future internet

Article

Holistic Utility Satisfaction in Cloud Data Centre Network Using Reinforcement Learning

Pejman Goudarzi [1,*,†], Mehdi Hosseinpour [1,†], Roham Goudarzi [2,†] and Jaime Lloret [3,*,†]

[1] ICT Research Institute (ITRC), Tehran 14155-3961, Iran
[2] Faculty of Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[3] Department of Communications, Polytechnic University of Valencia, Camino de Vera, 46022 Valencia, Spain
* Correspondence: pgoudarzi@itrc.ac.ir (P.G.); jlloret@dcom.upv.es (J.L.)
† These authors contributed equally to this work.

Abstract: Cloud computing leads to efficient resource allocation for network users. In order to achieve efficient allocation, many research activities have been conducted so far. Some researchers focus on classical optimisation theory techniques (such as multi-objective optimisation, evolutionary optimisation, game theory, etc.) to satisfy network providers and network users' service-level agreement (SLA) requirements. Normally, in a cloud data centre network (CDCN), it is difficult to jointly satisfy both the cloud provider and cloud customer′ utilities, and this leads to complex combinatorial problems, which are usually NP-hard. Recently, machine learning and artificial intelligence techniques have received much attention from the networking community because of their capability to solve complicated networking problems. In the current work, at first, the holistic utility satisfaction for the cloud data centre provider and customers is formulated as a reinforcement learning (RL) problem with a specific reward function, which is a convex summation of users' utility functions and cloud provider's utility. The user utility functions are modelled as a function of cloud virtualised resources (such as storage, CPU, RAM), connection bandwidth, and also, the network-based expected packet loss and round-trip time factors associated with the cloud users. The cloud provider utility function is modelled as a function of resource prices and energy dissipation costs. Afterwards, a Q-learning implementation of the mentioned RL algorithm is introduced, which is able to converge to the optimal solution in an online and fast manner. The simulation results exhibit the enhanced convergence speed and computational complexity properties of the proposed method in comparison with similar approaches from the joint cloud customer/provider utility satisfaction perspective. To evaluate the scalability property of the proposed method, the results are also repeated for different cloud user population scenarios (small, medium, and large).

Keywords: CDCN; QoS; VM; reinforcement learning; resource assignment

check for updates

Citation: Goudarzi, P.; Hosseinpour, M.; Goudarzi, R.; Lloret, J. Holistic Utility Satisfaction in Cloud Data Centre Network Using Reinforcement Learning. *Future Internet* **2022**, *14*, 368. https://doi.org/10.3390/fi14120368

Academic Editor: Ping Wang

Received: 16 November 2022
Accepted: 6 December 2022
Published: 8 December 2022

1. Introduction

Cloud computing is implemented by a system of distributed computers with virtualised resources in data networks. The virtualised resources of data centres consist of four essential components, which are the CPU, RAM, storage, and bandwidth. Cloud service providers deliver their service to customers through a cloud data centre network (CDCN). The CDCN consists of a number of cloud data centres (CDCs), which are connected through networking facilities in order to guarantee the specified service-level agreements (SLAs) for cloud users [1] or perform flexible big data management [2].

Different applications need different quality of service levels in terms of CPU cycles, storage capacity, RAM level, and access bandwidth. A utility function can be described for each network user, and this function expresses the SLA satisfaction level, in a numerical manner [3].

On the other side, the cloud provider also has its own utility function in terms of energy dissipation costs and other related parameters [3]. The cloud data centre provider/manager allocates the cloud resources to cloud users dynamically.

Machine learning (ML) techniques are very important for tackling scientific challenges in many fields such as agriculture, finance, automation, health, industry, etc. In fact, ML can be regarded as a sub-field of artificial intelligence (AI) [4]. An ML system, at first, makes observations of its surrounding environment and, then, improves its performance and efficiency for future tasks [5]. These observations are supported by data, and the sensors are the primary sources of the data. After analysing the mentioned input data, an ML algorithm generates and possibly updates a model from its surrounding/operational environment. From the communication system perspective, ML techniques analyse the traffic data and extract useful information/knowledge from them, which may not be available to humans by themselves [6]. Machine learning algorithms can be divided into four specific subtypes, which are: supervised learning [5], unsupervised learning [7,8], semi-supervised learning [9], and reinforcement learning [10,11].

In supervised learning algorithms, a function is learned that maps an input to an output based on example input–output pairs. After being trained on the training data, a supervised learning algorithm produces a function model. Then, the new examples can be mapped by this model [5]. If the data samples are not labelled or classified, unsupervised learning methods must be adopted for data pattern extraction. In another word, patterns that exist and are hidden within datasets can be identified by unsupervised learning algorithms [12–14]. In the machine learning literature, there exists an intermediate reasoning method, which is called reinforcement learning (RL). RL is a machine learning sub-type, which is model-free. It uses some agents, which interact with an unknown environment. The objective of this type of algorithm is to optimise the cumulative and long-term reward [9]. In RL methods, given the input data, the algorithm learns to take actions that maximise a cumulative reward [10,11].

All these types of learning empower the systems with the required intelligence to enhance their future performance based on current information/data. Communication networks may use ML for different use cases [15]. The major reasons for using ML are creating the required robustness or adaptivity in confronting the changing and normally unpredictable user, traffic, and network conditions [16,17].

Based on [18], virtualisation technology is one of the most-critical elements in cloud computing, which allows a physical resource to be virtually shared amongst different network users. Virtualisation can speed up the IT operations, and by optimising the use of the networking infrastructure, it reduces the operational costs.

In a cloud computing system, the job scheduling is an essential part. The job scheduling system allocates the required resources (in terms of virtual machines) to dynamically arriving workflows [19]. Normally, the jobs are performed in a priority-based manner. Jobs with a higher priority are scheduled to be performed prior to those with a lower priority level.

Most of the cloud providers consider only the cloud provider benefits and constraints in satisfying the user SLAs [20]. As the users' demands are different, cloud providers normally use some admission control policies to allocate resources to higher-priority users. This makes the lower-priority users unsatisfied, and this leads to customer churn [21].

Cloud providers must try to guarantee quality of service (QoS) parameters (such as transfer delay, response time, etc.) for satisfying their cloud users [22,23].

Cloud-based resource assignment in cloud computing has been addressed in many previous works such as [24–30]. In contrast to the work presented [30–32], in this paper, we focus on the problem of simultaneous SLA satisfaction for the whole CDCN ecosystem (including customers and provider). As different players may use different cloud-based applications (with different QoS requirements) and finding a solution strategy for joint SLA satisfaction for all of these users is very difficult due to the non-linear and time-dependent nature of the resulting utility functions, we used an intelligent reinforcement learning approach [33] for tackling the resulting complex and non-linear resource allocation/assignment problem.

The most-important contributions of the current work are as follows:

- Utility modelling was adopted for the overall CDCN system, in terms of virtualising the networking resources, which are: CPU, RAM, storage, and access/connection bandwidth.
- The energy consumption model of the cloud data centres is described and used in the utility modelling of the cloud manager.
- After modelling the utility/satisfaction functions associated with the cloud users/provider, a reinforcement learning sub-type (which is Q-learning) was adopted for optimised resource assignment to different cloud users, which also simultaneously satisfies the requirements of the CDCN manager/provider in terms of energy efficiency in a holistic manner.
- Finally, the online and model-free property of the Q-learning algorithm results in converging to optimal utility levels for both the cloud users and cloud provider in different cloud user population scenarios in a fast and low-complexity manner.

In this paper, an RL-based optimisation approach based on the Q-learning algorithm [34] was developed. The cloud provider of the CDNC system must deploy this algorithm and, based on that, assign optimal VM resources (CPU, storage, and RAM) and, also, the connection bandwidth to active cloud customers [33,35] in order to simultaneously optimise cloud user/provider utilities.

The rest of the paper is organised as follows. In Section 2, we review the related state-of-the-art. In Section 3, we present two utility models for describing the SLA satisfaction of CDCN users and the CDCN provider/manager. In Section 4, the Q-learning-based resource allocation methodology is developed and customised for optimal assignment of the cloud's resources to its customers, subject to multiple constraints. Section 5 gives the experimental analysis results. At the end, in Section 6, we introduce some open research areas and some important concluding remarks.

2. Related Work

In the sequel, we describe some viewpoints regarding resource allocation in the cloud environments.

In [3], the cloud data centre network was first modelled under the energy dissipation constraints. Then, the authors used the cooperative co-evolution evolutionary algorithms (CCEAs) for solving the resulting complex problem..

The authors in [36] addressed the challenge of virtual machine scheduling in cloud environments by presenting a new model that can analyse the distribution of the server CPU and airflow temperatures. They proposed GRANITE—a holistic scheduling method for optimising the total power dissipation of the data centre (DC).

The researchers in [37] addressed the problem of virtual machine scheduling in cloud environments by introducing a discrete-time Markov chain model to predict future resource usage.

In [38], Zhang et al. introduced a resource allocation methodology for reducing the joint SLA violations and power usage in cloud-based data centres. Their proposed algorithm comprises three sub-algorithms, which are improved for consolidating virtual machines in a dynamic manner.

A machine-learning-based fast and accurate thermal prediction model was proposed in [39] to aid the resource management system's online decision. The author also proposed an energy-efficient VM scheduling algorithm to minimise peak temperature in the data centre.

In [40], Gill et al. investigated a bio-inspired resource allocation algorithm (cuckoo search), which is aware of the reliability of energy consumption for controlling cloud resources including networks, storage, cooling infrastructures, and servers.

A. Heimerson et al. in [41] developed a simple model to represent the heat rejection system and energy usage in a small DC setup. The cooling system parameters (setpoints), IT workload, and load balancing features of the model are managed by an agent, which uses RL-based machine learning techniques. Their main contribution was a holistic approach for data centre management. The inputs of their proposed method were the logs associated with cloud services and metrics related to both the IT hardware and facility.

In [42], the authors addressed the load balancing problem under the constraint of achieving the minimum latency in fog networks. To solve the mentioned problem, the authors proposed a decision-making methodology, which is based on RL techniques. The mentioned methodology can obtain the optimum offloading strategy under the constraints in which both the activation/transition and reward functions are unknown.

The authors in [43] tried to improve multi-data centre systems using machine learning. They considered energy efficiency and tried to minimise the usage pattern of the cloud while maintaining the required user QoS levels. Rebai et al. used cloud federation for profit maximisation for the joint satisfaction of user requirements and networking demands [22].

In [32], the authors proposed a resource assignment strategy according to the cooperative game and principle of uncertainty for better cloud resource utilisation and end user satisfaction. In the paper [44], the authors proposed the DCloud concept for time sliding and bandwidth scaling to increase the cloud manager benefits and reduce the costs associated with different tenants. The work in [45] is a survey paper about resource assignment in the cloud environment and its associated challenges.

The authors in [46] proposed a tier-centric optimal cloud resource assignment methodology for addressing the problem of early and fast provisioning of information-technology-related resources in enterprise-level systems. The authors in [47] used bio-inspired methods for virtual machine allocation to demanding cloud users.

By considering the fairness among cloud users and the cloud system's resource utilisation, the authors in [48] developed a fair and efficient cloud resource assignment strategy based on game theory.

In [28], a dynamic bin-packing (DBP) methodology was adopted by Li et al. for cloud resource assignment. In [3], the authors used the cooperative co-evolution evolutionary algorithm, which is a multi-objective optimisation methodology, for solving the joint user–provider utility satisfaction problem in cloud data centre networks.

Multi-objective optimisation methods have been used by researchers such as [49] and [50] for addressing some cloud challenges such as high power consumption, virtual machine consolidation, and under-utilisation of resources.

In [51], Shaw et al. used reinforcement learning techniques for automating energy-efficient virtual machine consolidation in cloud data centres. In [52], Lin et al. proposed a time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. In [53], Lopez et al. developed a shallow neural network with kernel approximation for prediction problems in data communication networks.

Policy gradient methods, proximal policy optimisations, and imitation learning have also been used for cloud resource management [54–56]. In [54], the authors used reinforcement learning heuristics for efficiently scheduling data processing jobs on distributed compute clusters, which require complex algorithms. In [55], the authors proposed a framework for optimising the trade-off between QoS and energy consumption using reinforcement learning approaches. Guo et al. in [56] used a convolutional neural network to capture the cloud resource management model and utilise imitation learning in the reinforcement process to reduce the training time of the optimal policy. In contrast with all of the mentioned work, in the current paper, we have a holistic view of the cloud resource allocation optimisation problem by taking into account the benefits of cloud users (in terms of required cloud resources, expected network packet loss, and expected round-trip time) and the cloud provider (in terms of allocated resource prices and energy consumption costs) simultaneously. On the other hand, the Q-learning methodology was selected in the current work because it is an online and low-complexity algorithm, which can be tailored in a proper manner to tackle the time-varying aspects of the resource allocation in cloud data centre network systems.

In comparison with [3], in the current work, a more accurate user utility model was developed. This model incorporates the stochastic nature of packet loss and round-trip time (RTT) in calculating cloud user utility functions. Moreover, in all of the above papers (except [3]), cloud resource management was deployed by different strategies in order to satisfy multiple objectives, which target the cloud user side utility and cloud provider side utility in a non-holistic manner. In the current work, a joint resource scheduling strategy was developed that can enhance the resource usage in order to simultaneously satisfy the CDCN ecosystem and, in parallel, can consider energy saving constraints. In doing so, we selected online ML techniques in the form of RL-based optimisation. An agent (cloud provider) interacts with the environment (cloud resources and users) in an action–reward form in order to converge to an optimal resource assignment strategy in each time slot. According to our best knowledge, this is the first time that RL techniques have been employed in the resource allocation of cloud data centre networks for joint cloud provider/cloud user satisfaction considering energy constraints.

In Table 1, we classify the mentioned cloud resource allocation strategies into different categories.

Table 1. Classification of different cloud data centre resource allocation strategies.

Strategy	References
Multi-disciplinary	[2,19–22,24,28–30] [36,38,40,44,46,52,57]
ML-based	[39,41–43,51,53–56]
Game-theory-based	[26,32,48]
Evolutionary/multi-objective-based	[3,37,47,49,50]

3. Modelling of the CDCN

By definition, a CDC is composed of three essential elements, which are the cloud manager, virtualised data centres, and cloud users, as depicted in Figure 1. As shown in Figure 1, a cloud data centre network is composed of multiple physical data centres, which are under the control of a cloud manager/provider. Each CDC consists of a resource pool, which is comprised of VMs associated with the CPU cores, storage capacity, and RAM.

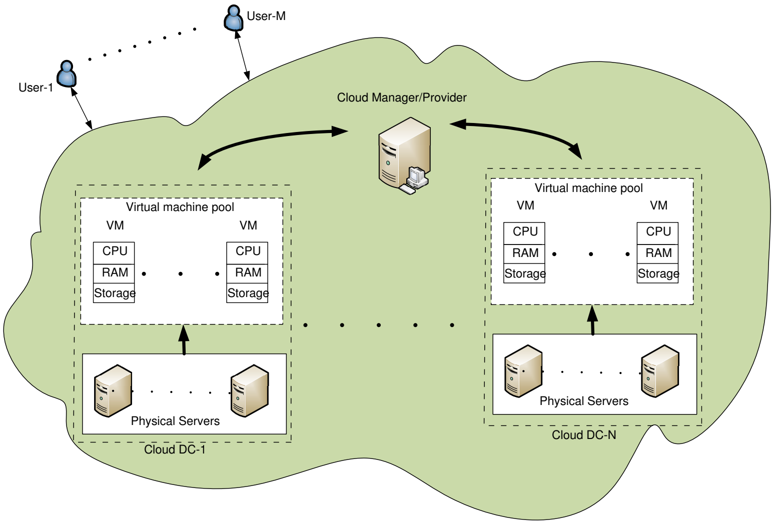


Figure 1. VM resource pools (CPU, RAM, etc.) in a typical cloud data centre network managed by a cloud provider.

For a clearer demonstration of the parameters that are used in this section, we include the system parameters in Table 2.

Table 2. System parameters.

Sets and Indices	Description
$\mathcal{S}(t)$ and $\mathcal{Z}^{\ell}(t)$	State and action spaces for time slot t and iteration ℓ
i, j	Indices of cloud users and cloud data centres
ℓ, L	Iteration number and number of training episodes in Q-learning
System Parameters	**Meaning**
t and τ	time slot/frame parameter and time frame period
$\Gamma, \gamma, \delta, \zeta, k, \alpha$	Some positive constants
Variables	**Description**
$N(t)$, $M(t)$	Number of CDCs and cloud users for time t
$\mathcal{C}^{(i)}(t)$,$\mathcal{D}^{(i)}(t)$, $\mathcal{G}^{(i)}(t)$,$\mathcal{B}^{(i)}(t)$	Total assigned CDCN resources to user i at t
$c_j^{(i)}(t)$,$d_j^{(i)}(t)$, $g_j^{(i)}(t)$,$b_j^{(i)}(t)$	assigned CDC j resources assigned to user i at t
$w_1(t)$,$w_2(t)$, $w_3(t)$,$w_4(t)$	Relative resource prices at time t
$w_3(t)$,$w_4(t)$	Unit time per unit resource price of each storage and bandwidth unit at time t
$\xi_j(t)$	Energy dissipation price for CDC j at time t
$\beta_1(t)$,$\beta_2(t)$,$\beta_3(t)$	Positive cloud energy consumption parameters at time slot t
$\mathcal{U}^{(i)}(t)$	Cloud user/customer utility function
$\mathcal{U}^{(CP)}(t)$	Utility of cloud provider at slot t
$C^{CP}(t)$,$D^{CP}(t)$, $G^{CP}(t)$,$B^{CP}(t)$	Existing CP resource pool at time t

We assumed that time is slotted/framed and can be described by $t = 0, 1, 2, \cdots$. The time slot duration/period is considered to be τ. We also assumed that, in the cloud data centre environment, the number of active CDCN users is denoted by $M(t)$, and the CDCN is comprised of $N(t)$ physical and distinct CDCs at time frame t.

3.1. The Utility Function Model for Cloud Users

Let us assume that a specific cloud customer i is running a specific application at time frame t, and the successful and quality-guaranteed execution of its application requires in total $\mathcal{C}^{(i)}(t)$ CPU core, $\mathcal{D}^{(i)}(t)$ RAM blocks, $\mathcal{G}^{(i)}(t)$ storage capacity, and connection bandwidth $\mathcal{B}^{(i)}(t)$ from the CP's resource scheduler. The CPU, storage capacity, RAM level, and connection bandwidth dedicated by DC_j to cloud customer i at time t are denoted by $c_j^{(i)}(t) \geq 0, g_j^{(i)}(t) \geq 0$, $d_j^{(i)}(t) \geq 0$, and $b_j^{(i)}(t) \geq 0$, respectively. Therefore, according to Equation (1), we have $\forall i, t$:

$$\mathcal{C}^{(i)}(t) = \sum_{j=1}^{N(t)} c_j^{(i)}(t), \qquad \mathcal{D}^{(i)}(t) = \sum_{j=1}^{N(t)} d_j^{(i)}(t),$$
$$\mathcal{G}^{(i)}(t) = \sum_{j=1}^{N(t)} g_j^{(i)}(t), \qquad \mathcal{B}^{(i)}(t) = \sum_{j=1}^{N(t)} b_j^{(i)}(t) \tag{1}$$

Any application type of cloud user i (such as FTP, video streaming, gaming, etc.) must be quality-guaranteed with an SLA, which is provided by the cloud manager/provider. It is assumed that cloud resource assignment is elastic. The experienced utility/satisfaction level of user i is a function of its assigned CDCN VMs including RAM, storage capacity, CPU, and also, connection bandwidth, total expected packet loss, and worst-case expected round-trip time. (In some scenarios, such as cloud gaming, the rendered video in the cloud must be delivered to the user by guaranteeing some packet loss ratio and delay thresholds. If these thresholds are not satisfied, the user's perceived quality of experience (QoE) can be substantially deteriorated.) For high-quality delivery of each network application, a minimum level of cloud data centre resources must be guaranteed. One of the best functional forms that can meet the mentioned requirements is the product logistic function. The product logistic form for modelling energy-efficient data centres has been used in previous work such as [58]. Furthermore, the proposed product logistic model is in fact a non-linear extension of the utility model previously proposed in [59]). Therefore, we selected this function for modelling the user utility/satisfaction for each cloud user $i = 1, 2, \cdots, M(t)$ in Equation (2):

$$\begin{aligned}\mathcal{U}^{(i)}(t) = 1 + \mathcal{U}_{max}\Bigg\{ &\left(1 + k_1^{(i)}\mathbf{exp}\left(-\alpha_1^{(i)}\mathcal{C}^{(i)}(t)\right)\right) \times \left(1 + k_2^{(i)}\mathbf{exp}\left(-\alpha_2^{(i)}\mathcal{D}^{(i)}(t)\right)\right) \times \\ &\left(1 + k_3^{(i)}\mathbf{exp}\left(-\alpha_3^{(i)}\mathcal{G}^{(i)}(t)\right)\right) \times \left(1 + k_4^{(i)}\mathbf{exp}\left(-\alpha_4^{(i)}\mathcal{B}^{(i)}(t)\right)\right)\Bigg\}^{-1} \\ &\times\mathbf{exp}\left(-\alpha_5^{(i)}\mathbb{E}[\mathcal{L}^{(i)}(t)] - \alpha_6^{(i)}\mathbb{E}[\mathcal{H}^{(i)}(t)]\right)\end{aligned} \tag{2}$$

where $\mathcal{U}_{max}$ is the maximal level of SLA satisfaction (based on [60], $\mathcal{U}_{max}$ was selected to be four). $k^{(i)} > 0$ and $\alpha^{(i)} > 0$ are the parameters associated with the logistic function. These parameters represent the steepness and horizontal drift associated with the logistic function. Therefore, these parameters must be selected based on the application types and characteristics for each user i.

$\mathbb{E}[\mathcal{L}^{(i)}(t)]$ and $\mathbb{E}[\mathcal{H}^{(i)}(t)]$ are the expected total packet loss ratio and round-trip time (RTT) associated with user i at time slot t and can be described as Equation (3) [23]:

$$\mathcal{L}^{(i)}(t) = 1 - \prod_{j \in \mathcal{N} \setminus \mathcal{N}^{-i}(t)} (1 - \mathcal{L}_j^{(i)}(t)) \tag{3}$$

where $\mathcal{L}_j^{(i)}(t)$ is the packet loss ratio associated with the communication path between user i and data centre j at time slot t. $\mathcal{N}$ and $\mathcal{N}^{-i}(t)$ are the set of all data centres and the subset of those data centres that do not assign any resource to user i at each time frame t, respectively.

If we assume a *Poisson* distribution of known mean rate parameter $\lambda_j^{(i)}$ for the packet loss process of each communication link between user i and data centre j [61] and if we assume that the loss processes in each path are statistically independent, we can write the expected total packet loss for user i as Equation (4):

$$\mathbb{E}[\mathcal{L}^{(i)}(t)] = 1 - \prod_{j\in\mathcal{N}\setminus\mathcal{N}^{-i}(t)} (1 - \mathbb{E}[\mathcal{L}_j^{(i)}(t)]) = 1 - \prod_{j\in\mathcal{N}\setminus\mathcal{N}^{-i}(t)} (1 - \lambda_j^{(i)}) \tag{4}$$

We assumed that the round-trip times associated with multiple communication paths between user i and the data centres is dominated by the largest one. Therefore, we can write, according to Equation (5):

$$\mathcal{H}^{(i)}(t) = \sup_{j\in\mathcal{N}\setminus\mathcal{N}^{-i}(t)} \{\mathcal{H}_j^{(i)}(t)\} \tag{5}$$

where $\mathcal{H}_j^{(i)}(t)$ is the measured round-trip time associated with the communication path between user i and data centre j at each time frame t.

If we also assume that the round-trip time processes for each communication path are statistically independent and have *truncated normal* distributions with parameter vector (μ, σ, x, y), we can write, according to Equation (6) [62]:

$$\mathbb{E}[\mathcal{H}^{(i)}(t)] = \mu_{\text{sup}}^{(i)} + \sigma_{\text{sup}}^{(i)}\sqrt{\frac{2}{\pi}} \times \frac{\mathbf{exp}\{-\frac{1}{2}(x_{\text{sup}}^{(i)})^2\} - \mathbf{exp}\{-\frac{1}{2}(y_{\text{sup}}^{(i)})^2\}}{\mathbf{erf}(\frac{y_{\text{sup}}^{(i)}}{\sqrt{2}}) - \mathbf{erf}(\frac{y_{\text{sup}}^{(i)}}{\sqrt{2}})} \tag{6}$$

where $\mathbf{erf}(\cdot)$ is the *error function* and $\mu_{\text{sup}}^{(i)}$ and $\sigma_{\text{sup}}^{(i)}$ are the mean and standard deviation of the normal distribution associated with the largest round-trip time path and $x_{\text{sup}}^{(i)}$ and $y_{\text{sup}}^{(i)}$ are their corresponding positive truncation parameters, and we have $x_{\text{sup}}^{(i)} < y_{\text{sup}}^{(i)}$.

For a clearer illustration of user utility modelling, in Figure 2, the user utility function calculation is summarised. As can be verified in Figure 2, for calculating each user utility function, we need the cloud data centres' resource pool info (CPU, RAM, storage, BW) and expected packet loss and round-trip times associated with each user session.

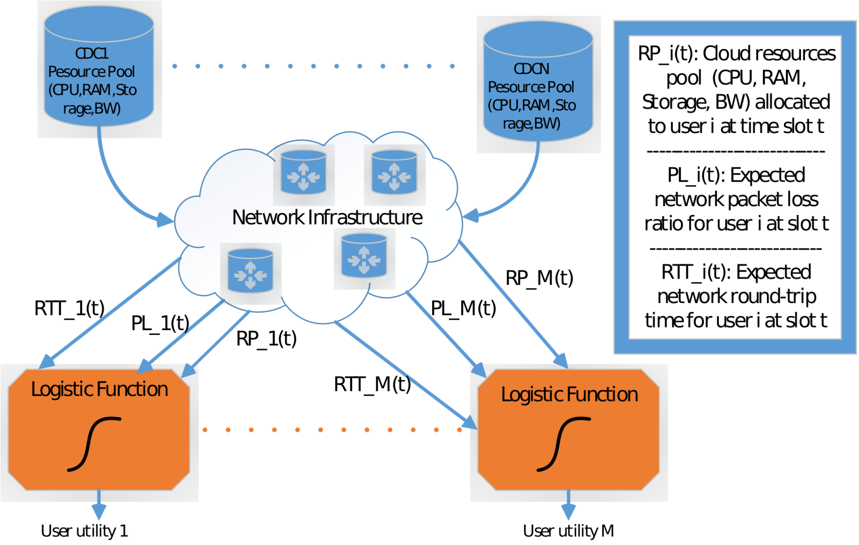

CDC1 Pesource Pool (CPU,RAM,Sto rage,BW)
CDCN Pesource Pool (CPU,RAM,Sto rage,BW)
RP_i(t): Cloud resources pool (CPU, RAM, Storage, BW) allocated to user i at time slot t
PL_i(t): Expected network packet loss ratio for user i at slot t
RTT_i(t): Expected network round-trip time for user i at slot t
Network Infrastructure
RTT_1(t)
PL_1(t)
RP_1(t)
RTT_M(t)
PL_M(t)
RP_M(t)
Logistic Function
Logistic Function
User utility 1
User utility M

Figure 2. Flow diagram of calculating user utility functions.

3.2. *The Utility Modelling for Cloud Data Centre Provider*

The utility function of the cloud data centre manager/provider has a close relationship with the resource usage and energy consumption costs. If we represent the cloud-assigned resource price for user i for each time frame t with $\mathcal{I}^{(i)}(t)$ and energy consumption cost of CDC j at each time frame t by $\xi_j(t)$, the utility function associated with cloud provider can be modelled as described in Equation (7):

$$\mathcal{U}^{(CP)}(t) = \mathcal{U}^{(CP)}_{max}\left(1 + k_7\mathbf{exp}\left(-\alpha_7\Big(\sum_{i=1}^{M(t)}\mathcal{I}^{(i)}(t) - \sum_{j=1}^{N(t)}\xi_j(t)\Big)\right)\right)^{-1} \tag{7}$$

$\mathcal{U}^{(CP)}_{max}$ is the maximal cloud-side utility. $\mathcal{I}^{(i)}(t)$ is in fact a resource pricing metric for user i. It is a weighted summation of allocated resources (RAM, CPU, connection bandwidth, and storage capacity) of the CDCN to a specific cloud customer i for time frame t, as described in Equation (8):

$$\mathcal{I}^{(i)}(t) = \tau\sum_{j=1}^{N(t)}\left(w_1(t)c_j^{(i)}(t) + w_2(t)d_j^{(i)}(t) + w_3(t)g_j^{(i)}(t) + w_4(t)b_j^{(i)}(t)\right) \tag{8}$$

For the description of the parameters w_i, please refer to Table 2.

Similar to [57], the energy dissipation cost for CDC j for time frame period τ can be represented as $\forall j = 1, 2, \cdots, N(t)$, according to Equation (9):

$$\xi_j(t) = \tau\left(\beta_{1j}(t)\sum_{i=1}^{M(t)}c_j^{(i)}(t) + \beta_{2j}(t)\sum_{i=1}^{M(t)}d_j^{(i)}(t) + \beta_{3j}(t)\sum_{i=1}^{M(t)}g_j^{(i)}(t)\right) \tag{9}$$

where $\beta_{1j}(t)$, $\beta_{2j}(t)$, and $\beta_{3j}(t)$ represent the amount of the relative per-unit price of the CPU, RAM, and storage, which is allocated from CDC j to all active cloud users at time slot t, respectively.

The schematic diagram associated with the cloud provider utility calculation process is depicted in Figure 3. As can be seen, for calculating the cloud provider's utility function, the information regarding the cloud data centres' energy dissipation and resource prices associated with cloud users is needed.

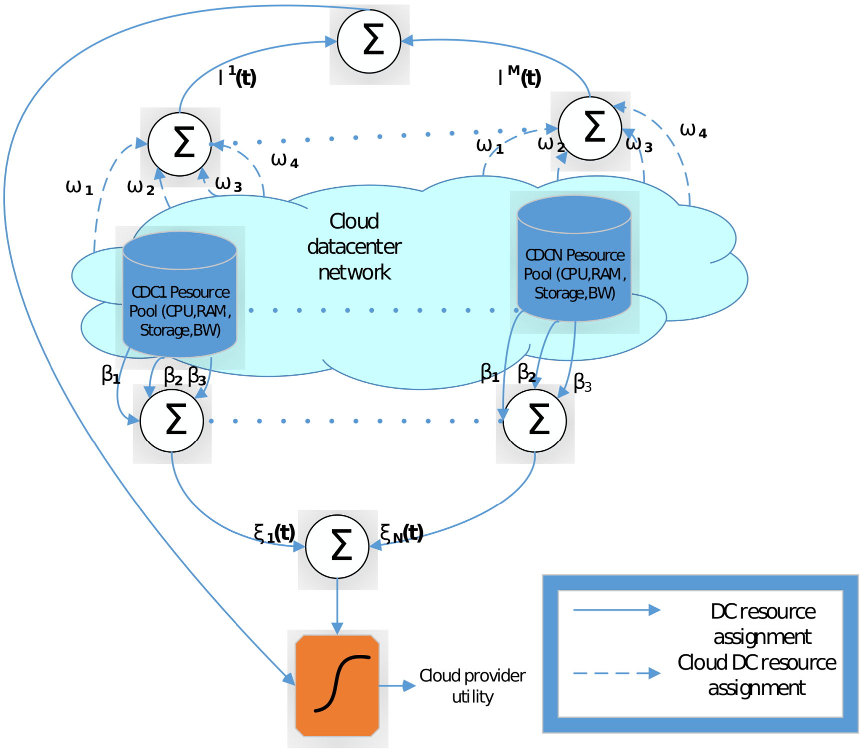

Σ
I 1(t)
I M(t)
ω1
ω2
ω3
ω4
Cloud datacenter network
CDC1 Pesource Pool (CPU,RAM, Storage,BW)
CDCN Pesource Pool (CPU,RAM, Storage,BW)
β1
β2
β3
ξ1(t)
ξN(t)
Cloud provider utility
DC resource assignment
Cloud DC resource assignment

Figure 3. Schematic diagram of calculating cloud provider utility function.

4. Reinforcement-Learning-Based Cloud Resource Allocation

In the current section, we describe two different components of the proposed RL-based algorithm. In the first part, the ML-based Q-learning model is introduced and related to the components of the cloud data centre system. In the second part, the proposed Q-learning-based resource allocation in the cloud data centre network algorithm (named QLRA) is developed and each of its components matched to the elements of the proposed holistic cloud provider/customer satisfaction problem in Section 3.

4.1. Q-Learning Model

In principle, in Q-learning, as a typical reinforcement learning method, the main objective is that a learning agent finds an optimum strategy for maximising a weighted and cumulative reward function by some repeated actions imposed on an unknown system [63,64]. In Q-learning, four essential elements exist, which are the *agent*, *state*, *action*, and *reward*, respectively. In Figure 4, the interactions between the agent (CDCN manager or CP in this case) and the environment (cloud CDCN system) are depicted. In each period/epoch, the Q-learning algorithm executes itself in the following manner. First, the learning agent observes the state $\mathbf{S}_t^\ell$ of the environment at iteration ℓ of time t. Then, the learning agent chooses a corresponding action $\mathbf{Z}_t^\ell$ among all possible actions. The environment gives a corresponding reward $\mathbf{R}_t^\ell$ for the action $\mathbf{Z}_t^\ell$. The learning agent works in two distinct phases, which are *exploration* and *exploitation*.

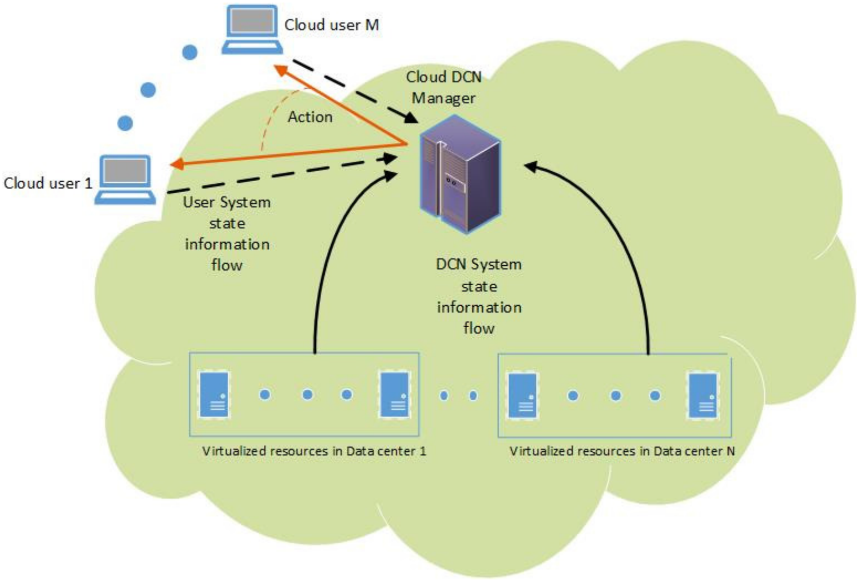

Cloud user M
Cloud DCN Manager
Action
Cloud user 1
User System state information flow
DCN System state information flow
Virtualized resources in Data center 1
Virtualized resources in Data center N

Figure 4. Mapping the elements of the proposed Q-learning algorithm to the CDCN.

In the exploitation phase, the learning agent selects an action with the largest reward value based on the past experiences. In the exploration phase, a random action in action set $\mathbf{Z}$ is normally selected. The ultimate goal is to identify the optimal action–state pair that can maximise the long-term reward. Hereafter, four elements of the proposed Q-learning methodology in the cloud DCN environment are described in more detail.

State and environment: In the cloud data centre network system, the environment is composed of different virtualised data centres with the associated resource pools and cloud users. The state is a vector that consists of the current available and free virtualised resources (CPU core, RAM, storage capacity, and connection bandwidth) of each data centre. The state can be represented as follows:

$\mathbf{S}_t^\ell$ is a $1 \times (6M(t))$ vector and is defined as $\mathbf{S}_t^\ell = [\mathcal{C}_\ell^{(1)}(t), \mathcal{D}_\ell^{(1)}(t), \mathcal{G}_\ell^{(1)}(t), \mathcal{B}_\ell^{(1)}(t),$ $\mathbf{E}[\mathcal{L}_\ell^{(1)}(t)], \mathbf{E}[\mathcal{H}_\ell^{(1)}(t)], \cdots, \mathcal{C}_\ell^{(M(t))}(t), \mathcal{D}_\ell^{(M(t))}(t), \mathcal{G}_\ell^{(M(t))}(t), \mathcal{B}_\ell^{(M(t))}(t), \mathbf{E}[\mathcal{L}_\ell^{(M(t))}(t)],$ $\mathbf{E}[\mathcal{H}_\ell^{(M(t))}(t)]]$, in iteration ℓ in time slot t.

Based on Equation (10), we also have the following four capacity constraints:

$$\sum_{k=1}^{M(t)} \mathcal{C}_{\ell}^{(k)}(t) \leq C^{CP}(t), (\mathbf{C}_1); \qquad \sum_{k=1}^{M(t)} \mathcal{D}_{\ell}^{(k)}(t) \leq D^{CP}(t), (\mathbf{C}_2)$$
$$\sum_{k=1}^{M(t)} \mathcal{G}_{\ell}^{(k)}(t) \leq G^{CP}(t), (\mathbf{C}_3); \qquad \sum_{k=1}^{M(t)} \mathcal{B}_{\ell}^{(k)}(t) \leq B^{CP}(t), (\mathbf{C}_4) \tag{10}$$

where $C^{CP}(t), D^{CP}(t)$, $G^{CP}(t)$, and $B^{CP}(t)$ are the existing total CPU amount, storage capacity, RAM, and connection bandwidth for the cloud provider for each time frame t.

The state space $\mathcal{S}(t)$ is a convex polytope, as described in Equation (11):

$$\mathcal{S}(t) = \prod_{k=1}^{6\times M(t)} [0, U_k(t)] \bigcap_{j=1}^{4} \mathbf{C}_j \tag{11}$$

where the upper limit for each state variable k is denoted by $U_k(t)$ for each time frame t and $\mathbf{C}_1, \cdots, \mathbf{C}_4$ are the corresponding constraint sets in Equation (10), respectively.

It must be mentioned that the state expression in Equation (11) has the potential for increasing the number of states exponentially. In addition, in a real data centre setting, we have four variables, which are the VM-based variables (CPU, RAM, storage) and connection bandwidth. Furthermore, two other independent variables are associated with each data centre user (expected RTT and expected packet loss), which results in six variable for each user. If we assume that 1000 simultaneous users want to use the CDCN resources, the state space has a large dimension of 6000, which makes the resource allocation problem have a large computational complexity. In such situations, the ordinary Q-learning algorithm may not be a good candidate, and a deep RL mechanism (such as DQN) should be adopted.

Agent: The agent in this paper is considered to be the cloud manager/provider (CP). At first, the CP receives the state information including the requested users' demand quadruples (CPU, RAM, storage, bandwidth) and available resources from cloud data centres in each time slot t (see Figure 4), then the CP assigns some resources to each cloud user based on the proposed reinforcement-learning-based algorithm.

Action: The action is the vector of virtualised resources that are allocated to each distinct cloud user in state ℓ for each time slot t and represented by $\mathbf{Z}_t^{\ell}$. The action space $\mathcal{Z}^{\ell}(t)$ depends on the current state $\mathbf{S}_t^{\ell}$ and is in fact a reverse-shifted version of state space $\mathcal{S}(t)$ (see Figure 5 for a simple two-dimensional example).

In Figure 5, a sample state space for a two-dimensional space with sample state variables x_1 and x_2 $(0 \leq x_1 \leq a, 0 \leq x_2 \leq b)$ and the constraint $(x_1 + x_2 \leq c)$ is depicted. As can be verified in this figure, the action space is a function of the current state $\mathbf{S}_t^{\ell}$.

Reward: The reward is the weighted sum of all cloud user utility functions and the cloud provider/manager utility function as: $\mathbf{R}_t^{\ell} \triangleq \Gamma \sum_{i=1}^{M(t)} \mathcal{U}^i(t) + (1-\Gamma)\mathcal{U}^{CP}(t)$, where Γ is a positive constant, $0 < \Gamma < 1$.

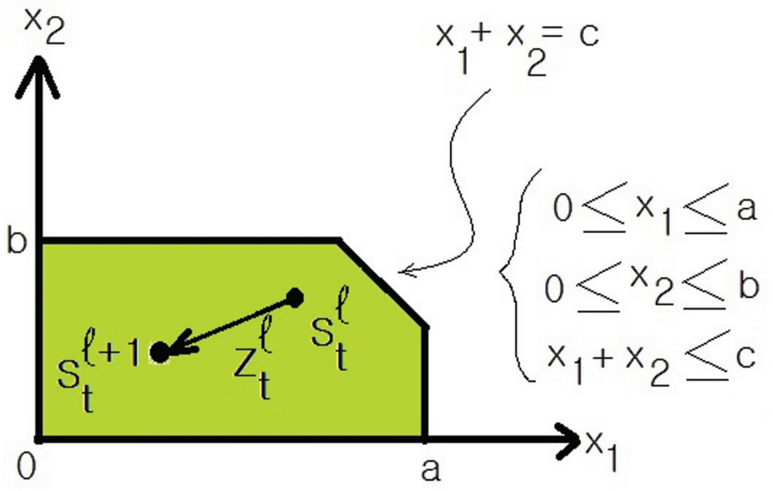


Figure 5. Two-dimensional representation of the state space and a sample action in iteration ℓ of time slot t.

4.2. The Proposed QLRA algorithm

In this subsection, we introduce the proposed Q-learning-based resource allocation (QLRA) algorithm based on the Q-learning methodology. As we described earlier, in the Q-learning algorithm, two distinct exploration and exploitation phases exist. To avoid local optima, the Q-learning method enters the exploration phase by a pre-defined strategy. Different strategies exist, which allows the Q-learning algorithm to switch between these two phases. Among these strategies, the ϵ-greedy policy is the one that is most frequently used [65].

In this policy, a parameter $\epsilon(0 \leq \epsilon \leq 1)$ is used to trim the exploration part. In other words, in the current training epoch, the agent explores a randomly chosen action with probability ϵ and, then, exploits this selected action with probability $(1 - \epsilon)$ [63]. To balance exploration and exploitation, we used a function introduced by the authors in [64]. This function enforces a high probability of exploration for the beginning of training and a high probability of exploitation in the last stages of the training periods, as described in Equation (12):

$$\epsilon = \frac{0.5}{1 + \mathbf{exp}\left(\frac{10\times(\ell - 0.4\times L)}{L}\right)} \tag{12}$$

in which L is the total number of training episodes and ℓ represents the current and remaining number of training episodes.

The Q-value/-score is normally initialised to 0 at the very beginning of the learning procedure. Then, the Q-value is updated according to the Q-learning update rule given by Equation (13). The Q-value is used for evaluating the performance of the specific actions and states.

$$Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell}) \leftarrow Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell}) + \gamma(\ell)\left(\mathbf{R}_t^{\ell} + \zeta \max_{\mathcal{Z}^{\ell}} Q(\mathbf{S}_t^{\ell+1}, \mathbf{Z}_t^{\ell}) - Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell})\right) \tag{13}$$

where $\mathbf{S}_t^{\ell}$ demonstrates the environment state in time slot t in training period/episode ℓ. $\mathbf{Z}_t^{\ell}$ denotes the action of the agent in time slot t of training episode ℓ. $\mathcal{Z}^{\ell}$ means the action space. ζ $(0 \leq \zeta \leq 1)$ is the so-called discount proportion. γ is the learning rate $(0 \leq \gamma(\ell) < 1)$, and $\mathbf{R}_t^{\ell}$ is the reward value in time slot t of training episode ℓ. $\max_{\mathcal{Z}^{\ell}} Q(\mathbf{S}_t^{\ell+1}, \mathbf{Z}_t^{\ell})$ is the maximum reward that can be obtained from state $\mathbf{S}_t^{\ell+1}$. Algorithm 1 represents the detailed QLRA algorithm.

Algorithm 1 Proposed QLRA algorithm.

```
1 Initialise $Q$-score and time slot, $Q$ = 0, $t$ = 0
2 $t = t + 1$
3 Generate state set $\mathcal{S}(t)$
4 **for** $\ell = 1 : L$ **do**:
5 Generate action set $\mathcal{Z}^{\ell}(t)$ and $\epsilon$ according to [64]:
6 $\epsilon = \frac{0.5}{1+exp^{(\frac{10\times(\ell-0.4\times L)}{L})}}$
7 Choose an action according to:
8 $$\mathbf{Z}_t^{\ell} = \begin{cases} \mathbf{Z}_t^{\ell} \in \max_{\mathcal{Z}^{\ell}} Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell}) & \text{with probability}(1-\epsilon) \\ \text{a uniformly random action in } \mathcal{Z}^{\ell} & \text{with probability}(\epsilon) \end{cases}$$
9 $\mathbf{S}_t^{\ell+1} = \mathbf{S}_t^{\ell} + \mathbf{Z}_t^{\ell}$
10 $\mathbf{R}_t^{\ell} = \Gamma \sum_{i=1}^{M(t)} \mathcal{U}^i(t) + (1-\Gamma)\mathcal{U}^{CP}(t)$
11 Update Q according to:
12 $Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell}) \leftarrow Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell}) + \gamma(\ell)\Big(\mathbf{R}_t^{\ell} + \zeta \max_{\mathcal{Z}^{\ell}} Q(\mathbf{S}_t^{\ell+1}, \mathbf{Z}_t^{\ell}) - Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell})\Big)$
13 $\mathbf{S}_t^{\ell} \leftarrow \mathbf{S}_t^{\ell+1}$
14 **If**
15 $|Q(\mathbf{S}_t^{\ell+1}, \mathbf{Z}_t^{\ell}) - Q(\mathbf{S}_t^{\ell}, \mathbf{Z}_t^{\ell})| < \delta$ ($\delta$ is a small positive constant)
16 **Go to 3**
17 Obtain the final $Q$-score.
```

To give a clearer insight to the readers about the mentioned algorithm, the flowchart of the proposed QLRA algorithm is depicted in Figure 6.

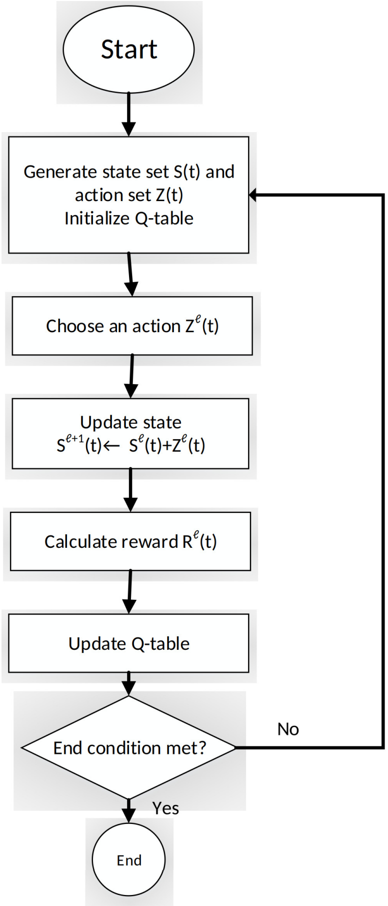

Start
Generate state set S(t) and action set Z(t) Initialize Q-table
Choose an action Z$^{\ell}$(t)
Update state S$^{\ell+1}$(t)← S$^{\ell}$(t)+Z$^{\ell}$(t)
Calculate reward R$^{\ell}$(t)
Update Q-table
End condition met?
No
Yes
End

Figure 6. Flowchart of the proposed QLRA algorithm.

4.3. Remark

It was shown in [66] that, under the following two conditions (Equation (14)), the proposed QLRA algorithm will converge to the optimal resource allocation values with a probability of one (based on the above fact, we selected the simple form $\gamma(\ell) = \frac{0.5}{\ell}$ (where ℓ is the iteration number of the QLRA method) in Section 5.3 of the simulation results Section):

$$\sum_{\ell=1}^{\infty} \gamma(\ell) = \infty \quad , \quad \sum_{\ell=1}^{\infty} \gamma^2(\ell) < \infty \tag{14}$$

5. Simulation Results

To investigate the performance of the proposed reinforcement learning algorithm, the numerical analysis is presented in this section. The numerical analysis is composed of three parts, which are the experimental analysis parameter descriptions, simulating the proposed QLRA algorithm, and the performance comparison with the related work.

5.1. Numerical Analysis Parameter Description

For the simulation, a PC with a 2.93 GHz Inter(R) i7 CPU and 8 GB of memory running Linux Ubuntu 18.04 LTS was selected. We used python3, and its matplotlib library was used.

During the simulation, we selected three different user population scenarios, which were a large number of cloud users, a medium number of cloud users, and a small number of cloud users. To describe it more clearly, assume that data centre network is capable of serving Y users on average for the current time slot. If the number of requesting users is around Y, we call it a small number of users. On the other hand, if the number of active requesting cloud users is around $2Y$ and $5Y$ on average, we call these medium and large user populations, respectively.

In the numerical analysis, we selected the number of large, medium, and small cloud users to be 50, 20, and 5, respectively. Table 3, represents other simulation variables that were adopted for the proposed QLRA algorithm, in greater detail.

Table 3. Simulation parameters.

Parameter	Value
Number of users (M)	5 (small), 20 (medium), 50 (large)
Number of data centres (N)	6
Minimum user bandwidth ($\mathcal{B}_{min}$)	100 Mbps
Minimum user CPU cores ($\mathcal{C}_{min}$)	1 Core
Minimum user RAM ($\mathcal{D}_{min}$)	1 Gigabyte
Minimum user storage ($\mathcal{G}_{min}$)	100 Megabytes
Minimum data centre racks	10
Total cloud provider racks	500
Maximum cloud provider utility	20
Maximum achievable user utility	5
Γ	0.8
γ	0.1, 0.2, 0.5
ζ	0.1, 0.2, 0.5
ω_i, i = 1, …, 4	5
(α_1, α_2, α_3, α_4, α_5, α_6, α_7)	(0.01, 0.1, 0.0001, 0.001, 0.1, 0.1, 0.1)

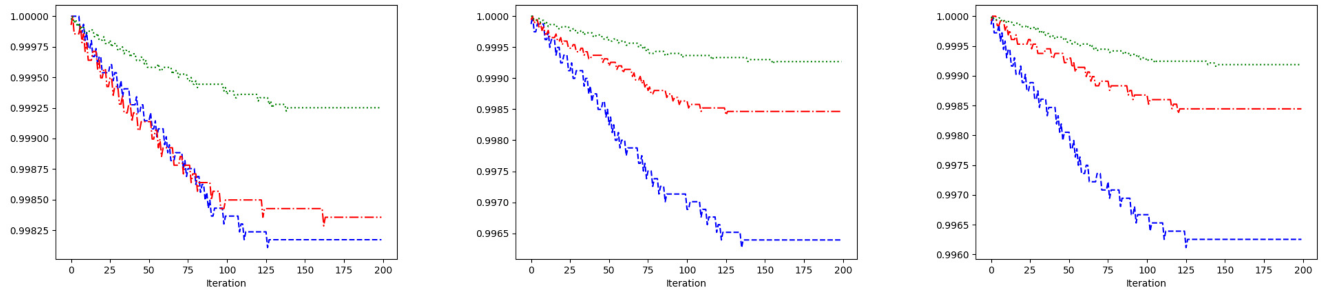
Table 3. *Cont.*

Parameter	Value
$k_i, i = 1, \ldots, 7$	5
β_1	0.05
β_2	0.5
β_3	0.0005
τ	0.5
δ	0.001
L	10,000

5.2. QLRA Algorithm Results and Comparative Analysis

To simulate the QLRA algorithm, we used some simulation parameters, which are depicted in Table 3. In three different scenarios, the convergence property of the average user utility (which is described in Equation (2)) was investigated for multiple learning parameters (in terms of γ and ζ) when different user population scenarios (small (blue), medium (red), large (green)) were taken into account. As can be verified, the QLRA is able to find the optimal resource allocation based on model-free user-data centre network interactions as described in the previous sections in each time slot of the system.

Figure 7a–c show the convergence behaviour of the average cloud user utility (refer to Equation (7)) for multiple learning parameters (in terms of γ and ζ) and, also, for different user populations (small (blue), medium (red), and large (green)). As can be seen, in Figure 7a–c, the mean user utility functions converged to optimal values after approximately 175 iterations.



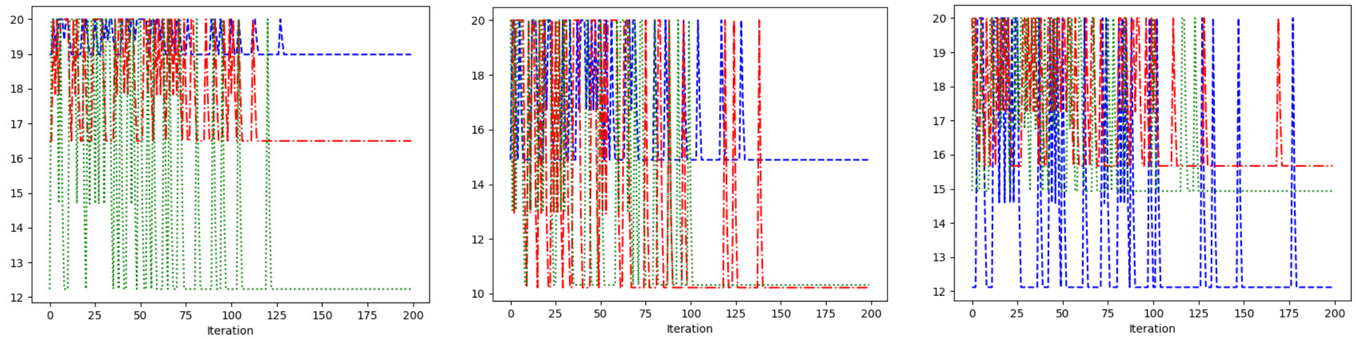
(**a**) $\gamma = 0.5, \zeta = 0.1$ (**b**) $\gamma = 0.1, \zeta = 0.5$ (**c**) $\gamma = 0.2, \zeta = 0.2$

Figure 7. Convergence behaviour of average user utility (2) for multiple user populations (small (blue), medium (red), large (green)).

Figure 8a–c show the convergence behaviour of the average cloud manager satisfaction/utility function for multiple learning parameters (in terms of γ and ζ) and, also, for different user populations (small (blue), medium (red), and large (green)). As can be verified in Figure 8a–c, the user cloud provider utility functions can be converged to optimal values after approximately 175 iterations.

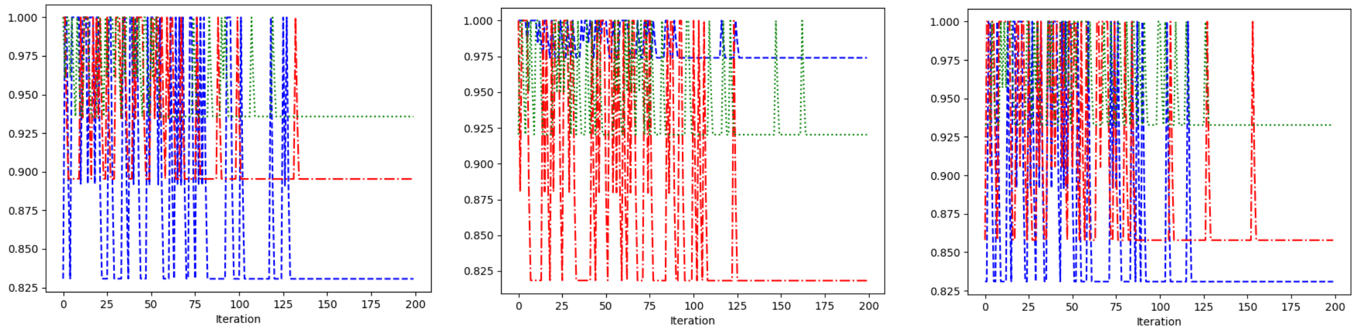
In Figure 9a–c, the convergence property of the mean reward function of the proposed Q-learning algorithm for different learning parameters (in terms of γ and ζ) and also for different user populations (small (blue), medium (red) and large (green)). As can be seen in Figure 9a–c, the mean reward function can converge to optimal values after approximately 175 iterations.

Finally, in Figure 10a–c, the convergence property of the mean queue function (Q) of the proposed Q-learning algorithm for different learning parameters (in terms of γ and ζ) and, also, for different user populations (small (blue), medium (red), and large (green)) can be seen. As can be verified in Figure 10a–c, the mean queue function can also converge to optimal values after approximately 150 iterations.



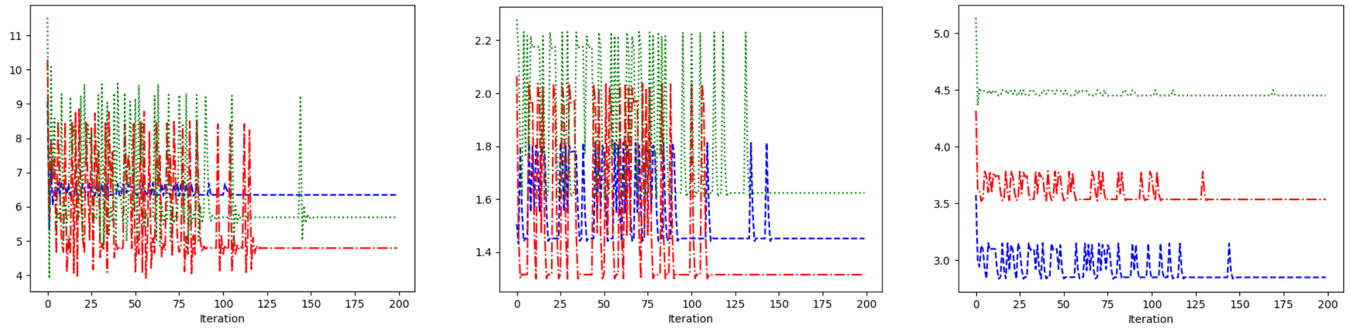
(**a**) $\gamma = 0.5$, $\zeta = 0.1$ (**b**) $\gamma = 0.1$, $\zeta = 0.5$ (**c**) $\gamma = 0.2$, $\zeta = 0.2$

Figure 8. Convergence behaviour of average CP utility (7) for multiple user populations (small (blue), medium (red), large (green)).



(**a**) $\gamma = 0.5$, $\zeta = 0.1$ (**b**) $\gamma = 0.1$, $\zeta = 0.5$ (**c**) $\gamma = 0.2$, $\zeta = 0.2$

Figure 9. Convergence behaviour of average reward function for multiple user populations (small (blue), medium (red), large (green)).



(**a**) $\gamma = 0.5$, $\zeta = 0.1$ (**b**) $\gamma = 0.1$, $\zeta = 0.5$ (**c**) $\gamma = 0.2$, $\zeta = 0.2$

Figure 10. Convergence behaviour of average queue function for multiple user populations (small (blue), medium (red), large (green)).

5.3. Performance Comparison with Similar Approaches

In this part, the performance of the proposed reinforcement-learning-based rate allocation algorithm is compared with the dynamic bin-packing resource allocation method in [28] (Xu et al.), the CCEA method in [3], and the game-theory-based resource allocation method (Li et al.) [48] from the simultaneous cloud user/cloud provider utility satisfaction perspective. A joint and exponentially weighted average—which incorporates all cloud users' and cloud provider utilities into account—utility function was designed for different slots t, as described by Equation (15):

$$\mathcal{U}(t) \triangleq \alpha(t) \sum_{i=1}^{M(t)} \mathcal{U}^{(i)}(t) + (1-\alpha(t))\mathcal{U}^{(CP)}(t) \tag{15}$$

We previously described the cloud user and cloud provider utility functions in Equations (2) and (7) of Section 3, respectively. The positive $0 < \alpha(t) < 1$ is in fact an exponentially weighted parameter to leverage the total cloud user utilities and cloud provider utility at each time frame t. We selected five different cloud data centres ($N(t) = 5$) that serve 120 demanding cloud users ($M(t) = 120$) in each time frame t. We repeated each run of the methods 200 different times and, then, took the average result to cover the probabilistic properties of multiple algorithms. The results are presented in Figure 11. It can be verified that the proposed RL-based method improved the mean convergence performance to the optimal solution from the perspective of simultaneous cloud provider/user satisfaction. The most important reason that similar work in [48] and [28] had worse mean satisfaction performance is the fact that these methods do not consider the joint SLA satisfaction for both cloud users and cloud providers and their main point of interest was optimising the resources from only the cloud provider or cloud user point of view. In comparison with our previous work [3], the proposed QLRA algorithm has better convergence behaviour and is able to converge to an optimal resource assignment strategy in the early stages of the simulation.

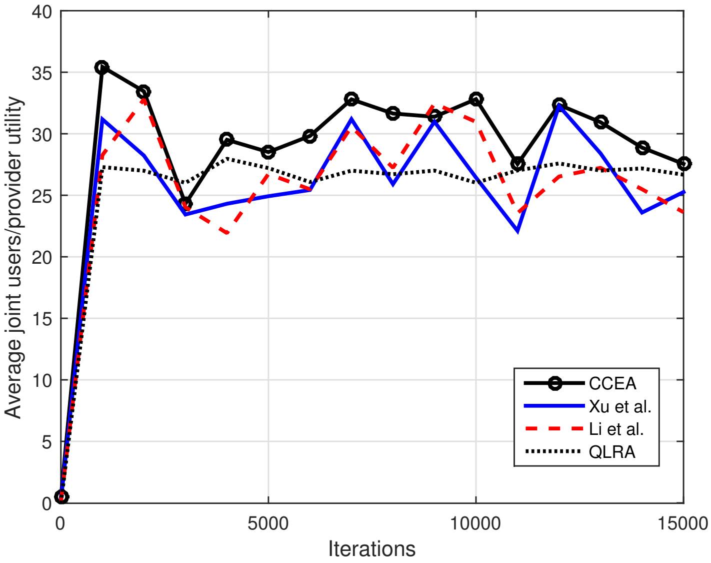


Figure 11. Average joint users'/provider utility comparison.

In Table 4, we compare the computational complexities of the different algorithms in terms of the mean execution time. As can be verified, the online feature of the proposed algorithm results in a reduced computational time in comparison with similar approaches. On the other hand, we compared the mean holistic utility according to Equation (15) between different methods during 1 time slot and 10 time slots, respectively. It can be verified that, although in the short term, the proposed QLRA method has a bit lower mean value in comparison with the CCEA method, in the long term, it has a comparable performance with it and a superior performance in comparison with the other methods.

Table 4. Computational complexity and mean holistic utility over time slot comparison.

Algorithm	Mean Execution Time (Seconds)	Mean Time Slot Holistic Utility	Mean 10 Time Slots' Holistic Utility
CCEA	2.93	28.03	35.993
QLRA	1.54	27.7	35.931
Xu et al.	3.44	26.12	29.241
Li et al.	7.45	27.23	33.031

6. Conclusions

In this paper, after mathematical modelling of the cloud users' and cloud provider utility functions, a Q-learning-based optimisation algorithm (called QLRA) was developed for joint cloud customer/provider utility satisfaction. The cloud system parameters are mapped into the agent, state, action, and reward elements of a Q-learning algorithm. The experimental findings demonstrate the effectiveness of the developed QLRA from the perspective of the average holistic utility satisfaction of cloud data centre users and cloud data centre providers in comparison with similar approaches. Another important feature of the proposed method is the online and model-free property of the RL algorithm, which results in converging to the optimal utility levels for both cloud users and cloud providers in different cloud user population scenarios under energy efficiency constraints. We also compared the mean computational complexity and mean holistic utility over multiple time slots between different methods. It was shown that the proposed QLRA method, while having comparable long-term mean holistic utility performance, has a better mean computational complexity in comparison with similar approaches.

One of the future open research issues is the modification of the developed QLRA for massive cloud users using deep Q-learning or deep RL techniques. We also suggest using emerging game-based techniques such as mean field game (MFG) theory and, also, using novel evolutionary game strategies for simultaneous utility satisfaction for the overall cloud data centre network system under energy efficiency constraints.

Author Contributions: Conceptualisation, P.G. and J.L.; methodology, P.G.; software, M.H. and R.G.; validation, P.G., M.H. and R.G.; formal analysis, P.G.; investigation, P.G.; resources, P.G.; data curation, P.G.; writing—original draft preparation, P.G.; writing—review and editing, P.G. and J.L.; visualisation, M.H. and R.G.; supervision, P.G.; project administration, P.G.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: We would like to express our gratitude to ITRC and UPV for their joint support of this research.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
CDC	Cloud data centre
CDCN	Cloud data centre network
CP	Cloud provider
DC	Data centre
DCN	Data centre network
ML	Machine learning
QoE	Quality of experience
QoS	Quality of service

RL	Reinforcement learning
RTT	Round-trip time
SLA	Service-level agreement
VM	Virtual machine

References

1. Wang, B.; Qi, Z.; Ma, R.; Guan, H.; Vasilakos, A.V. A survey on data centre networking for cloud computing. *Comput. Netw.* **2015**, *91*, 528–547. [CrossRef]
2. Stergiou, C.L.; Psannis, K.E.; Gupta, B.B. InFeMo: Flexible Big Data Management Through a Federated Cloud System. *ACM Trans. Internet Technol.* **2022**, *22*, 1–22. [CrossRef]
3. Goudarzi, P.; Hosseinpour, M.; Ahmadi, M.R. Joint customer/provider evolutionary multi-objective utility maximization in cloud data centre networks. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2021**, *45*, 479–492. [CrossRef]
4. Samuel, A.L. Some studies in machine learning using the game of checkers. *IBM J. Res. Develop.* **1959**, *3*, 210–229. [CrossRef]
5. Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach. In *Prentice-Hall*, 3rd ed.; Pearson Education, Inc.: Upper Saddle River, NJ, USA, 2009.
6. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
7. Barlow, B.H. Unsupervised learning. *Neural Comput.* **1989**, *1*, 295–311. [CrossRef]
8. Ghahramani, Z. Unsupervised learning. In *Summer School on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 72–112.
9. Zhu, X. Semi-supervised learning literature survey. In *Tech. Rep.* ; Dept. Comput. Sci., Univ. Wisconsin-Madison: Madison, WI, USA, 1530. Available online: https://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf (accessed on 15 November 2022).
10. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
11. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
12. Tryon, R.C. *Cluster Analysis: Correlation Profile and Orthometric (Factor) Analysis for the Isolation of Unities in Mind and Personality*; Edwards brother, Inc.: Lillington, NC, USA, 1939.
13. Estivill-Castro, V. Why so many clustering algorithms: A position paper. *ACM SIGKDD Explor. Newslett.* **2002**, *4*, 65–75. [CrossRef]
14. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2327. [CrossRef]
15. Ahmad, I.; Shahabuddin, S.; Malik, H.; Harjula, E.; Leppänen, T.; Loven, L.; Anttonen, A.; Sodhro, A.H.; Alam, M.M.; Juntti, M.; et al. Machine Learning Meets Communication Networks: Current Trends and Future Challenges. *IEEE Access* **2020**, *8*, 223418–223460. [CrossRef]
16. Buda, T.S.; Assem, H.; Xu, L.; Raz, D.; Margolin, U.; Rosensweig, E.; Lopez, D.R.; Corici, M.-I.; Smirnov, M.; Mullins, R.; et al. Can machine learning aid in delivering new use cases and scenarios in 5G? In Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 1279–1284.
17. Ahmad, I.; Shahabuddin, S.; Kumar, T.; Harjula, E.; Meisel, M.; Juntti, M.; Sauter, T.; Ylianttila, M. Challenges of AI in wireless networks for IoT. *IEEE Ind. Electron. Mag.* **2020**, *15*, 1–16.
18. Jain, N.; Choudhary, S. Overview of virtualization in cloud computing. In Proceedings of the 2016Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 18–19 March 2016.
19. Razaque, A.; Vennapusa, N.R.; Soni, N.; Janapati, G.S.; Vangala, K.R. Task scheduling in Cloud computing. In Proceedings of the IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 9 April 2016.
20. Zhu, F.; Li, H.; Lu, J. A service level agreement framework of cloud computing based on the Cloud Bank model. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 25–27 May 2012.
21. Wu, L.; Garg, S.K.; Buyya, R. SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments. *J. Comput. Syst. Sci.* **2012**, *78*, 1280–1299. [CrossRef]
22. Rebai, S. Resource allocation in Cloud federation. These de Doctorat Conjoint Telecom SudParis et L'Universite Pierre et Marie Curie, 2017. Available online: https://theses.hal.science/tel-01534528/document (accessed on 15 November 2022).
23. Goudarzi, P. Multi-Source Video Transmission with Minimized Total Distortion Over Wireless Ad Hoc Networks. *Wirel. Pers. Commun.* **2009**, *50*, 329–349. [CrossRef]
24. Zheng, K.; Meng, H.; Chatzimisios, P.; Lei, L.; Shen, X. An SMDP-based resource allocation in vehicular cloud computing systems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7920–7928. [CrossRef]
25. Zhou, Z.; Bambos, N. A general model for resource allocation in utility computing. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015.
26. Khasnabish, J.N.; Mitani, M.F.; Rao, S. Generalized nash equilibria for the service provisioning problem in cloud systems. *IEEE Trans. Serv. Comput.* **2013**, *6*, 429–442.
27. Johari, R.; Mannor, S.; Tsitsiklis, J.N. Efficiency loss in a network resource allocation game: The case of elastic supply. *IEEE Trans. Autom. Control* **2005**, *50*, 1712–1724. [CrossRef]
28. Li, Y.; Tang, X.; Cai, W. Dynamic bin packing for on-demand cloud resource allocation. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 157–170. [CrossRef]

29. Liang, H.; Cai, L.; Huang, D.; Shen, X.; Peng, D. An SMDP-based service model for interdomain resource allocation in mobile cloud networks. *IEEE Trans. Veh. Technol.* **2012**, *61*, 157–170.
30. Zu, D.; Liu, X.; Niu, Z. Joint Resource Provisioning for Internet Datacentres with Diverse and Dynamic Traffic. *IEEE Trans. Cloud Comput.* **2017**, *5*, 71–84.
31. Goudarzi, P.; Sheikholeslam, F. A fast fuzzy-based (Ω, *α*)-fair rate allocation algorithm. In Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, Denver, CO, USA, 4–8 April 2005.
32. Pilla, P.S.; Rao, S. Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory. *IEEE Syst. J.* **2016**, *10*, 637–648. [CrossRef]
33. Li, B.; Li, J.; Tang, K.; Yao, X. Many-Objective Evolutionary Algorithms: A Survey. *ACM Comput. Surv.* **2015**, *48*, 1–35. [CrossRef]
34. Deb, K. Multi-objective evolutionary algorithms. In *Springer Handbook of Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 995–1015.
35. Chang, R.; Li, M.; Li, K.; Yao, X. Evolutionary Multiobjective Optimization Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection. *IEEE Trans. Evol. Comput.* **2017**, *22*, 692–706. [CrossRef]
36. Li, X.; Garraghan, P.; Jiang, X.; Wu, Z.; Xu, J. Holistic Virtual Machine Scheduling in Cloud Datacentres towards Minimizing Total Energy. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *29*, 1317–1331. [CrossRef]
37. Sayadnavard, M.H.; Haghighat, A.T.; Rahmani, A.M. A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centres. *Eng. Sci. Technol. Int. J.* **2021**, *6*, 100995.
38. Zhang, C.; Wang, Y.; Lv, Y.; Wu, H.; Guo, H. An Energy and SLA-Aware Resource Management Strategy in Cloud Data Centers. *Sci. Program.* **2019**, *2019*, 3204346. [CrossRef]
39. Ilager, S. Machine Learning-based Energy and Thermal Efficient Resource Management Algorithms for Cloud Data Centres. Ph.D. Dissertation, University of Melbourne, Melbourne, Australia, 2021.
40. Gill, S.S.; Garraghan, P.; Stankovski, V.; Casale, G.; Thulasiram, R.K.; Ghosh, S.K.; Ramamohanarao, K.; Buyya, R. Holistic Resource Management for Sustainable and Reliable Cloud Computing: An Innovative Solution to Global Challenge. *J. Syst. Software* **2019**, *155*, 104–129. [CrossRef]
41. Heimerson, A.; Brännvall, R.; Sjölund, J.; Eker, J.; Gustafsson, J. Towards a Holistic Controller: Reinforcement Learning for Data Center Control. *e-Energy* **2021**, 424–429. Available online: https://dl.acm.org/doi/10.1145/3447555.3466581 (accessed on 15 November 2022).
42. Baek, J.Y.; Kaddoum, G.; Garg, S.; Kaur, K.; Gravel, V. Managing Fog Networks Using Reinforcement Learning Based Load Balancing Algorithm. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019.
43. Garcia, J.L.B. Improved Self-management of DataCenter Systems Applying Machine Learning. Ph.D. Dissertation, Universitat Politecnica de Catalunya, Barcelona, Spain, 2013.
44. Li, D.; Chen, C.; Guan, J.; Zhang, Y.; Zhu, J.; Yu, R. DCloud: Deadline-Aware Resource Allocation for Cloud Computing Jobs. *IEEE Trans. Parallel Distrib. Syst.* **2016**, 27, 2248–2260. [CrossRef]
45. Parikh, S.M. A survey on cloud computing resource allocation techniques. In Proceedings of the 2013 Engineering (NUiCONE), Ahmedabad, India, 28–30 November 2013.
46. Khasnabish, J.N.; Mitani, M.F.; Rao, S. Tier-Centric Resource Allocation in Multi-Tier Cloud Systems. *IEEE Trans. Cloud Comput.* **2017**, *5*, 576–589. [CrossRef]
47. Liu, X.F.; Zhan, Z.H.; Deng, J.D.; Li, Y.; Gu, T.; Zhang, J. An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing. *IEEE Trans. Evol. Comput.* **2017**, *22*, 113–128. [CrossRef]
48. Xu, X.; Yu, H. A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing. *Math. Probl. Eng.* **2014**, *2014*, 915878. [CrossRef]
49. Ashraf, A.; Porres, I. Multi-objective dynamic virtualmachine consolidation in the cloud using ant colony system. *Int. J. Parallel, Emergent Distrib. Syst.* **2017**, *33*, 103–120. [CrossRef]
50. Md, Feraus, H.; Murshed, M.; Calheiros, R.N.; Buyya, R. Multi-objective, Decentralized Dynamic Virtual Machine Consolidation using ACO Metaheuristic in Computing Clouds. In Concurrency and Computation: Practice and Experience. 2016. Available online: https://ui.adsabs.harvard.edu/abs/2017arXiv170606646H/abstract (accessed on 15 November 2022).
51. Shaw, R.; Howley, E.; Barrett, E. Applying Reinforcement Learning towards automating energy efficient virtual machine consolidation in cloud data centres. *Inf. Syst.* **2022**, *107*, 101722. [CrossRef]
52. Lin, B.; Zhu, F.; Zhang, J.; Chen, J.; Chen, X.; Xiong, N.N.; Lloret, J. A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4254–4265. [CrossRef]
53. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. *Expert Syst. Appl.* **2019**, *124*, 196–208. [CrossRef]
54. Mao, H.; Schwarzkopf, M.; Venkatakrishnan, S.B.; Meng, Z.; Alizadeh, M. Learning scheduling algorithms for data processing clusters. In Proceedings of the ACM Special Interest Group on Data Communication, Beijing, China, 19–23 August 2019; pp. 270–288.
55. Ghafouri, S.; Saleh-Bigdeli, A.A.; Doyle, J. Consolidation of Services in Mobile Edge Clouds using a Learning-based Framework. In Proceedings of the IEEE World Congress on Services (SERVICES), Beijing, China, 18–23 October 2020; pp. 116–121.

56. Guo, W.; Tian, W.; Ye, Y.; Xu, L.; Wu, K. Cloud resource scheduling with deep reinforcement learning and imitation learning. *IEEE Internet Things J.* **2020** , *8*, 3576–3586. [CrossRef]
57. Buyya, R.; Beloglazov, A.; Abawajy, J. Energy-efficient management of data centre resources for cloud computing: A vision architectural elements and open challenges. In Proceedings of the Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, NV, USA, 12–15 July 2010.
58. Das, R.; Kephart, J.O.; Lenchner, J.; Hamann, H. Utility-Function-Driven Energy-Efficient Cooling in Data Centers. In Proceedings of the 7th International Conference on Autonomic Computing (ICAC), Washington, DC, USA, 7–11 June 2010.
59. Ranaldo, N.; Zimeo, E. Capacity-Aware Utility Function for SLA Negotiation of Cloud Services. In Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC), Dresden, Germany, 9–12 December 2013.
60. ITU-T. Vocabulary for Performance and Quality of Service. 2006. Available online: https://www.itu.int/rec/T-REC-P.10 (accessed on 15 November 2022).
61. Wei, D.X.; Cao, P.; Low, S.H. Packet Loss Burstiness: Measurements and Implications for Distributed Applications. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium, IPDPS, Long Beach, CA, USA, 26–30 March 2007.
62. Elteto, T.; Molnar, S. On the distribution of round-trip delays in TCP/IP networks. In Proceedings of the 24th Conference on Local Computer Networks. LCN, Lowell, MA, USA, 18–20 October 1999.
63. Yu, L.; Zhang, C.; Jiang, J.; Yang, H.; Shang, H. Reinforcement learning approach for resource allocation in humanitarian logistics. *Expert Syst. Appl.* **2021**, *173*, 114663. [CrossRef]
64. Semrov, D.; Marsetic, R.; Zura, M.; Todorovski, L.; Srdic, A. Reinforcement learning approach for train rescheduling on a single-track railway. *Transp. Res. Part B Methodol.* **2016**, *86*, 250–267. [CrossRef]
65. Russel, S.J.; Norvig, P. *Artificial Intelligence, a Random Approach*, 3rd ed.; Pearson Education: New York, NY, USA. 2003.
66. Regehr, M.T.; Ayoub, A. An Elementary Proof that Q-learning Converges Almost Surely. *arXiv* **2021**, arXiv:2108.02827.