gestión de energía y control de servicios.

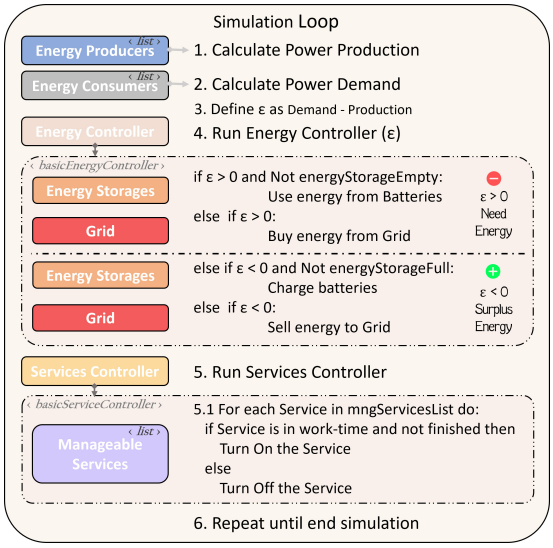

Figure 4.10: Simulation Loop.

El primer paso en cada iteración es determinar para ese instante exacto de tiempo la producción total de energía (energía disponible para su uso) basándose en la lista de fuentes de energía disponibles, como la energía solar en ese momento del día y la energía eólica actual. A continuación, se calcula la demanda total de energía de la lista de consumidores, que incluye todo tipo de servicios y dispositivos, como sistemas ciberfísicos, climatizadores, luces, sistemas de videovigilancia, etc.

Por último, se calcula ε como la energía necesaria en ese ciclo (*demanda menos producción*) y se llama al controlador de energía para que decida qué acciones tomar con respecto a la gestión de la energía. El controlador de energía es uno de los dos principales puntos de extensión que Sim-PowerCS proporciona a los desarrolladores, en este caso se muestra el comportamiento de una versión básica que podría mejorarse fácilmente para

incluir métodos de aprendizaje automático para tomar decisiones basadas en patrones.

El algoritmo de ejemplo determina primero si se necesita energía en esa iteración porque la demanda supera a la producción o si, por el contrario, tenemos un excedente de producción. En el caso de necesitar energía, comprueba si las baterías no están descargadas para utilizarlas como fuente de energía, en caso contrario compra la energía a la red (incurriendo en un coste que se acumula). Por otro lado, si tiene un excedente de producción primero comprueba si las baterías no están cargadas para utilizar la energía sobrante para cargarlas, en caso contrario vende la energía sobrante a la red (incurriendo en un beneficio monetario que se registra). Como puede verse, el algoritmo energético del ejemplo es relativamente sencillo pero eficaz en muchos escenarios, aunque puede mejorarse fácilmente para tener en cuenta patrones de generación de energía, consumo energético o precios de la energía en la red.

Tras determinar las acciones energéticas y ejecutarlas, se llama al controlador de los servicios (consumidores de energía) para que decida en función del estado energético actual qué hacer con cada servicio (encenderlos o apagarlos). Sim-PowerCS dispone de tres algoritmos de control de servicios, el más básico de ellos es el que se muestra en la figura como ejemplo inicial. Este algoritmo se centra exclusivamente en los servicios gestionables, aunque existen otros servicios que también generan consumo energético, han sido definidos como no controlables, por lo que no están sujetos al control del algoritmo. El comportamiento de este método es muy simple y sirve como referencia del peor caso de rendimiento con respecto a cualquier posible implementación más inteligente. Básicamente, el controlador se limita a determinar si un servicio se encuentra dentro de la franja horaria en la que puede encenderse y aún no ha superado su tiempo máximo de encendido para mantenerlo encendido o, de lo contrario, apagar el servicio.

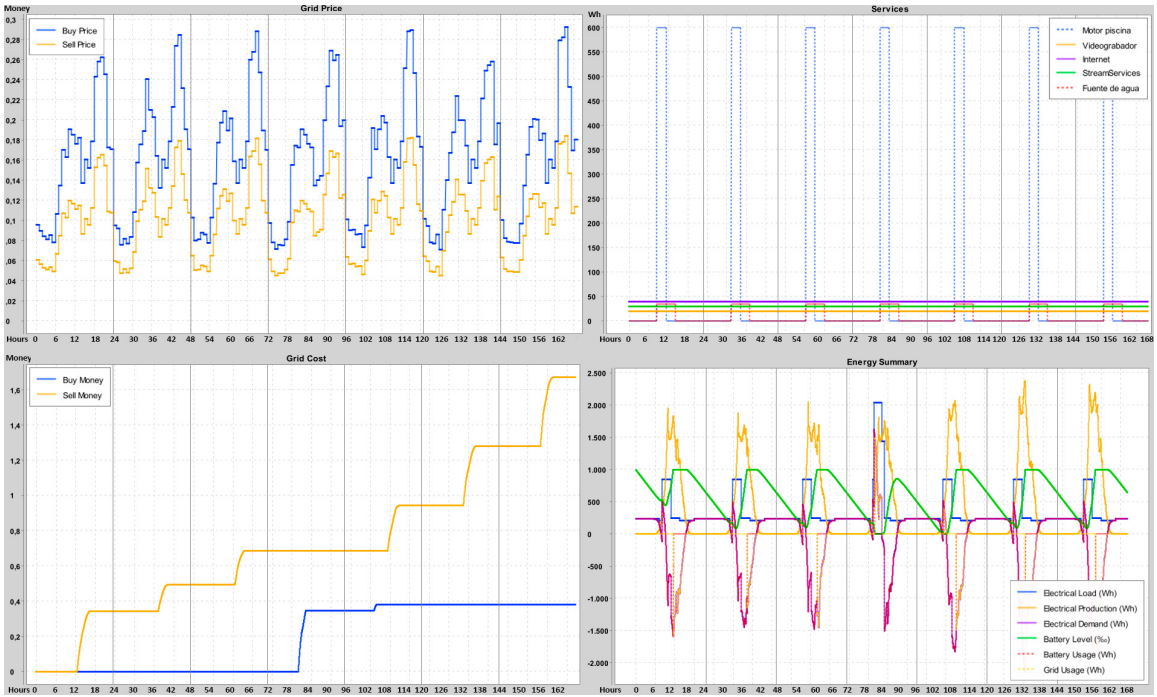

Figure 4.11: Resumen de simulación.

Una vez completados todos los pasos del bucle de simulación, toda la información de esa iteración se almacena en el registro de datos y se inicia la iteración del minuto siguiente. Finalmente, tras realizar todos los minutos de la simulación se finaliza el programa y se genera una serie de ficheros e imágenes que detallan los resultados de la simulación. La figura 4.11 presenta un ejemplo de simulación de siete días, donde se observan los precios variables de compra y venta de energía en la red, el consumo energético de los servicios desplegados, los costos asociados a la compra-venta de energía, y un resumen de cada parámetro energético de la simulación. Este resumen incluye la generación solar, el consumo total, el uso y nivel de las baterías, así como el uso de la red eléctrica.

Un aspecto destacado del simulador es su capacidad para generar datos de manera procedural utilizando modelos estadísticos predefinidos. Esto permite crear escenarios de prueba con datos realistas y similares a los de un entorno implementado en el mundo real, al tiempo que ofrece la posibilidad de introducir variaciones procedurales en dichos datos, logrando así simulaciones más dinámicas y versátiles. La figura 4.12 ilustra este comportamiento. En la parte izquierda, se presenta el modelo base, donde la producción de energía solar se define por hora, acompañada de un intervalo de confianza. En contraste, la parte derecha muestra una generación procedural de datos para un día específico, en la que se introducen pequeñas fluctuaciones usando ruido Perlin, creando un comportamiento más realista y variable entre días.

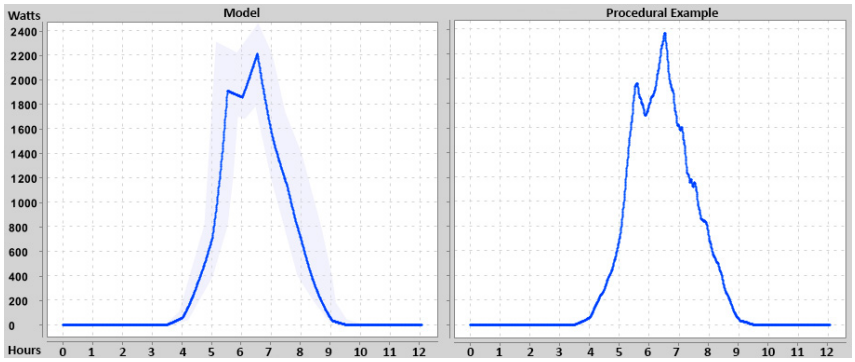

Figure 4.12: Solar energy production model with confidence interval (left) and procedural energy generation example (right).

En conclusión, los simuladores desempeñan un papel crucial en la comprensión y optimización del uso energético en hogares y edificios inteligentes. En este contexto, presentamos Sim-PowerCS, una herramienta de software flexible y extensible diseñada para simular sistemas energéticos. Su propósito es facilitar la prueba y optimización de diversos algoritmos de gestión de energía y mecanismos de control de servicios inteligentes en un entorno controlado. Sim-PowerCS incluye dos puntos clave de control: la gestión energética y el control de servicios. Además, permite integrar múltiples fuentes de energía, sistemas de almacenamiento y consumidores energéticos, proporcionando un marco versátil para la simulación y análisis. Este simulador resulta fundamental para nuestro próximo trabajo, ya que proporciona un entorno de pruebas ideal para desarrollar y evaluar métodos más avanzados de control de servicios en hogares inteligentes con limitaciones energéticas significativas, como la ausencia de conexión a la red eléctrica.

4.3.3 An adaptive energy orchestrator for cyberphysical systems using multiagent reinforcement learning

A partir del simulador desarrollado en la publicación anterior, se pueden probar diversas propuestas de algoritmos para gestionar los servicios de una casa inteligente. En particular, en este tercer trabajo destaca el interés por implementar técnicas de aprendizaje por refuerzo multiagente, dado que son métodos altamente dinámicos y autónomos, capaces de adaptarse a los cambios en los sistemas que controlan. El objetivo principal de esta tercera publicación, no incluida en el compendio, es evaluar el desempeño del aprendizaje por refuerzo en un entorno de Smart House, donde cada dispositivo actúa como un agente individual que toma decisiones basadas en el estado general de la vivienda.

Para destacar las capacidades dinámicas y el rendimiento superior que un algoritmo de aprendizaje por refuerzo puede ofrecer en comparación con otras alternativas, se ha diseñado un caso de uso donde la gestión eficiente de la energía es crucial. En este escenario, la vivienda no cuenta con conexión a la red eléctrica, dependiendo exclusivamente de la energía generada por los paneles solares, la almacenada en baterías, y, de manera excepcional, de un generador de emergencia cuyo uso está severamente limitado. La figura 4.13 ilustra los sistemas de energía disponibles en la casa inteligente.

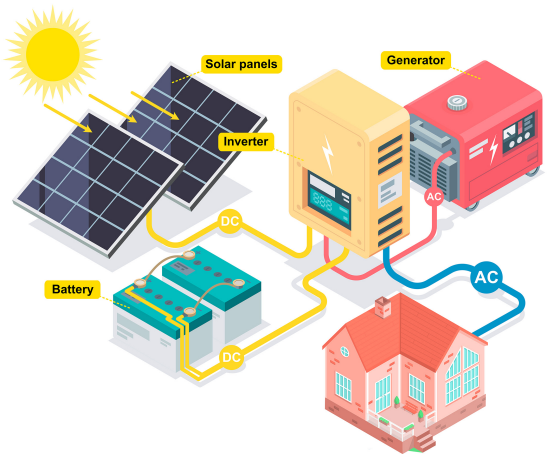

Figure 4.13: Casa Inteligente y sus fuentes de energía

La principal fuente de energía del sistema es el conjunto de paneles solares, que genera electricidad cuando hay suficiente luz solar y almacena el excedente en un banco de baterías. Cuando la producción solar disminuye, el inversor utiliza la energía almacenada en las baterías para satisfacer las demandas del sistema. Sin embargo, esto puede llevar a una descarga completa de las baterías, lo que no solo reduce su vida útil, sino que también provoca un apagado abrupto de todos los servicios. Para prevenir una situación de energía insuficiente, se cuenta con un generador que puede activarse de manera excepcional cuando el nivel de las baterías es crítico, proporcionando energía temporal hasta que estas se recarguen adecuadamente.

Por ello, la gestión eficiente de la energía resulta fundamental para evitar el agotamiento de las baterías durante la noche y maximizar el aprovechamiento de los picos

de generación solar. Para abordar este desafío, se ha usado el plano de control del simulador para regular el funcionamiento de los servicios, ajustando su estado según los niveles de producción y consumo energético. Este plano actúa como un orquestador central del sistema, enfocado en optimizar la gestión de la energía total.

Los servicios gestionados por el orquestador son tanto virtuales (*vídeo en streaming*) como físicos (*nevera, bomba de piscina*), por lo que se modelan como servicios ciberfísicos con un consumo definido por hora (*Wh*) y una regla de ejecución (*horario de encendido/apagado y tiempo máximo de ejecución*).

Además, los servicios CPS pueden separarse en dos grupos dependiendo de si el orquestador los gestiona de forma inteligente. Los servicios No Gestionables son aquellos que el orquestador no puede controlar dinámicamente su funcionamiento, por lo que se limita a controlar su encendido y apagado de acuerdo a su horario de ejecución diario definido (*regla de ejecución*). Por el contrario, los servicios gestionables son aquellos que el orquestador puede gestionar de forma inteligente, apagándolos cuando sea necesario (*baja energía o carga inesperada*) y desplazando su ejecución (*pico de generación solar*) en función del estado energético actual, la prioridad del servicio, el horario de ejecución del servicio y el tiempo máximo permitido para la ejecución del servicio.

Además, el orquestador también tiene en cuenta la existencia de imprevistos que complican el suministro energético y obligan a tomar medidas para reducir o desplazar el consumo de energía. Por ejemplo, eventos inesperados pueden ser la reducción de la producción solar en días nublados, la conexión puntual de un dispositivo de alta demanda energética a la red eléctrica del sistema o el vaciado completo de las baterías. La figura 4.14 resume los componentes del sistema propuesto y algunos ejemplos de servicios.

El orquestador gestiona tanto servicios virtuales (como streaming de video) como físicos (como el frigorífico o el motor de una piscina). Estos servicios se modelan como sistemas ciberfísicos (CPS) con un consumo energético definido por hora (Wh) y reglas específicas de operación (horarios de encendido/apagado y tiempo máximo de ejecución permitido).

Los servicios CPS se dividen en dos categorías según el nivel de control ejercido por el orquestador:

- **Servicios no gestionables:** Son aquellos cuyo funcionamiento no puede ser ajustado dinámicamente por el orquestador. Este se limita a encenderlos y apagarlos según su horario diario predefinido, siguiendo estrictamente sus reglas de operación.

- **Servicios gestionables:** Son aquellos que el orquestador controla de forma inteligente. Puede apagar estos servicios en situaciones críticas, como baja energía disponible o cargas inesperadas, y ajustar su ejecución desplazándola a momentos más óptimos, como durante picos de generación solar. Las decisiones se toman en función del estado energético del sistema, la prioridad del servicio, su horario de operación y el tiempo máximo permitido para su ejecución.

El orquestador también tiene en cuenta imprevistos que afectan el suministro energético, como días nublados que reducen la generación solar, la conexión inesperada de dispositivos con alta demanda energética o el vaciado completo de las baterías. Ante estas

situaciones, implementa estrategias para reducir o desplazar el consumo energético según sea necesario.

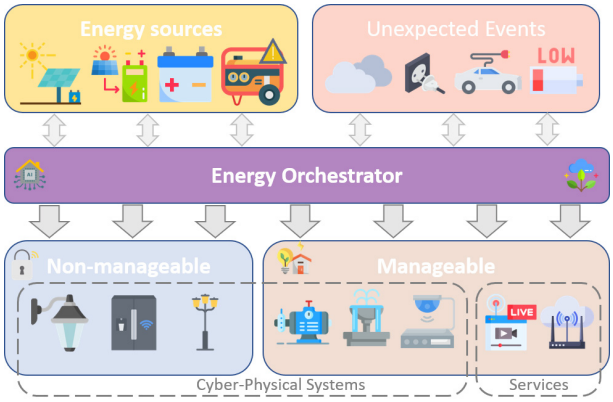

Figure 4.14: Sistema propuesto

Para llevar a cabo nuestra propuesta, utilizamos el simulador Sim-PowerCS, donde recreamos el escenario descrito anteriormente. En este entorno, implementamos en el plano de control el método basado en aprendizaje por refuerzo (RL) y otras alternativas, con el objetivo de realizar pruebas comparativas. La figura 4.15 proporciona una visión general de la arquitectura de los algoritmos de control propuestos. El orquestador de energía recibe cada minuto información sobre el estado de generación y consumo de energía de los servicios proporcionados por el simulador o los sensores del mundo real. En cada actualización de información el orquestador genera un conjunto de eventos que envía al plano de control donde lo recibe la interfaz principal del controlador de servicios.

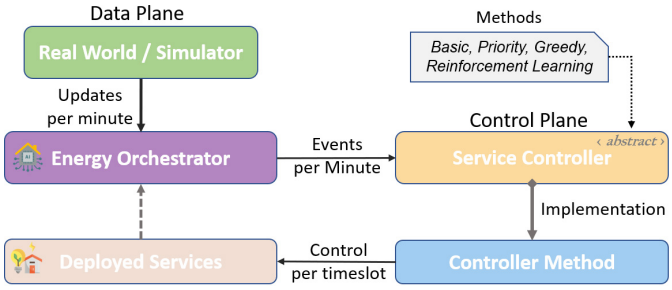

Figure 4.15: Arquitectura del orquestador

El controlador de servicios recopilará información sobre eventos y tomará decisiones de control sobre los servicios en cada franja horaria de acuerdo con el comportamiento de la implementación específica del controlador. Las franjas horarias son intervalos de tiempo definidos por el usuario, por ejemplo 60 minutos, en los que el algoritmo de control sólo recopilará información para la toma de decisiones al principio de la siguiente franja horaria. De esta manera, el agente RL realizará acciones en el primer minuto de su timeslot de control y calculará la recompensa en el ultimo minuto, dando tiempo a que la acciones tengan repercusión en el sistema. La figura 4.16 ilustra el comportamiento de control descrito.

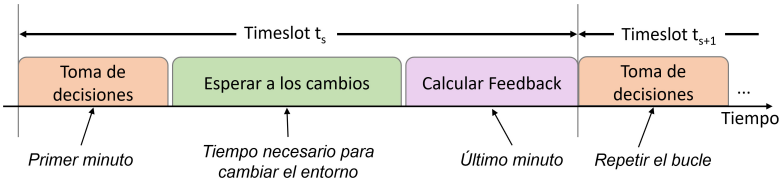

Figure 4.16: Estructura temporal del proceso de control RL

Para evaluar el rendimiento del método basado en aprendizaje por refuerzo, se diseñan dos soluciones adicionales que emplean enfoques más tradicionales, los cuales pueden además ser alternativas válidas y ofrecer un buen desempeño en determinados escenarios.

Como primera aproximación al sistema de control de servicios, se desarrolla un algoritmo relativamente simple que, al detectar un nivel crítico de energía en una franja horaria específica, ordena los servicios por prioridad y desactiva aquellos de menor importancia. Cuando la producción de energía aumenta, el algoritmo reactiva los servicios siguiendo nuevamente el orden de prioridad. Este método, aunque es sencillo, resulta adecuado para muchos escenarios y proporciona un rendimiento aceptable, aun teniendo como principal desventaja una limitada flexibilidad.

Para la segunda alternativa, se implementa un método greedy que emplea las mismas heurísticas que el algoritmo de aprendizaje por refuerzo, pero con un menor coste computacional. Este enfoque determina un valor que representa el estado energético global del sistema y lo compara con el valor heurístico de cada servicio; si el servicio tiene un valor inferior, se desactiva. Este método logra un rendimiento aceptable y ofrece cierta flexibilidad ante los cambios, aunque carece de capacidad de anticipación.

Finalmente, la solución basada en aprendizaje por refuerzo se diseña como un sistema multiagente, donde cada servicio actúa como un agente independiente que toma decisiones según el estado del sistema. El objetivo es que, de manera conjunta, todos los agentes optimicen una única función de recompensa agregada, con la diferencia respecto a los métodos anteriores de que esta solución permite a los agentes aprender patrones del entorno y adaptarse rápidamente a cambios abruptos o situaciones inesperadas. La recompensa obtenida tras la ejecución de una acción se define como una función a trozos de dos elementos que depende de la acción realizada, de los parámetros del servicio y también del nuevo estado del entorno. De este modo, la recompensa del servicio "**S**" en el momento "**t**" se formula como la siguiente suma ponderada entre la prioridad del servicio (S_p), el tiempo de funcionamiento actual (S_{rt}), el tiempo de espera (S_{wt}), el nivel de la batería a la inversa (B_t), el nivel inverso mínimo diario de la batería (B_t^m) y el uso del generador (G_t):

$$C_t(S) = \begin{cases} \beta_2 S_{wt} - \beta_4 B_t^m & a = 0 \\ \beta_0 S_p + \beta_1 S_{rt} + \beta_2 S_{wt} + \beta_3 B_t + \beta_4 B_t^m + \beta_5 G_t & a = 1 \end{cases} \tag{4.15}$$

Donde cada β_x es la constante de trade-off para cada parámetro y la actualizacion de los valores Q de cada accion se hace siguiendo la formula de Q-Learning (eq. 4.13) explicada con anterioridad.

Utilizando el simulador, es posible evaluar en un mismo entorno cada uno de los

métodos con el fin de identificar su comportamiento y rendimiento en tiempo real. Para ello, se configura Sim-PowerCS con un caso de uso basado en una vivienda equipada con servicios inteligentes, la cual no está conectada a la red eléctrica y su suministro de energía proviene de un sistema compuesto por paneles solares con picos de generación de 2.2 kW, un conjunto de baterías con una capacidad de 7 kWh y un generador de gasolina que solo debe activarse en situaciones estrictamente necesarias.

La simulación abarca un periodo de 25 días virtuales. Durante los primeros 4 días, el clima está nublado, lo que reduce la generación solar al 80%. Además, se registran consumos imprevistos de entre 1.2 y 1.8 kWh en los días 4, 13, 17 y 22, específicamente entre las 10 a.m. y las 12 p.m. El sistema también implementa los servicios detallados en la tabla 4.1, donde se especifican su prioridad, consumo energético y las reglas de activación correspondientes.

Table 4.1: Servicios CPS desplegados

Servicio	**Smart**	**Prioridad**	**Carga**	**Regla**
Luces vallas	No	-	15 W	8pm-8am
Luces fachada	No	-	10 W	8pm-8am
Frigorífico	No	-	120 W	All time
CCTV DVR	Sí	10	20 W	All time
Internet	Sí	8	40 W	All time
Motor Piscina	Sí	4	600 W	9am-5pm (max 3h)
Streaming Svcs.	Sí	2	30 W	All time
Fuente	Sí	1	35 W	9am-3pm

Las pruebas realizadas incluyen cuatro algoritmos de control para el orquestador, el primero llamado **Basic** (*BSC*) no realiza ningún control inteligente de los servicios, sólo enciende y apaga los servicios según sus reglas de ejecución. El segundo método, el **Priority-based** (*PB*), ordena los servicios según su prioridad y si detecta que hay un nivel crítico de energía, apaga los servicios menos prioritarios hasta que aumente la producción de energía. El tercero es un método **Greedy** (*GDY*) que se basa en comparar el valor de una heurística dinámica que depende del estado del entorno con el valor de la heurística de cada servicio, en caso de que el servicio tenga un valor inferior se apaga. Por último, el cuarto es el algoritmo multiagente **reinforcement learning** (*RL*) explicado anteriormente que decide dinámicamente si apagar o encender servicios en cada franja horaria (sujeto a reglas de ejecución). Debido a la naturaleza de los algoritmos RL, el agente necesita aprender gradualmente las mejores acciones a tomar en función de las recompensas. Por este motivo, el algoritmo requiere repetir la simulación varias veces (guardando las tablas Q) para converger progresivamente a la mejor política. En nuestro escenario fue necesario repetir la simulación cinco veces para obtener la mejor solución la cual indicamos en los resultados como *RL5*.

Table 4.2: Rendimiento de los métodos

	BSC	**PB**	**GDY**	**RL**	**RL5**
CCTV DVR	100%	93.7%	100%	99.8%	100%
Internet	100%	89.3%	100%	100%	100%
Pool Pump	100%	90.7%	72.0%	88%	70.7%
Streaming Services	100%	74%	100%	100%	99.2%
Fountain	100%	11.3%	49.3%	98.7%	99.3%
Generador (Watts)	*20180*	*5934*	*7502*	*14680*	*7335*

Como se muestra en la tabla 4.2, el método Básico representa el peor rendimiento posible ya que mantiene todos los servicios encendidos, por lo que el resultado de consumo de generador de este método sirve como límite superior de los métodos a comparar.

La segunda propuesta, el método Priority, proporciona el mejor resultado con respecto al uso del generador a costa de reducir significativamente el tiempo de ejecución de los servicios de menor prioridad. Este método podría considerarse útil en escenarios sencillos en los que no se requiere dinamismo y sólo es relevante la gestión basada en prioridades, sin embargo, en escenarios como el que proponemos, es necesaria una mayor flexibilidad y adaptabilidad en tiempo real de los algoritmos para maximizar el tiempo de funcionamiento de todos los servicios al tiempo que se reduce el uso del generador.

Por otro lado, el método Greedy ofrece un rendimiento final superior en cuanto al consumo del generador y al mantener los servicios encendidos en la medida de lo posible. Esto se debe al uso de una heurística dinámica que cambia en función del estado energético, apagando todos los servicios que tengan un valor más bajo. Así, cuando la producción solar es baja o la batería alcanza niveles críticos, se apagan preferentemente los servicios de menor prioridad y mayor consumo (por ejemplo, el motor de la piscina). La desventaja de este método es su escaso dinamismo, ya que la compensación entre prioridad, consumo, tiempo de ejecución y estado energético se define manualmente y no se adapta fácilmente a los cambios del sistema.

En cambio, el método basado en el aprendizaje por refuerzo es completamente flexible, ya que se adapta continuamente a los cambios y no necesita definir ninguna heurística de antemano. El inconveniente es la necesidad de repetir varias veces su ejecución (episodios) hasta que el algoritmo converge a la solución óptima. Por ello, en la tabla se muestran para comparar el resultado de la simulación con el método RL inicial (sin conocimiento) y el resultado tras ejecutar la simulación 5 veces manteniendo las tablas Q de cada agente.

Como se puede observar, el método sin conocimiento inicial apenas apaga los servicios ya que aún no conoce las acciones más adecuadas a realizar, lo que conlleva un elevado consumo del generador. Por el contrario, el método entrenado consigue la mejor solución ya que su consumo de generador es el menor manteniendo los servicios encendidos en la medida de lo posible. El método propuesto no sólo se adapta en tiempo real a los imprevistos, sino que también es capaz de desplazar la ejecución de los servicios programados a las horas de mayor producción solar o a las horas que estima más adecuadas.

Para ilustrar visualmente la comparación de los métodos según el uso del generador,

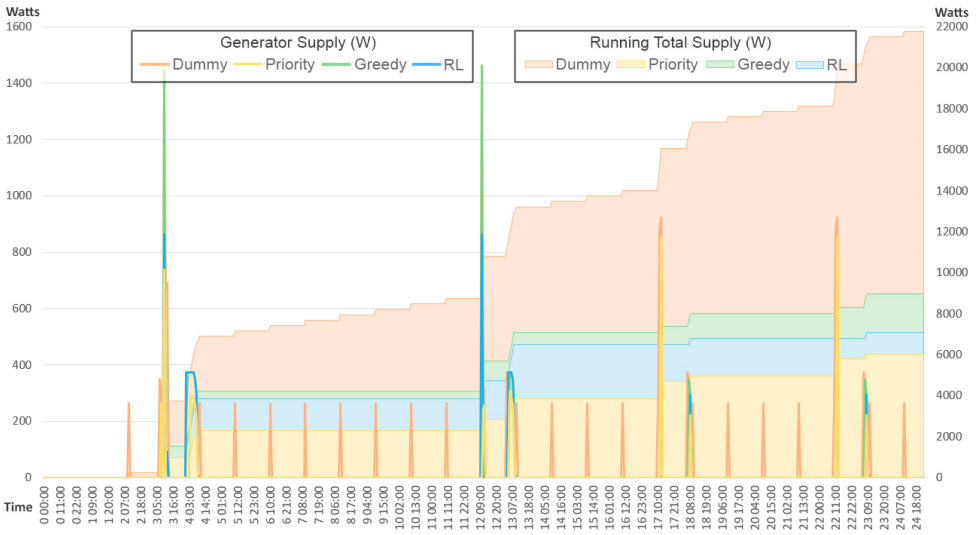

Figure 4.17: Comparación del uso del generador de cada método

la Figura 4.17 presenta el uso puntual y acumulado del generador, medido en vatios, a lo largo de la simulación. El método Básico emplea el generador de manera continua cada noche cuando las baterías se agotan. Por otro lado, los métodos Prioritario y Greedy limitan el uso del generador a situaciones donde se producen consumos inesperados. De manera similar, el método RL recurre al generador únicamente cuando la producción de energía es insuficiente debido a un consumo inesperado. Además, este método logra reducir aún más el uso del generador al desplazar o limitar los tiempos de funcionamiento de servicios con alto consumo energético.

La Figura 4.18 muestra la comparación entre los métodos en distintos momentos de la simulación. Esta figura detalla el funcionamiento de los métodos en un periodo de tres días de simulación, permitiendo observar con mayor precisión el comportamiento de cada algoritmo en diversas situaciones. Cada gráfico en la figura representa el consumo de los servicios activos en cada franja horaria (timeslot), junto con el nivel de batería (en milésimas) y el consumo energético (en vatios).

La primera gráfica ejemplifica el comportamiento por defecto del sistema, en el que los servicios se encienden según sus reglas de ejecución, provocando que el sistema se quede sin energía durante las horas punta tanto el segundo como el tercer día. Además, el tercer día se produce un consumo inesperado de 1,2 kW que agota las baterías. Por otro lado, la segunda gráfica ilustra el comportamiento anteriormente mencionado del método basado en prioridades, cuando se producen situaciones energéticas críticas, los servicios de menor prioridad en cada franja horaria se apagan progresivamente en orden hasta que mejora la producción de energía y, a continuación, se vuelven a encender gradualmente. Asimismo, la tercera gráfica muestra cómo el método Greedy detecta el estado de batería baja del segundo día para no encender ningún servicio de baja prioridad. Además, durante el consumo inesperado del tercer día también apaga los servicios de baja prioridad para ahorrar energía. En cambio, la última gráfica evidencia cómo el algoritmo RL es capaz de adaptar dinámicamente la ejecución de los servicios para reducir el uso del generador

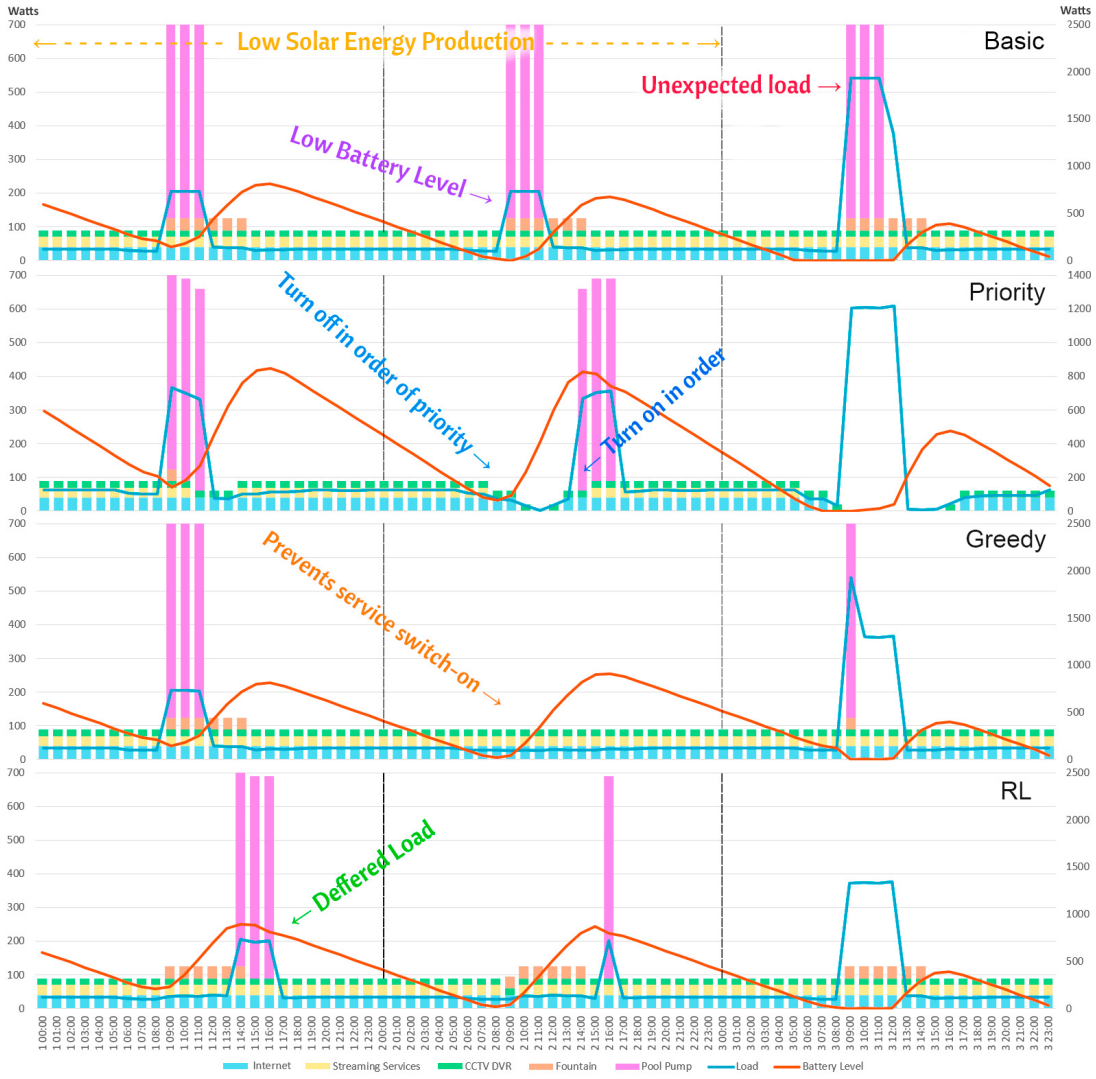

Figure 4.18: Simulación de tres días de cada método

maximizando el tiempo de funcionamiento. Uno de los resultados más importantes del método propuesto es el desplazamiento del servicio del motor de la piscina al periodo de mayor nivel de batería y producción solar. Asimismo, el encendido parcial del servicio del motor en caso de detectar un bajo nivel de producción de energía o el apagado total en caso de un alto consumo inesperado demuestran el adecuado comportamiento adaptativo del algoritmo.

En resumen, la implementación del aprendizaje por refuerzo multiagente en la optimización energética de hogares o edificios inteligentes ofrece un potencial significativo. Este enfoque basado en RL destaca por su capacidad adaptativa, permitiendo ajustar el consumo energético de distintos servicios y aplicar ciclos de funcionamiento parciales o intermitentes en situaciones críticas. Todo esto se logra de forma autónoma y sin necesidad de establecer reglas predefinidas.

4.3.4 Adapting Containerized Workloads for the Continuum Computing

Por último, el cuarto trabajo realizado se centra en una publicación con un enfoque más práctico. En esta, se identifica a Kubernetes como la plataforma más utilizada para Cloud Computing y la gestión de microservicios, y se analiza cómo adaptarla para gestionar arquitecturas basadas en los principios del Computing Continuum. Debido a que K8s está diseñado principalmente para modelos de computación en la nube, presenta ciertas limitaciones al integrar los requisitos del Continuum. Algunas de estas limitaciones pueden resolverse utilizando las herramientas que ofrece el propio framework, mientras que otras requieren modificar o reescribir componentes internos de Kubernetes.

La arquitectura utilizada es una pirámide Edge-Fog-Cloud que forma un continuum. Para lograrlo, cada capa está compuesta por clústeres de K8s interconectados mediante la herramienta Liqo.io. De este modo, la arquitectura permite ofrecer una serie de servicios que se desplegarán en los nodos y capas más adecuados según distintos criterios, como la latencia máxima permitida, restricciones de privacidad específicas de cada país o políticas de antiafinidad. La figura 4.19 ilustra los competentes y niveles de la arquitectura usada.

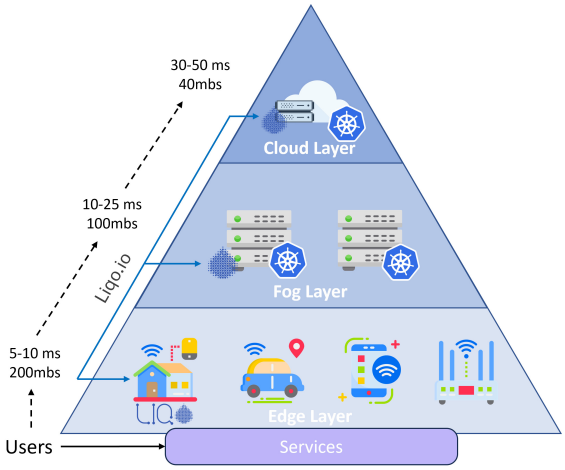

Figure 4.19: Arquitectura del continuum K8s

El primer nivel (Edge Layer) es el más cercano a los usuarios y consiste en un conjunto de dispositivos de baja potencia y capacidad de computo que están lo suficientemente cerca de los usuarios como para proporcionar acceso a los servicios con baja latencia y gran ancho de banda. El segundo nivel (Fog Layer) es un punto intermedio de la arquitectura en el que se despliegan servidores más potentes que se encargan de todas las tareas pesadas que los dispositivos de borde son incapaces de procesar. Estos dispositivos suelen estar relativamente cerca (misma región o estado) de los dispositivos de borde, por lo que su latencia no es muy alta y tienen un ancho de banda aceptable. Por último, el tercer nivel representa la Nube, que consiste en grandes centros de datos con gran capacidad de computación. La desventaja de la Nube es que, debido al reducido número de nodos y a la gran distancia que la separa de los usuarios finales, su latencia suele ser muy alta y su ancho de banda muy bajo.

Para verificar la arquitectura desarrollada, se diseña un conjunto de aplicaciones o servicios que se desplegarán en los distintos nodos. Cada aplicación presenta requisitos específicos de CPU y RAM, similares a un despliegue estándar en Kubernetes. Además, se pueden definir condiciones adicionales, como afinidad, antiafinidad, latencia mínima y deseada, restricciones de ubicación, o requisitos específicos de hardware (por ejemplo, GPU, RAM con ECC, discos NVMe, instrucciones AES-NI, entre otros).

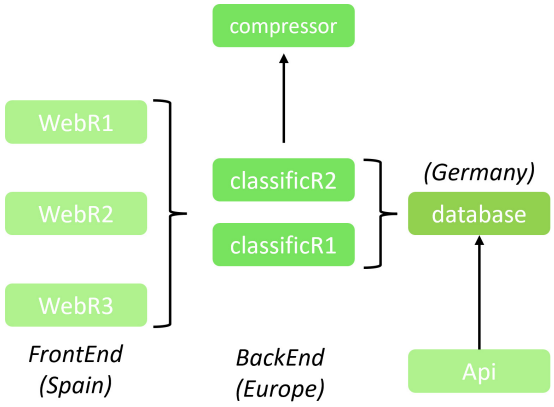

Figure 4.20: Servicios de la aplicación de ejemplo

La Figura 4.20 presenta el esquema de dependencias de una aplicación de ejemplo diseñada para emular una versión simplificada de una aplicación de clasificación de imágenes. En la capa más cercana a los usuarios se encuentra el front-end, una aplicación web progresiva (PWA), que debe desplegarse en España, donde se ubican los usuarios objetivo, y contar con tres réplicas para garantizar la escalabilidad. La capa intermedia corresponde al proceso de clasificación de imágenes, una tarea con una carga computacional moderada, que debe desplegarse en Europa para cumplir con las normativas del RGPD. Además, se incluyen tres servicios complementarios: Una base de datos, que debe estar desplegada en Alemania, sede de la empresa; Una API pública, con bajos requerimientos de cómputo; Un servicio de obtención del modelo del clasificador, que realiza una compresión del modelo, una tarea con alto costo computacional.

Para realizar las pruebas en un entorno realista de Kubernetes, se desplegaron 10 máquinas virtuales en diferentes ubicaciones y cada una de ellas se configuró para alojar un clúster de Kubernetes. Cada máquina virtual se modificó para que tuviera el rendimiento de CPU, la latencia y el ancho de banda definidos mediante las herramientas cpulimit[133] y Traffic Control (tc)[134]. De esta manera,

Para llevar a cabo las pruebas en un entorno realista de Kubernetes, se desplegaron 10 máquinas virtuales en distintas ubicaciones, cada una configurada para alojar un clúster de Kubernetes. Se ajustó el rendimiento de CPU, la latencia y el ancho de banda de cada máquina virtual utilizando las herramientas cpulimit [133] y Traffic Control (tc) [134]. De este modo, se creó un entorno de pruebas con 10 nodos, distribuidos de la siguiente forma: dos nodos representan la nube (ubicados en Hannover y Turing), tres nodos corresponden al nivel Fog (ubicados en Madrid, Valencia y Murcia), y cinco nodos pertenecen al nivel Edge. La Tabla 4.3 resume la información técnica del entorno de pruebas. Como puede

verse, el conjunto de nodos k8s es heterogéneo, ya que tienen distinta potencia de cálculo, capacidades, latencias y ubicaciones.

Table 4.3: Lista de nodos de cloud, fog y edge

	Hannover	**Turin**	**Madrid**	**Valencia**	**Murcia**	**Edge**
milliCPU	8000	10000	6000	6000	6000	4000
RAM	8 GB	10 GB	4 GB	4 GB	4 GB	2 GB
Available CPU	2768	4250	2457	3900	3120	2500 650 2500 1200 3120
Available RAM	6656	7680	3584	2304	3072	1792 1664 1664 1280 1408
CPU Speed	31500	43500	17500	19000	21500	7000 10200 8500 9500 11000
Location	Germany	Italy	Spain	Spain	Spain	Spain
Latency	40ms	55ms	15ms	20ms	25ms	5, 7, 8, 9, 10 ms
Avg Network Delay	4552ms	3538ms	2125ms	1800ms	1535ms	500ms 650ms 800ms 900ms 1000ms
Bandwidth	50 mbs	40 mbs	80 mbs	90 mbs	100 mbs	210-160 mbs
Capabilities	RAM-ECC GPU AES-NI	RAM-ECC GPU AES-NI	RAM-ECC GPU AES-NI	RAM-ECC GPU AES-NI	RAM-ECC GPU AES-NI	AES-NI

El paso esencial para integrar el paradigma del Computing Continuum con Kubernetes y desplegar adecuadamente los servicios de ejemplo es la reimplementación del módulo de planificación (scheduler) por defecto de K8s para que pueda tener en cuenta los requisitos específicos del Continuum. Este módulo se encarga de determinar en qué nodo se desplegará una aplicación cuando se define un servicio, proceso conocido como la asignación de un Pod a un Node. El planificador por defecto de K8s considera únicamente aspectos básicos, como el uso de CPU y RAM. Además, los puntos de extensión que ofrece no son suficientes para incorporar requisitos más complejos, como la latencia máxima permitida o las capacidades de hardware necesarias para cada servicio (por ejemplo, el uso de GPU). Por otra parte, el enfoque de asignación Pod-Node de Kubernetes es de tipo greedy, es decir, asigna cada Pod individualmente sin considerar múltiples asignaciones simultáneamente. Esto conduce, en muchos casos, a soluciones subóptimas y difícilmente puede lograr un buen rendimiento en despliegues de servicios muy numerosos.

Por este motivo, nuestra propuesta consiste en desarrollar un nuevo módulo de planificación con un comportamiento más avanzado. El primer aspecto clave de este planificador es su capacidad para realizar despliegues en bloques de Pods (batches), lo que permite obtener soluciones más óptimas. En esencia, el problema que debe resolver es una variante

del problema de asignación de tareas a nodos, similar al descrito en la ecuación 4.5, el cual se puede abordar con distintos algoritmos dependiendo del tiempo disponible y la precisión requerida. El segundo aspecto clave de la solución es la posibilidad de elegir, de manera dinámica o a petición del usuario, el método de optimización más adecuado para resolver el problema. Se puede seleccionar entre varios enfoques, como métodos óptimos, greedy, algoritmos genéticos, optimización multiobjetivo, round-robin, el planificador por defecto de Kubernetes, entre otros. Esta aproximación logra satisfacer las especificaciones del Continuum además de ofrecer cierta flexibilidad en el uso de K8s.

Para demostrar el funcionamiento de la solución desarrollada se realizan una serie de experimentos sobre la infraestructura de pruebas descrita anteriormente, en la cual se despliegan los servicios definidos tres veces para forzar la saturación de los nodos y ver donde acaban finalmente desplegados los servicios. La figura 4.21 muestra un ejemplo de asignación de los servicios, mostrando el estado inicial de los nodos y las tres etapas de despliegue batch de los mismo servicios.

Figure 4.21: Evolución de la asignación de batches de servicios a la infraestructura

El primer paso muestra el estado inicial de los nodos que hemos definido, mostrando a la derecha de cada uno de ellos una barra con el porcentaje de la cantidad de CPU requerida por los servicios del nodo y debajo un número con la cantidad de miliCPU disponibles. En el siguiente paso podemos ver cómo se asigna el conjunto de servicios del caso de uso (utilizando el algoritmo óptimo) en la infraestructura, destacando cómo la base de datos se coloca en el nodo de Alemania (requisito de localización), las tareas más pesadas (clasificar y comprimir) van a parar a los nodos más rápidos (Turin y Madrid) y el resto de servicios web se distribuyen entre los nodos con menor latencia. Por tanto, se confirma que se están respetando los requisitos del Continuum, ya que los servicios se distribuyen en una infraestructura multinivel de la forma más óptima en función de la latencia, la velocidad de cálculo, etc., y respetando también restricciones como la ubicación y la antiafinidad.

Sin embargo, para confirmar que propuesta funciona correctamente y de forma dinámica,

es necesario forzar a los nodos a alcanzar su capacidad máxima para que el método tenga que buscar nodos alternativos que respeten las restricciones. Para ello, la asignación de los servicios del caso de uso se repite dos veces más (sin sobrescribirlos), de forma que algunos de los nodos saturan su capacidad. Esto puede observarse en la tercera etapa (abajo a la izquierda), en la que el servicio de compresión se asigna de nuevo al nodo más potente (Turin) llevándolo al límite de la capacidad de la CPU, de forma que el servicio de clasificación se distribuye entre los dos siguientes nodos más potentes (Madrid y Valencia).

Del mismo modo, a medida que continuamos hasta el último paso (abajo a la derecha) se hace más complicado asignar servicios ya que muchos de los mejores nodos candidatos ya están al máximo de su capacidad, por lo que los servicios intensivos en computación empiezan a distribuirse cada vez más cerca del borde, por ejemplo el servicio de compresión se envía al nodo de Murcia ya que es el que menos trabajo tiene y tiene una potencia de computación aceptable. Paralelamente se aprecia como en todo momento los servicios que requieren menor latencia (Web y Api) tienen a estar concentrados en los nodos Edge.

Tras analizar gráficamente un ejemplo de cómo se distribuirían los pods por los nodos utilizando es sistema propuesto, procederemos a describir el segundo conjunto de pruebas realizadas para demostrar el funcionamiento del planificador desarrolado. Para estas pruebas, desplegaremos los servicios del caso de uso tres veces (d1, d2, d3) sobre la infraestructura inicial de la misma forma que en la prueba anterior, utilizando cada uno de los métodos descritos: el despliegue completo de todos los servicios, el despliegue de cada batch de servicios individualmente, el despliegue de cada batch de servicios individualmente utilizando el algoritmo Genético en lugar del anterior basado en Backtracking, el despliegue pod-pod con el algoritmo greedy (similar a como haría K8s pero considerando los requisitos del Continuum), el algoritmo que asigna pods aleatoriamente cumpliendo las restricciones, un método basado en round-robin, y por último, un algoritmo completamente aleatorio. A continuación, se mide el resultado de la asignación de nodos pod en la infraestructura de nodos virtuales realizando una serie de pruebas de estrés con la herramienta JMeter de Apache. Estas pruebas evaluarán si el rendimiento de los servicios desplegados, concretamente el tiempo medio de respuesta, se ve afectado cuando se incrementa significativamente la carga sobre ellos. Los resultados completos de todas las pruebas se resumen en la Figura 4.22, mostrando el tiempo medio de respuesta para cada servicio y despliegue según el método de asignación utilizado. Para realizar las pruebas de estrés, se ha configurado JMeter de forma que cada 4 segundos se genera una petición a cada servicio multiplicada por el número de usuarios de dicho servicio, simulando así un uso intensivo de todas las aplicaciones desplegadas en la infraestructura. Al inicio de la prueba JMeter, los nodos son capaces de resolver las peticiones a los servicios en pocos milisegundos, pero a medida que aumenta el número de peticiones, se saturan, provocando que algunos servicios tarden varios segundos en procesar cada petición. El resultado final de la prueba será el tiempo medio de respuesta total de cada uno de los servicios de cada despliegue teniendo en cuenta todas las peticiones realizadas durante la prueba de estrés.

Como se muestra en la Figura 4.22, el peor resultado (barras más altas, por tanto mayor tiempo de respuesta) proviene del algoritmo totalmente aleatorio (barras grises), lo cual es obvio ya que la elección de los nodos no es en absoluto adecuada. Es más, se puede

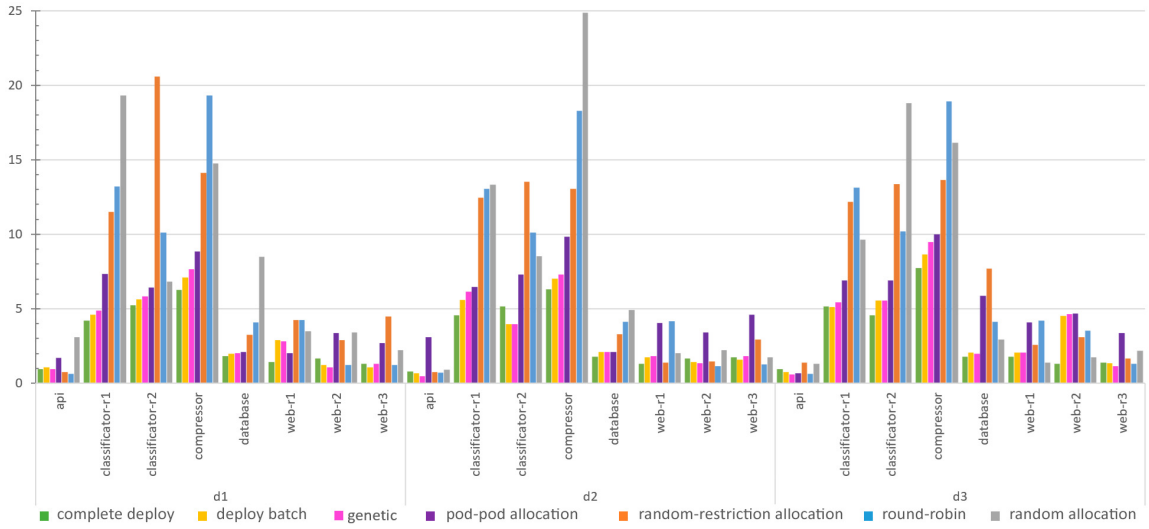

Figure 4.22: Tiempo medio de respuesta de cada servicio para cada método

observar como ni siquiera es una asignación óptima en el sentido de utilizar adecuadamente los recursos disponibles, ya que algunos servicios (por ejemplo, d3-web-r1 y d3-web-r2) tienen un tiempo de respuesta inferior al resto del método, esto es debido a que están asignados en nodos muy potentes por lo que no están saturados, pero ocupan recursos muy caros necesarios para otros servicios con una carga de trabajo mayor (compresor o clasificador). Del mismo modo, el método basado en round-robin (barras azules) tiene un rendimiento reducido que es ligeramente superior al método anterior simplemente porque distribuye los servicios de forma equitativa entre los nodos.

Por otro lado, el método aleatorio que respeta las restricciones (barras naranjas) mejora el tiempo de respuesta al tener en cuenta las restricciones del problema, pero en algunos servicios (por ejemplo, d1-classif-r2) da un mal resultado, ya que sigue siendo un método aleatorio que no elige el mejor nodo. Sin embargo, como ya se ha mencionado, es un buen método para definir el rendimiento de referencia que debe mejorarse, ya que tiene en cuenta los requisitos del continuo pero no elige el mejor nodo posible.

Como primera aproximación a la tarea de asignar pods a nodos tenemos el algoritmo greedy (barras moradas) que sigue un patrón de asignación pod-pod pero tiene en cuenta que el mejor nodo (menor valor de la función de coste) debe ser elegido del conjunto de nodos válidos (cumplen los requisitos del despliegue). Como se puede observar en el gráfico, en todos los servicios el resultado es aceptable y se mantiene similar entre los tres despliegues, sin embargo algunos servicios con poca carga de trabajo (por ejemplo web) tienden a tener un bajo rendimiento ya que muchos acaban en los mismos nodos. Dado que la asignación pod por pod se realiza de forma individual no es posible optimizar el conjunto completo de asignaciones de forma que algunos nodos que tienen una buena función de coste para un determinado servicio acaben con varias instancias de ese servicio ya que en cada etapa aparece el mejor candidato, cuando si se consideran todas las asignaciones a la vez es posible identificar que es mejor desplegarlas en otros nodos aunque su función de coste sea peor pero se optimiza el coste total.

Añadir la posibilidad de poder resolver el problema de asignación en bloques de servi-

cios (batch) conlleva una clara mejora en el rendimiento de las aplicaciones. El resultado se observa en las barras amarillas que representan el método que despliega los pods en tres lotes para optimizar la elección de nodos en cada lote mediante un algoritmo óptimo. Este método ofrece un rendimiento mucho mejor que los demás, donde se observa una clara mejora entre los esquemas pod-pod y batch, lo que justifica la necesidad de mejorar el proceso de asignación pod-nodo para que pueda considerar varios pods al mismo tiempo. Además, la alternativa basada en un Algoritmo Genético ofrece un rendimiento bastante similar al óptimo basado en Backtracking, pero con un menor tiempo de ejecución, lo que permite su aplicación en entornos reales donde el tiempo de ejecución es crítico, aunque la solución no sea la mejor de todas.

Por último, realizamos una prueba en la que ejecutamos el método óptimo (barras verdes) para realizar la asignación de los tres deploys como uno solo, dando así el máximo resultado óptimo de nuestras pruebas. Podemos apreciar en el gráfico como mejora el tiempo de respuesta de los servicios ya que la asignación es ligeramente mejor, pero aun así la mejora es limitada, por lo que se puede deducir que dividir el deploy entre tres no tiene un impacto crítico en el rendimiento y permite que computacionalmente sea posible resolver el problema de optimización reduciendo considerablemente el tamaño del mismo.

En conclusión, aunque el planificador nativo de Kubernetes es funcional en la mayoría de los casos, no resulta óptimo para escenarios de Continuum Computing, como demuestran los ejemplos Round-Robin y Random-Restriction. Integrar los requisitos del Continuum en el planificador permite mejorar el rendimiento, como se observa con métodos que respetan restricciones específicas o aplican estrategias greedy de asignación Pod-Pod. Sin embargo, el rendimiento sigue estando limitado por el esquema de asignación secuencial Pod-a-Pod que utiliza Kubernetes. Para lograr mejoras significativas, es necesario implementar asignaciones en bloque (por lotes o batch). Esto se evidencia en los ejemplos de despliegues por lotes, donde las mejoras son claras en situaciones de alta demanda o infraestructura saturada, ya que se eligen los nodos de manera más eficiente al considerar el conjunto completo de servicios.

Por otro lado, a partir de las conclusiones obtenidas se plantea aplicar la solución en un escenario más sofisticado con un orquestador avanzado basado en "*Intents*" que permite no solo usar K8s sino cualquier otro tipo de plataforma para desplegar los servicios. Esta propuesta se detalla en la publicación "*Dynamic Multi-Method Allocation for Intent-based Security Orchestration*"[41], donde se describe un sistema que, a partir de un conjunto de políticas, determina la mejor implementación (software) y el nodo más adecuado para el despliegue de cada una de ellas. Además, dado el mayor tamaño del problema a resolver, se propone añadir un metaorquestador como componente adicional. Este metaorquestador decide dinámicamente qué método utilizar según el tamaño del problema y el tiempo disponible. Por ejemplo, para políticas con pocos elementos, se puede elegir un algoritmo óptimo o de alta precisión. En cambio, para políticas con muchos elementos, se optaría por un algoritmo subóptimo que requiera menor tiempo de ejecución, como un algoritmo genético.

Capítulo 5

Conclusiones y trabajo futuro

En las últimas décadas, el avance de la tecnología ha sido innegablemente imparable, con un crecimiento exponencial en las capacidades y posibilidades de los dispositivos año tras año. En tiempos recientes, hemos presenciado la expansión de internet hasta el punto de aspirar a que todos los dispositivos sean capaces de conectarse a la red. Esto, sumado al desarrollo de dispositivos compactos que incorporan capacidades de cómputo en objetos cotidianos, ha transformado profundamente nuestra interacción con la tecnología. Estos avances convergen hacia un futuro cada vez más "conectado" e "inteligente". Ejemplo de ello es la amplia adopción del concepto de Internet de las Cosas y el protagonismo de la Inteligencia Artificial, que ha captado gran atención en los últimos años gracias a importantes progresos. Entre ellos destacan los modelos extensos de lenguaje (LLM) para la creación de chatbots y los modelos de difusión capaces de generar imágenes a partir de texto, marcando un punto de inflexión en la evolución tecnológica.

No obstante, todo avance trae consigo nuevos retos y complicaciones. En este sentido, el creciente número de dispositivos conectados a la red, ya sean aquellos que siguen el modelo tradicional de IoT o los basados en las nuevas técnicas de Edge Computing, plantea importantes desafíos en distintas áreas de la informática, que abarcan desde la seguridad hasta la gestión eficiente de los datos. Esto no solo requiere desarrollar nuevas estrategias de seguridad para proteger la privacidad y la información sensible de los usuarios, sino también optimizar el uso de los recursos de red y mejorar la interoperabilidad entre diferentes plataformas y dispositivos.

Un paradigma emergente y muy prometedor es el Edge Computing, que resalta el papel crucial de los dispositivos más cercanos a los usuarios. Estos dispositivos deben ser capaces de llevar a cabo parte del procesamiento necesario para los servicios, e incluso la totalidad de este, limitando la interacción con la nube a consultas puntuales o al envío de datos ya procesados o agregados. Esta estrategia desplaza la computación hacia los dispositivos cercanos a los usuarios, lo que permite reducir la latencia y optimizar el uso del ancho de banda general. No obstante, la incorporación de un gran número de dispositivos heterogéneos en el sistema dificulta la toma de decisiones sobre dónde ejecutar las tareas necesarias para los servicios. Este desafío se intensifica con la propuesta del Computing Continuum, que plantea que cualquier elemento dentro del sistema, desde la nube hasta el usuario final, podría ser capaz de realizar tareas de procesamiento.

Como ya hemos mencionado, a este problema se le conoce como el Problema de Asignación de Tareas, el cual consiste en determinar el conjunto óptimo de asignaciones de tareas a nodos con el objetivo de optimizar una función de costo específica y cumplir diversas restricciones. Para abordarlo, existen múltiples enfoques, que van desde soluciones óptimas hasta alternativas subóptimas basadas en métodos heurísticos.

Con base en lo anterior, el objetivo principal de esta tesis es analizar los fundamentos del proceso de distribución de cargas de trabajo entre nodos, identificar posibles oportunidades de mejora y desarrollar el software necesario para su implementación en sistemas reales, alineándolos con el concepto de Computing Continuum. En consecuencia, se pueden resumir las principales conclusiones obtenidas durante la realización de la tesis en la siguiente lista:

- Aunque el concepto de Internet de las Cosas esté ampliamente estudiado, existen aun ciertas limitaciones cuando se aplican enfoques de computación distribuida entre un conjunto de nodos. Existen muy pocos enfoques que consideren IoT como un sistema colaborativo.

- En la actualidad las redes IoT no están preparadas para adaptarse a la idea de Edge Computing, dificultando notoriamente el despliegue de sistemas basados en Computing Continuum. El paradigma IoT sigue el enfoque clásico de interacción continua con la nube y no realizar apenas preprocesamiento.

- Existen paralelamente casos de uso y escenarios que pueden hacer uso de la tecnología de Edge y Continuum con un gran potencial, como pueden ser ciudades y hogares inteligentes. Servirían como una evolución de los ejemplos ampliamente estudiados de IoT para considerar las nuevas propuestas de Edge y Continuum Computing

- El problema de asignación de tareas a nodos es bien conocido pero no se trata de manera adecuada en IoT y Edge Computing, además de no contemplar requisitos muy importantes del Continuum como pueden ser las diferentes características hardware de cada nodo.

- Se reconoce el potencial para el problema de asignación tarea-nodo de las técnicas basadas en aprendizaje por refuerzo, dado que son capaces de adaptase en tiempo real a cambios bruscos y anticiparse a eventos periódicos.

- Aun así, se demuestra en un escenario de Edge Computing las limitaciones del enfoque tradicional de RL y se propone una mejora de la técnica para que explote la arquitectura basada en capas. En esta mejora se demuestra el rendimiento positivo de permitir a los agentes RL delegar la decisión a otros agentes y aprender de lo que hagan.

- Las ciudades y casas inteligentes también pueden beneficiarse del concepto de Continuum y aplicar técnicas de aprendizaje por refuerzo para gestionar sus servicios. Esto queda demostrado con una serie de experimentos realizados en la simulación de una casa inteligente.

- Actualmente no se aplican técnicas de machine learning ni arquitecturas como Edge Computing en las soluciones mas populares de la industria, como Kubernetes, por lo que es necesario desarrollar todo un conjunto de herramientas para poder implementar nuestras propuestas en estos sistemas.
- Todas las soluciones desarrolladas han sido probadas en diferentes casos de uso para demostrar su eficacia, además de haber usado parte de los resultados para resolver problemas semejantes en otros proyectos.

Los resultados y conclusiones obtenidos en esta tesis han contribuido a consolidar el conocimiento fundamental sobre la arquitectura del Computing Continuum, así como de temas afines, entre los que destacan áreas de especial interés como las el problema de asignación de tareas a nodos, aprendidaje por refuerzo, Smart Cities y la optimización energética.

5.1 Trabajo futuro

A lo largo del desarrollo de esta tesis, han surgido numerosos temas interesantes que, por razones de alcance, no pudieron ser abordados dentro de los objetivos iniciales. Asimismo, al profundizar en los distintos aspectos del Computing Continuum, emergieron nuevas líneas de investigación y desafíos que representan oportunidades para futuros estudios.

Como primer trabajo futuro destacable tendríamos el uso de de técnicas mucho mas complejas de aprendizaje por refuerzo, como las basadas en redes neuronales (DeepRL). Esta versión mejorada de RL soluciona la limitación de que el espacio de estado tiene que ser discreto, pero añade un sobrecoste de memoria y computo difícilmente asumible por los dispositivos mas pequeños del Continuum. Por lo tanto, un punto interesante para futuros trabajos seria el de estudiar técnicas de DeepRL con redes muy pequeñas e implementarlas de la manera mas reducida posible para integrarlas en dispositivos con poca capacidad de cómputo.

Por otro lado, el orquestador desarrollado e implementado en Kubernetes demuestra un gran potencial, no solo para gestionar el Continuum y resolver el problema de asignación tarea-nodo, sino también para abordar otras cuestiones, como la toma de decisiones en temas de seguridad, la selección del software que implementará el servicio requerido, entre muchas otras. En este contexto, se considera que un camino interesante para futuros trabajos sería mejorar las capacidades del orquestador, tanto en cuanto a sus funcionalidades como a su semántica, de manera que pueda comprender una mayor variedad de restricciones y requerimientos en sus despliegues. Además, sería valioso incorporar una amplia gama de algoritmos que permitan resolver dinámicamente los problemas de decisión, adaptándose a entornos más complejos y variables. Precisamente, estos temas ya están siendo explorados como base para futuros trabajos tras la tesis. Entre los aspectos destacados se encuentra la ampliación del análisis del problema de optimización, incorporando condiciones relacionadas con la seguridad y privacidad de cada nodo particular, lo que permitiría excluir nodos si resultan poco confiables según estos parámetros. Además, aunque algunos algoritmos sobresalgan frente a otros, se ha demostrado la utilidad de

disponer de una variedad de métodos para resolver el problema de asignación con mayor o menor precisión, a cambio de memoria o tiempo. Por consiguiente, otra dirección de investigación futura que se está abordando es la meta-orquestación, que permitirá seleccionar dinámicamente el algoritmo adecuado en cada petición según el tamaño del problema, los requisitos de tiempo, la memoria disponible y otros parámetros relevantes.

Por último, en el problema de asignación de tareas a nodos, la investigación se ha centrado principalmente en métodos que buscan optimizar un único objetivo, aunque este pueda integrar varios factores mediante una suma ponderada o un estado conjunto. Sin embargo, no se ha explorado en profundidad el uso de métodos puramente multiobjetivo, que permitirían abordar de manera simultánea múltiples criterios independientes. Esto abre una línea de investigación prometedora, ya que los métodos multiobjetivo podrían ofrecer soluciones más equilibradas y flexibles en escenarios complejos.

De forma paralela a la investigación y el trabajo desarrollado en esta tesis, se ha colaborado en proyectos europeos que han contribuido a ampliar los horizontes de la investigación realizada y han proporcionado un entorno para aplicar las técnicas desarrolladas. Por ello, como trabajo futuro, se propone continuar ampliando la integración de las soluciones en los casos de uso de dichos proyectos. Un ejemplo de ello es el trabajo desarrollado en el proyecto FLUIDOS, donde la idea del continuum desempeña un papel fundamental. En este proyecto se utiliza Kubernetes y orquestadores personalizados para los despliegues, lo que guarda una estrecha relación con el trabajo realizado en esta tesis. Gracias a ello, muchas de las ideas propuestas se han podido validar en los escenarios del proyecto. Asimismo, se están explorando propuestas muy interesantes que podrían formar parte de estudios futuros, tales como la aplicación de intents avanzados en el proceso de orquestación, la incorporación de un sistema de reserva y tarificación para el uso de nodos, la implementación de descriptores de recursos ofrecidos (flavors) y el desarrollo de metodologías de seguridad para proteger el proceso de offloading en nodos externos.

Capítulo 6

Publicaciones que componen la tesis doctoral

6.1 A multi-layer guided reinforcement learning-based tasks offloading in edge computing

Abstract The breakthrough in Machine Learning (ML) techniques and the popularity of the Internet of Things (IoT) has increased interest in applying Artificial Intelligence (AI) techniques to the new paradigm of Edge Computing. One of the challenges in edge computing architectures is the optimal distribution of the generated tasks between the devices in each layer (i.e., cloud-fog-edge). In this paper, we propose to use Reinforcement Learning (RL) to solve the Task Assignment Problem (TAP) at the edge layer and then we propose a novel multi-layer extension of RL (ML-RL) techniques that allows edge agents to query an upper-level agent with more knowledge to improve the performance in complex and uncertain situations. We first formulate the task assignment process considering the trade-off between energy consumption and execution time. We then present a greedy solution as a baseline and implement our multi-layer RL proposal in the PureEdgeSim simulator. Finally several simulations of each algorithm are evaluated with different numbers of devices to verify scalability. The simulation results show that reinforcement learning solutions outperformed the heuristic-based solutions and our multi-layer approach can significantly improve performance in high device density scenarios.

Title	A multi-layer guided reinforcement learning-based tasks offloading in edge computing
Authors	Alberto Robles-Enciso, Antonio Skarmeta
Journal	Computer Networks
Impact factor (2021)[3]	5.49
JCR Rank (2021)[3]	Computer science, hardware & architecture: 8/54 (84.2, Q1) Computer science, information systems: 41/158 (78.8, Q1)
Publisher	Elsevier
Date	November 2022
ISSN	1389-1286
EISSN	1872-7069
DOI	https://doi.org/10.1016/j.comnet.2022.109476
State	Published
Contribution	Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Investigation

6.2 Sim-PowerCS: An extensible and simplified open-source energy simulator

Abstract In today's world, optimizing energy consumption is critical to combat climate change and natural resource depletion. Smart control of building and home devices is a promising approach to achieving this goal, but due to the difficulties of testing in real environments it is necessary to rely on simulators for a comprehensive evaluation of algorithms in a realistic simulated context at a low cost. In this paper we present a Java-based simulator that allows users to define energy sources and services, test different algorithms, run procedural simulations, and independently optimize energy management and service control. Sim-PowerCS is designed to be easy to understand, use, and extend, enabling any researcher to quickly adapt it to their specific use cases.

Title	Sim-PowerCS: An extensible and simplified open-source energy simulator
Authors	Alberto Robles-Enciso, Ricardo Robles-Enciso, Antonio Skarmeta
Journal	SoftwareX
Impact factor (2022)[3]	3.4
JCR Rank (2022)[3]	Computer science, software engineering: 38/108 (65.3, Q2)
Publisher	Elsevier
Date	June 2023
ISSN	2352-7110
EISSN	2352-7110
DOI	https://doi.org/10.1016/j.softx.2023.101467
State	Published
Contribution	Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing

6.3 Adapting Containerized Workloads for the Continuum Computing

Abstract Container and microservices management platforms are currently one of the most important tools for cloud computing, but since the scope of these tools is homogeneous cloud architectures they have serious limitations in adapting to new computing paradigms. Therefore, using the default scheduler in heterogeneous node systems faces significant limitations when tasked with orchestrating workloads in a Continuum Computing environment, as the nodes have very different characteristics and restrictions. To solve this limitation we decided to use Kubernetes as it is the most popular Container management tool and we propose to replace the native scheduler with a reimplementation that gives us complete flexibility for the process of assigning pods to nodes, providing a framework to design algorithms that considers all the necessary parameters for the deployment of services in a Continuum. In addition, we address one of the most limiting aspects of the K8s scheduler, its pod-by-pod allocation approach, which makes it difficult to optimise the complete set of allocations. To test our proposal we design a use case and perform several tests on a real environment based on virtual machines, in which stress tests are conducted to measure the performance of each method. We then present a series of results to justify the benefits of our proposal, including the reduced performance provided by the pod-pod approach and how a batch-based approach greatly improves efficiency. The results show the usefulness of using batch-based approaches and how the Kubernetes scheduler extension points are not enough to support the requirements of the Continuum.

Title	Adapting Containerized Workloads for the Continuum Computing
Authors	Alberto Robles-Enciso, Antonio Skarmeta
Journal	IEEE Access
Impact factor (2023)[3]	3.4
JCR Rank (2023)[3]	Computer science, information systems: 87/249 (65.3, Q2) Engineering, electrical & electronic: 122/352 (65.5, Q2)
Publisher	IEEE
Date	July 2024
ISSN	2169-3536
EISSN	2169-3536
DOI	`https://doi.org/10.1109/ACCESS.2024.3434585`
State	Published
Contribution	Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Investigation

Bibliografía

[1] A. Čolaković, M. Hadžialić, Internet of things (iot): A review of enabling technologies, challenges, and open research issues, Computer Networks 144 (2018) 17–39. doi:https://doi.org/10.1016/j.comnet.2018.07.017.

[2] T. H. Noor, S. Zeadally, A. Alfazi, Q. Z. Sheng, Mobile cloud computing: Challenges and future research directions, Journal of Network and Computer Applications 115 (2018) 70–85. doi:https://doi.org/10.1016/j.jnca.2018.04.018.

[3] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, R. Ranjan, Fog computing: Survey of trends, architectures, requirements, and research directions, IEEE Access 6 (2018) 47980–48009. doi:10.1109/ACCESS.2018.2866491.

[4] S. Tanwar, S. Tyagi, I. Budhiraja, N. Kumar, Tactile internet for autonomous vehicles: Latency and reliability analysis, IEEE Wireless Communications 26 (04 2019). doi:10.1109/MWC.2019.1800553.

[5] A. Mehrabi, M. Siekkinen, T. Kämäräinen, A. Yla-Jaaski, Multi-tier cloudvr: Leveraging edge computing in remote rendered virtual reality, ACM Trans. Multimedia Comput. Commun. Appl. 17 (2) (may 2021). doi:10.1145/3429441.

[6] J. Liu, Q. Zhang, Offloading schemes in mobile edge computing for ultra-reliable low latency communications, IEEE Access 6 (2018) 12825–12837. doi:10.1109/ACCESS.2018.2800032.

[7] K. Cao, Y. Liu, G. Meng, Q. Sun, An overview on edge computing research, IEEE Access 8 (2020) 85714–85728. doi:10.1109/ACCESS.2020.2991734.

[8] K. Zhang, S. Leng, Y. He, S. Maharjan, Y. Zhang, Mobile edge computing and networking for green and low-latency internet of things, IEEE Communications Magazine 56 (5) (2018) 39–45. doi:10.1109/MCOM.2018.1700882.

[9] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet of Things Journal 5 (1) (2018) 439–449. doi:10.1109/JIOT.2017.2767608.

[10] D. Evans, The internet of things, How the Next Evolution of the Internet is Changing Everything, Whitepaper, Cisco Internet Business Solutions Group (IBSG) 1 (2011) 1–12.

[11] W. Shi, S. Dustdar, The promise of edge computing, Computer 49 (5) (2016) 78–81. doi:10.1109/MC.2016.145.

[12] D. Kimovski, R. Mathá, J. Hammer, N. Mehran, H. Hellwagner, R. Prodan, Cloud, fog, or edge: Where to compute?, IEEE Internet Computing 25 (4) (2021) 30–36. doi:10.1109/MIC.2021.3050613.

[13] C. Bu, J. Wang, Computing tasks assignment optimization among edge computing servers via sdn, Peer-to-Peer Networking and Applications 14 (2021) 1190–1206.

[14] T. Öncan, A survey of the generalized assignment problem and its applications, INFOR: Information Systems and Operational Research 45 (3) (2007) 123–141.

[15] C. Chakraborty, K. Mishra, S. K. Majhi, H. K. Bhuyan, Intelligent latency-aware tasks prioritization and offloading strategy in distributed fog-cloud of things, IEEE Transactions on Industrial Informatics 19 (2) (2023) 2099–2106. doi:10.1109/TII.2022.3173899.

[16] S. K. Mishra, S. Mishra, A. Alsayat, N. Z. Jhanjhi, M. Humayun, K. S. Sahoo, A. K. Luhach, Energy-aware task allocation for multi-cloud networks, IEEE Access 8 (2020) 178825–178834. doi:10.1109/ACCESS.2020.3026875.

[17] F. Faticanti, M. Savi, F. De Pellegrini, D. Siracusa, Locality-aware deployment of application microservices for multi-domain fog computing, Computer Communications 203 (2023) 180–191. doi:https://doi.org/10.1016/j.comcom.2023.02.012.
URL https://www.sciencedirect.com/science/article/pii/S0140366423000506

[18] U. Saleem, Y. Liu, S. Jangsher, Y. Li, T. Jiang, Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing, IEEE Transactions on Wireless Communications 20 (1) (2021) 360–374. doi:10.1109/TWC.2020.3024538.

[19] D. Romero Morales, H. Romeijn, The Generalized Assignment Problem and Extensions, Vol. B, Springer, Germany, 2005, pp. 259–311. doi:10.1007/b102533.

[20] M. W. P. Savelsbergh, A branch-and-price algorithm for the generalized assignment problem, Oper. Res. 45 (1997) 831–841.

[21] M. Garey, D. Johnson, Computers and intractability: A guide to the theory of np-completeness, 1978.

[22] J. Bellendorf, Z. Ádám Mann, Classification of optimization problems in fog computing, Future Generation Computer Systems 107 (2020) 158–176. doi:https://doi.org/10.1016/j.future.2020.01.036.

[23] J. Hassannataj Joloudari, S. Mojrian, H. Saadatfar, I. Nodehi, F. Fazl, S. Khanjani Shirkharkolaie, R. Alizadehsani, H. M. D. Kabir, R. S. Tan, U. Acharya, Resource allocation problem and artificial intelligence: the state-of-the-art review (2009–2023) and open research challenges, Multimedia Tools and Applications (2024) 1–44doi:10.1007/s11042-024-18123-0.

[24] E. H. Houssein, A. G. Gad, Y. M. Wazery, P. N. Suganthan, Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends, Swarm and Evolutionary Computation 62 (2021) 100841. doi:https://doi.org/10.1016/j.swevo.2021.100841.
URL https://www.sciencedirect.com/science/article/pii/S221065022100002X

[25] C. Ren, X. Lyu, W. Ni, H. Tian, W. Song, R. P. Liu, Distributed online optimization of fog computing for internet of things under finite device buffers, IEEE Internet of Things Journal 7 (6) (2020) 5434–5448. doi:10.1109/JIOT.2020.2979353.

[26] J. Liang, Y. Long, Y. Mei, T. Wang, Q. Jin, A distributed intelligent hungarian algorithm for workload balance in sensor-cloud systems based on urban fog computing, IEEE Access 7 (2019) 77649–77658. doi:10.1109/ACCESS.2019.2922322.

[27] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, M. Rovatsos, Fog orchestration for internet of things services, IEEE Internet Computing 21 (2) (2017) 16–24. doi:10.1109/MIC.2017.36.

[28] A. Shakarami, M. Ghobaei-Arani, A. Shahidinejad, A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective, Computer Networks 182 (2020) 107496. doi:https://doi.org/10.1016/j.comnet.2020.107496.

[29] J. Liu, G. Wang, X. Guo, S. Wang, Q. Fu, Deep reinforcement learning task assignment based on domain knowledge, IEEE Access 10 (2022) 114402–114413. doi:10.1109/ACCESS.2022.3217654.

[30] M. A. Wiering, Convergence and divergence in standard and averaging reinforcement learning, in: European conference on machine learning, Springer, 2004, pp. 477–488.

[31] T. Xu, Y. Liang, G. Lan, Crpo: A new approach for safe reinforcement learning with convergence guarantee, in: International Conference on Machine Learning, PMLR, 2021, pp. 11480–11491.

[32] Flexible, scaLable and secUre decentralIzeD Operating System - horizon europe, https://cordis.europa.eu/project/id/101070473, accessed: 2024-12-01.

[33] Adapt-&-Play Holistic cOst-Effective and user-frieNdly Innovations with high replicability to upgrade smartness of eXisting buildings with legacy equipment - horizon europe, https://cordis.europa.eu/project/id/893079, accessed: 2024-12-01.

[34] A. Robles-Enciso, A. F. Skarmeta, Task offloading in computing continuum using collaborative reinforcement learning, in: A. González-Vidal, A. Mohamed Abdelgawad, E. Sabir, S. Ziegler, L. Ladid (Eds.), Internet of Things, Springer International Publishing, Cham, 2022, pp. 82–95.

[35] A. Robles-Enciso, A. F. Skarmeta, A multi-layer guided reinforcement learning-based tasks offloading in edge computing, Computer Networks 220 (2023) 109476. doi:

https://doi.org/10.1016/j.comnet.2022.109476.
URL https://www.sciencedirect.com/science/article/pii/S1389128622005102

[36] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta, Multi-agent reinforcement learning-based energy orchestrator for cyber-physical systems, in: I. Chatzigiannakis, I. Karydis (Eds.), Algorithmic Aspects of Cloud Computing, Springer Nature Switzerland, Cham, 2024, pp. 100–114.

[37] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta Gómez, An adaptive energy orchestrator for cyberphysical systems using multiagent reinforcement learning, Smart Cities 7 (6) (2024) 3210–3240. doi:10.3390/smartcities7060125.
URL https://www.mdpi.com/2624-6511/7/6/125

[38] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta, Sim-powercs: An extensible and simplified open-source energy simulator, SoftwareX 23 (2023) 101467. doi:https://doi.org/10.1016/j.softx.2023.101467.
URL https://www.sciencedirect.com/science/article/pii/S2352711023001632

[39] P. Gonzalez-Gil, A. Robles-Enciso, J. A. Martínez, A. F. Skarmeta, Architecture for orchestrating dynamic dnn-powered image processing tasks in edge and cloud devices, IEEE Access 9 (2021) 107137–107148. doi:10.1109/ACCESS.2021.3101306.

[40] A. Robles-Enciso, A. F. Skarmeta, Adapting containerized workloads for the continuum computing, IEEE Access 12 (2024) 104102–104114. doi:10.1109/ACCESS.2024.3434585.

[41] A. Robles-Enciso, J. M. Bernabé Murcia, A. Molina Zarca, A. Skarmeta Gomez, Dynamic multi-method allocation for intent-based security orchestration, Journal of Network and Systems Management 33 (1) (2024) 1–28. doi:10.1007/s10922-024-09896-8.
URL https://doi.org/10.1007/s10922-024-09896-8

[42] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, Society for Industrial and Applied Mathematics, 2012. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611972238, doi:10.1137/1.9781611972238.
URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972238

[43] T. Öncan, A survey of the generalized assignment problem and its applications, INFOR: Information Systems and Operational Research 45 (2007) 123 – 141.
URL https://api.semanticscholar.org/CorpusID:1696231

[44] P. Avella, M. Boccia, I. Vasilyev, A computational study of exact knapsack separation for the generalized assignment problem, Computational Optimization and Applications 45 (2010) 543–555.

[45] A. Turab, W. Sintunavarat, On the existence and uniqueness of the solution of a probabilistic functional equation approached by the banach fixed point theorem, 2020.

[46] V. Heidrich-Meisner, M. Lauer, C. Igel, M. A. Riedmiller, Reinforcement learning in a nutshell, in: ESANN, 2007.

[47] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[48] D. Rahbari, M. Nickray, Low-latency and energy-efficient scheduling in fog-based iot applications, Turkish Journal of Electrical Engineering and Computer Sciences 27 (2019) 1406–1427.

[49] Z. Jia, J. Yu, X. Ai, X. Xu, D. Yang, Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm, Aerospace Science and Technology 76 (2018) 112–125. doi:https://doi.org/10.1016/j.ast.2018.01.025.

[50] T. Sen, H. Shen, Machine learning based timeliness-guaranteed and energy-efficient task assignment in edge computing systems, 2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC) (2019) 1–10.

[51] J. Wang, L. Zhao, J. Liu, N. Kato, Smart resource allocation for mobile edge computing: A deep reinforcement learning approach, IEEE Transactions on Emerging Topics in Computing (2019) 1–1.

[52] X. Huang, S. Leng, S. Maharjan, Y. Zhang, Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks, IEEE TVT 70 (9) (2021) 9282–9293. doi:10.1109/TVT.2021.3096928.

[53] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, Y. Zhang, Reinforcement learning-based mobile offloading for edge computing against jamming and interference, IEEE Transactions on Communications 68 (10) (2020) 6114–6126. doi:10.1109/TCOMM.2020.3007742.

[54] M. F. Argerich, J. Fürst, B. Cheng, Tutor4rl: Guiding reinforcement learning with external knowledge, in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2020.

[55] D. Rahbari, M. Nickray, Task offloading in mobile fog computing by classification and regression tree, Peer-to-Peer Networking and Applications 13 (2020) 104–122.

[56] M. Adhikari, M. Mukherjee, S. Srirama, Dpto: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing, IEEE Internet of Things Journal 7 (2020) 5773–5782.

[57] L. Liu, D. Qi, N. Zhou, Y. Wu, A task scheduling algorithm based on classification mining in fog computing environment, Wireless Communications and Mobile Computing 2018 (2018) 1–11. doi:10.1155/2018/2102348.

[58] H. Zhang, R. Wang, W. Sun, H. Zhao, Mobility management for blockchain-based ultra-dense edge computing: A deep reinforcement learning approach, IEEE TWC 20 (11) (2021) 7346–7359. doi:10.1109/TWC.2021.3082986.

[59] L. Tong, Y. Li, W. Gao, A hierarchical edge cloud architecture for mobile computing, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9. doi:10.1109/INFOCOM.2016.7524340.

[60] W. Sun, H. Zhang, R. Wang, Y. Zhang, Reducing offloading latency for digital twin edge networks in 6g, IEEE Transactions on Vehicular Technology 69 (10) (2020) 12240–12251. doi:10.1109/TVT.2020.3018817.

[61] L. Zhou, J. Li, F. Li, Q. Meng, J. Li, X. Xu, Energy consumption model and energy efficiency of machine tools: a comprehensive literature review, Journal of Cleaner Production 112 (2016) 3721–3734. doi:https://doi.org/10.1016/j.jclepro.2015.05.093.
URL https://www.sciencedirect.com/science/article/pii/S0959652615006617

[62] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, P. Chatellier, Modeling and forecasting building energy consumption: A review of data-driven techniques, Sustainable Cities and Society 48 (2019) 101533. doi:https://doi.org/10.1016/j.scs.2019.101533.
URL https://www.sciencedirect.com/science/article/pii/S2210670718323862

[63] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, A. Amira, Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives, Applied Energy 287 (2021) 116601. doi:https://doi.org/10.1016/j.apenergy.2021.116601.
URL https://www.sciencedirect.com/science/article/pii/S0306261921001409

[64] A. Balali, A. Yunusa-Kaltungo, R. Edwards, A systematic review of passive energy consumption optimisation strategy selection for buildings through multiple criteria decision-making techniques, Renewable and Sustainable Energy Reviews 171 (2023) 113013. doi:https://doi.org/10.1016/j.rser.2022.113013.
URL https://www.sciencedirect.com/science/article/pii/S1364032122008942

[65] A. Alzoubi, Machine learning for intelligent energy consumption in smart homes, International Journal of Computations, Information and Manufacturing (IJCIM) 2 (1) (May 2022). doi:10.54489/ijcim.v2i1.75.
URL https://journals.gaftim.com/index.php/ijcim/article/view/75

[66] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, C. S. Lai, A multi-agent reinforcement learning-based data-driven method for home energy management, IEEE Transactions on Smart Grid 11 (4) (2020) 3201–3211. doi:10.1109/TSG.2020.2971427.

[67] B. Si, Z. Tian, X. Jin, X. Zhou, X. Shi, Ineffectiveness of optimization algorithms in building energy optimization and possible causes, Renewable Energy 134 (2019) 1295–1306. doi:https://doi.org/10.1016/j.renene.2018.09.057.
URL https://www.sciencedirect.com/science/article/pii/S0960148118311200

[68] M. Jarić, N. Budimir, M. Pejanovic, I. Svetel, A review of energy analysis simulation tools, 2013.
URL https://machinery.mas.bg.ac.rs/bitstream/handle/123456789/4384/bitstream_10476.pdf

[69] D. Crawley, C. Pedersen, L. Lawrie, F. Winkelmann, Energyplus: Energy simulation program, Ashrae Journal 42 (2000) 49–56.
URL https://simulationresearch.lbl.gov/dirpubs/46002.pdf

[70] P. Ellis, P. Torcellini, D. Crawley, Simulation of energy management systems in energyplus, 2007, pp. 1346–1353.
URL https://www.nrel.gov/docs/fy08osti/41482.pdf

[71] W. A. Beckman, L. Broman, A. Fiksel, S. A. Klein, E. Lindberg, M. Schuler, J. Thornton, Trnsys the most complete solar energy system modeling and simulation software, Renewable Energy 5 (1) (1994) 486–488, climate change Energy and the environment. doi:https://doi.org/10.1016/0960-1481(94)90420-0.
URL https://www.sciencedirect.com/science/article/pii/0960148194904200

[72] P. A. Østergaard, Reviewing energyplan simulations and performance indicator applications in energyplan simulations, Applied Energy 154 (2015) 921–933. doi:https://doi.org/10.1016/j.apenergy.2015.05.086.
URL https://www.sciencedirect.com/science/article/pii/S0306261915007199

[73] H. Lund, J. Z. Thellufsen, P. A. Østergaard, P. Sorknæs, I. R. Skov, B. V. Mathiesen, Energyplan – advanced analysis of smart energy systems, Smart Energy 1 (2021) 100007. doi:https://doi.org/10.1016/j.segy.2021.100007.
URL https://www.sciencedirect.com/science/article/pii/S2666955221000071

[74] S. R. G. L. B. N. Laboratory, Simergy–a graphical user interface for energyplus (2013).
URL https://ipo.lbl.gov/lbnl2998/

[75] R. Guglielmetti, D. Macumber, N. Long, Openstudio: An open source integrated analysis platform; preprint (12 2011).
URL https://www.osti.gov/biblio/1032670

[76] W. Ketter, J. Collins, P. Reddy, Power tac: A competitive economic simulation of the smart grid, Energy Economics 39 (2013) 262–270. doi:10.1016/j.eneco.2013.04.015.

[77] S. Gottwalt, W. Ketter, C. Block, J. Collins, C. Weinhardt, Demand side management—a simulation of household behavior under variable prices, Energy Policy 39 (12) (2011) 8163–8174, clean Cooking Fuels and Technologies in Developing Economies. doi:10.1016/j.enpol.2011.10.016.

[78] B. Nepal, M. Yamaha, A. Yokoe, T. Yamaji, Electricity load forecasting using clustering and arima model for energy management in buildings, JAPAN ARCHITECTURAL REVIEW 3 (1) (2020) 62–76. doi:10.1002/2475-8876.12135.

[79] T. Panapongpakorn, D. Banjerdpongchai, Short-term load forecast for energy management systems using time series analysis and neural network method with average true range, in: 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019, pp. 86–89. doi:10.1109/ICA-SYMP.2019.8646068.

[80] M. L. Di Silvestre, E. Riva Sanseverino, Modelling energy storage systems using fourier analysis: An application for smart grids optimal management, Applied Soft Computing 14 (2014) 469–481. doi:10.1016/j.asoc.2013.08.018.

[81] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, A. Bemporad, Model predictive control (mpc) for enhancing building and hvac system energy efficiency: Problem formulation, applications and opportunities, Energies 11 (3) (2018). doi:10.3390/en11030631.

[82] G. Bruni, S. Cordiner, V. Mulone, V. Rocco, F. Spagnolo, A study on the energy management in domestic micro-grids based on model predictive control strategies, Energy Conversion and Management 102 (2015) 50–58. doi:10.1016/j.enconman.2015.01.067.

[83] F. Zhou, Y. Li, W. Wang, C. Pan, Integrated energy management of a smart community with electric vehicle charging using scenario based stochastic model predictive control, Energy and Buildings 260 (2022) 111916. doi:10.1016/j.enbuild.2022.111916.

[84] A. Zamhuri Fuadi, I. N. Haq, E. Leksono, Support vector machine to predict electricity consumption in the energy management laboratory, Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) 5 (3) (2021) 466 – 473. doi:10.29207/resti.v5i3.2947.

[85] Z. Ma, C. Ye, W. Ma, Support vector regression for predicting building energy consumption in southern china, Energy Procedia 158 (2019) 3433–3438, innovative Solutions for Energy Transitions. doi:10.1016/j.egypro.2019.01.931.

[86] H. Zhong, J. Wang, H. Jia, Y. Mu, S. Lv, Vector field-based support vector regression for building energy consumption prediction, Applied Energy 242 (2019) 403–414. doi:10.1016/j.apenergy.2019.03.078.

[87] Z. Tian, B. Si, X. Shi, Z. Fang, An application of bayesian network approach for selecting energy efficient hvac systems, Journal of Building Engineering 25 (2019) 100796. doi:10.1016/j.jobe.2019.100796.

[88] T. Shoji, W. Hirohashi, Y. Fujimoto, Y. Amano, S. ichi Tanabe, Y. Hayashi, Personalized energy management systems for home appliances based on bayesian networks, Journal of International Council on Electrical Engineering 5 (1) (2015) 64–69. doi:10.1080/22348972.2015.1115171.

[89] M. Amayri, S. Ploix, H. Kazmi, Q.-D. Ngo, A. El Safadi, Estimating occupancy from measurements and knowledge using the bayesian network for energy management, Journal of Sensors 2019 (2019) 1–12. doi:10.1155/2019/7129872.

[90] M. Ayenew, H. Lei, X. Li, K. Kekeba, M. Assefa, A. T. Tegene, S. B. Muhammed, H. L. Leka, Data analytics and machine learning for reliable energy management: A case study, in: 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2022, pp. 1–6. doi:10.1109/ICCWAMTIP56608.2022.10016478.

[91] M. A. Ayub, H. Khan, J. Peng, Y. Liu, Consumer-driven demand-side management using k-mean clustering and integer programming in standalone renewable grid, Energies 15 (3) (2022). doi:10.3390/en15031006.

[92] S. Umetani, Y. Fukushima, H. Morita, A linear programming based heuristic algorithm for charge and discharge scheduling of electric vehicles in a building energy management system, Omega 67 (2017) 115–122. doi:10.1016/j.omega.2016.04.005.

[93] K. Bio Gassi, M. Baysal, Analysis of a linear programming-based decision-making model for microgrid energy management systems with renewable sources, International Journal of Energy Research 46 (6) (2022) 7495–7518. doi:10.1002/er.7656.

[94] J. Fedjaev, S. A. Amamra, B. Francois, Linear programming based optimization tool for day ahead energy management of a lithium-ion battery for an industrial microgrid, in: 2016 IEEE International Power Electronics and Motion Control Conference (PEMC), 2016, pp. 406–411. doi:10.1109/EPEPEMC.2016.7752032.

[95] A. C. Luna, N. L. Diaz, M. Graells, J. C. Vasquez, J. M. Guerrero, Mixed-integer-linear-programming-based energy management system for hybrid pv-wind-battery microgrids: Modeling, design, and experimental verification, IEEE Transactions on Power Electronics 32 (4) (2017) 2769–2783. doi:10.1109/TPEL.2016.2581021.

[96] M. S. Javadi, M. Gough, M. Lotfi, A. Esmaeel Nezhad, S. F. Santos, J. P. Catalão, Optimal self-scheduling of home energy management system in the presence of photovoltaic power generation and batteries, Energy 210 (2020) 118568. doi:10.1016/j.energy.2020.118568.

[97] H. Tischer, G. Verbic, Towards a smart home energy management system - a dynamic programming approach, in: 2011 IEEE PES Innovative Smart Grid Technologies, 2011, pp. 1–7. doi:10.1109/ISGT-Asia.2011.6167090.

[98] Q. Wei, D. Liu, G. Shi, Y. Liu, Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming, IEEE Transactions on Industrial Electronics 62 (7) (2015) 4203–4214. doi:10.1109/TIE.2014.2388198.

[99] C. Liu, Y. Wang, L. Wang, Z. Chen, Load-adaptive real-time energy management strategy for battery/ultracapacitor hybrid energy storage system using dynamic programming optimization, Journal of Power Sources 438 (2019) 227024. doi:10.1016/j.jpowsour.2019.227024.

[100] Z. Song, X. Guan, M. Cheng, Multi-objective optimization strategy for home energy management system including pv and battery energy storage, Energy Reports 8 (2022) 5396–5411. doi:10.1016/j.egyr.2022.04.023.

[101] B. Rajasekhar, N. Pindoriya, W. Tushar, C. Yuen, Collaborative energy management for a residential community: A non-cooperative and evolutionary approach, IEEE Transactions on Emerging Topics in Computational Intelligence 3 (3) (2019) 177–192. doi:10.1109/TETCI.2018.2865223.

[102] A. Arabali, M. Ghofrani, M. Etezadi-Amoli, M. S. Fadali, Y. Baghzouz, Genetic-algorithm-based optimization approach for energy management, IEEE Transactions on Power Delivery 28 (1) (2013) 162–170. doi:10.1109/TPWRD.2012.2219598.

[103] F.-J. Ferrández-Pastor, S. Gómez-Trillo, M. Nieto-Hidalgo, J.-M. García-Chamizo, R. Valdivieso-Sarabia, Intelligent power management system using hybrid renewable energy resources and decision tree approach, Proceedings 2 (19) (2018). doi:10.3390/proceedings2191239.

[104] M. Shcherbakov, V. Kamaev, N. Shcherbakova, Automated electric energy consumption forecasting system based on decision tree approach, IFAC Proceedings Volumes 46 (9) (2013) 1027–1032, 7th IFAC Conference on Manufacturing Modelling, Management, and Control. doi:10.3182/20130619-3-RU-3018.00486.

[105] A. Bassi, A. Shenoy, A. Sharma, H. Sigurdson, C. Glossop, J. H. Chan, Building energy consumption forecasting: A comparison of gradient boosting models, IAIT2021, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3468784.3470656.

[106] H. Lu, F. Cheng, X. Ma, G. Hu, Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower, Energy 203 (2020) 117756. doi:10.1016/j.energy.2020.117756.

[107] N. Aksoy, I. Genc, Predictive models development using gradient boosting based methods for solar power plants, Journal of Computational Science 67 (2023) 101958. doi:10.1016/j.jocs.2023.101958.

[108] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (11) (2018) e00938. doi:10.1016/j.heliyon.2018.e00938.

[109] S. Xie, X. Hu, S. Qi, K. Lang, An artificial neural network-enhanced energy management strategy for plug-in hybrid electric vehicles, Energy 163 (2018) 837–848. doi:10.1016/j.energy.2018.08.139.

[110] C. Fan, J. Wang, W. Gang, S. Li, Assessment of deep recurrent neural network-based strategies for short-term building energy predictions, Applied Energy 236 (2019) 700–710. doi:10.1016/j.apenergy.2018.12.004.

[111] Z. A. Khan, A. Ullah, I. Ul Haq, M. Hamdy, G. Maria Mauro, K. Muhammad, M. Hijji, S. W. Baik, Efficient short-term electricity load forecasting for effective energy management, Sustainable Energy Technologies and Assessments 53 (2022) 102337. doi:10.1016/j.seta.2022.102337.

[112] A. H. Ganesh, B. Xu, A review of reinforcement learning based energy management systems for electrified powertrains: Progress, challenge, and potential solution, Renewable and Sustainable Energy Reviews 154 (2022) 111833. doi:10.1016/j.rser.2021.111833.

[113] Q. Fu, Z. Han, J. Chen, Y. Lu, H. Wu, Y. Wang, Applications of reinforcement learning for building energy efficiency control: A review, Journal of Building Engineering 50 (2022) 104165. doi:10.1016/j.jobe.2022.104165.

[114] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, X. Guan, A review of deep reinforcement learning for smart building energy management, IEEE Internet of Things Journal 8 (15) (2021) 12046–12063. doi:10.1109/JIOT.2021.3078462.

[115] P. Ladosz, L. Weng, M. Kim, H. Oh, Exploration in deep reinforcement learning: A survey, Information Fusion 85 (2022) 1–22. doi:10.1016/j.inffus.2022.03.003.

[116] N. Mazyavkina, S. Sviridov, S. Ivanov, E. Burnaev, Reinforcement learning for combinatorial optimization: A survey, Computers & Operations Research 134 (2021) 105400. doi:10.1016/j.cor.2021.105400.

[117] Y. Ji, J. Wang, J. Xu, X. Fang, H. Zhang, Real-time energy management of a microgrid using deep reinforcement learning, Energies 12 (12) (2019). doi:10.3390/en12122291.

[118] W. Li, H. Cui, T. Nemeth, J. Jansen, C. Ünlübayir, Z. Wei, L. Zhang, Z. Wang, J. Ruan, H. Dai, X. Wei, D. U. Sauer, Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles, Journal of Energy Storage 36 (2021) 102355. doi:10.1016/j.est.2021.102355.

[119] L. Fan, H. Su, W. Wang, E. Zio, L. Zhang, Z. Yang, S. Peng, W. Yu, L. Zuo, J. Zhang, A systematic method for the optimization of gas supply reliability in natural gas pipeline network based on bayesian networks and deep reinforcement learning, Reliability Engineering & System Safety 225 (2022) 108613. doi:10.1016/j.ress.2022.108613.

[120] D. Bisen, U. K. Lilhore, P. Manoharan, F. Dahan, O. Mzoughi, F. Hajjej, P. Saurabh, K. Raahemifar, A hybrid deep learning model using cnn and k-mean clustering for energy efficient modelling in mobile edgeiot, Electronics 12 (6) (2023). doi:10.3390/electronics12061384.

[121] R. Kuppusamy, Y. Teekaraman, K. Kumar, Fuzzy-based energy management system with decision tree algorithm for power security system, International Journal of Computational Intelligence Systems 12 (2019) 1173. doi:10.2991/ijcis.d.191016.001.

[122] D. Rosendo, P. Silva, M. Simonin, A. Costan, G. Antoniu, E2clab: Exploring the computing continuum through repeatable, replicable and reproducible edge-to-cloud experiments, in: 2020 IEEE International Conference on Cluster Computing (CLUSTER), 2020, pp. 176–186. doi:10.1109/CLUSTER49012.2020.00028.

[123] A. Marchese, O. Tomarchio, Orchestrating serverless applications in the cloud-to-edge continuum, in: Proceedings of the 1st International Workshop on Middleware for the Computing Continuum, Mid4CC '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 12–17. doi:10.1145/3631309.3632834.
URL https://doi.org/10.1145/3631309.3632834

[124] K. Fu, W. Zhang, Q. Chen, D. Zeng, M. Guo, Adaptive resource efficient microservice deployment in cloud-edge continuum, IEEE Transactions on Parallel and Distributed Systems 33 (8) (2022) 1825–1840. doi:10.1109/TPDS.2021.3128037.

[125] Q. Luo, S. Hu, C. Li, G. Li, W. Shi, Resource scheduling in edge computing: A survey, IEEE Communications Surveys & Tutorials 23 (4) (2021) 2131–2165. doi:10.1109/COMST.2021.3106401.

[126] C. Carrión, Kubernetes scheduling: Taxonomy, ongoing issues and challenges, ACM Comput. Surv. 55 (7) (dec 2022). doi:10.1145/3539606.
URL https://doi.org/10.1145/3539606

[127] S. Wen, R. Han, K. Qiu, X. Ma, Z. Li, H. Deng, C. H. Liu, K8ssim: A simulation tool for kubernetes schedulers and its applications in scheduling algorithm optimization, Micromachines 14 (3) (2023). doi:10.3390/mi14030651.
URL https://www.mdpi.com/2072-666X/14/3/651

[128] T. Lebesbye, J. Mauro, G. Turin, I. Yu, Boreas – A Service Scheduler for Optimal Kubernetes Deployment, 2021, pp. 221–237. doi:10.1007/978-3-030-91431-8_14.

[129] A. Marchese, O. Tomarchio, Network-aware container placement in cloud-edge kubernetes clusters, in: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2022, pp. 859–865. doi:10.1109/CCGrid54584.2022.00102.

[130] A. Marchese, O. Tomarchio, Extending the kubernetes platform with network-aware scheduling capabilities, Springer-Verlag, 2022, p. 465–480. doi:10.1007/978-3-031-20984-0_33.
URL https://doi.org/10.1007/978-3-031-20984-0_33

[131] F. Rossi, V. Cardellini, F. Lo Presti, M. Nardelli, Geo-distributed efficient deployment of containers with kubernetes, Computer Communications 159 (05 2020). doi:10.1016/j.comcom.2020.04.061.

[132] S. Böhm, G. Wirtz, Towards orchestration of cloud-edge architectures with kubernetes, 2021. doi:10.1007/978-3-031-06371-8_14.

[133] A. Marletta, Cpu usage limiter for linux, Sourceforge. net (2012).

[134] M. P. Stanic, Tc–traffic control, Linux QOS Control Tool (2001).

Publicaciones

[P1] P. Gonzalez-Gil, A. Robles-Enciso, J. A. Martínez, A. F. Skarmeta, Architecture for orchestrating dynamic dnn-powered image processing tasks in edge and cloud devices, IEEE Access 9 (2021) 107137–107148. doi:10.1109/ACCESS.2021.3101306.

[P2] A. Robles-Enciso, A. F. Skarmeta, Task offloading in computing continuum using collaborative reinforcement learning, in: A. González-Vidal, A. Mohamed Abdelgawad, E. Sabir, S. Ziegler, L. Ladid (Eds.), Internet of Things, Springer International Publishing, Cham, 2022, pp. 82–95.

[P3] A. Robles-Enciso, A. F. Skarmeta, A multi-layer guided reinforcement learning-based tasks offloading in edge computing, Computer Networks 220 (2023) 109476. doi:https://doi.org/10.1016/j.comnet.2022.109476.
URL https://www.sciencedirect.com/science/article/pii/S1389128622005102

[P4] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta, Sim-powercs: An extensible and simplified open-source energy simulator, SoftwareX 23 (2023) 101467. doi:https://doi.org/10.1016/j.softx.2023.101467.
URL https://www.sciencedirect.com/science/article/pii/S2352711023001632

[P5] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta, Multi-agent reinforcement learning-based energy orchestrator for cyber-physical systems, in: I. Chatzigiannakis, I. Karydis (Eds.), Algorithmic Aspects of Cloud Computing, Springer Nature Switzerland, Cham, 2024, pp. 100–114.

[P6] A. Robles-Enciso, R. Robles-Enciso, A. F. Skarmeta Gómez, An adaptive energy orchestrator for cyberphysical systems using multiagent reinforcement learning, Smart Cities 7 (6) (2024) 3210–3240. doi:10.3390/smartcities7060125.
URL https://www.mdpi.com/2624-6511/7/6/125

[P7] A. Robles-Enciso, A. F. Skarmeta, Adapting containerized workloads for the continuum computing, IEEE Access 12 (2024) 104102–104114. doi:10.1109/ACCESS.2024.3434585.

[P8] A. Robles-Enciso, J. M. Bernabé Murcia, A. Molina Zarca, A. Skarmeta Gomez, Dynamic multi-method allocation for intent-based security orchestration, Journal

of Network and Systems Management 33 (1) (2024) 1–28. doi:10.1007/s10922-024-09896-8.
URL https://doi.org/10.1007/s10922-024-09896-8