Multi-Agent Reinforcement Learning with Shared Policy for Cloud Quota Management Problem

Tong Cheng*, Hang Dong*, Lu Wang*, Bo Qiao*, Si Qin*, Qingwei Lin*
Dongmei Zhang*, Saravan Rajmohan§, Thomas Moscibroda†

Microsoft Research* Microsoft 365§ Microsoft Azure†

ABSTRACT

Quota is often used in resource allocation and management scenarios to prevent abuse of resource and increase the efficiency of resource utilization. Quota management is usually fulfilled with a set of rules maintained by the system administrator. However, maintaining these rules usually needs deep domain knowledge. Moreover, arbitrary rules usually cannot guarantee both high resource utilization and fairness at the same time. In this paper, we propose a reinforcement learning framework to automatically respond to quota requests in cloud computing platforms with distinctive usage characteristics for users. Extensive experimental results have demonstrated the superior performance of our framework on achieving a great trade-off between efficiency and fairness.

CCS CONCEPTS

• **Applied computing** → **Enterprise computing infrastructures**; • **Computing methodologies** → *Multi-agent reinforcement learning*; **Multi-agent reinforcement learning**; **Multi-agent planning**.

KEYWORDS

reinforcement learning , cloud computing, multi-agent system

ACM Reference Format:

Tong Cheng*, Hang Dong*, Lu Wang*, Bo Qiao*, Si Qin*, Qingwei Lin*, Dongmei Zhang*, Saravan Rajmohan§, Thomas Moscibroda†. 2023. Multi-Agent Reinforcement Learning with Shared Policy for Cloud Quota Management Problem. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543873.3584634

1 INTRODUCTION

Cloud computing is playing an important role in modern computing industry by providing elastic resources in a pay-as-you-go paradigm [21, 25]. From the perspective of the cloud computing platform, it is ideal to achieve high utilization of computing resources to improve the overall profit. However, there are many other factors to consider, including but not limited to coordinating the resource usage of users to avoid shortage of computing resources, enlarging capacity for growing users, ensuring the fairness of resource allocation, etc [23].

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3584634

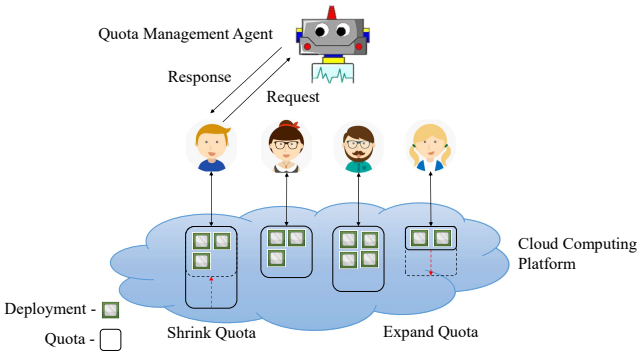

Figure 1: Cloud Quota Management Problem

To prevent the abrupt growth of customers that could violate the available capacity, cloud computing platforms such as Amazon Web Services and Microsoft Azure usually set a default limit of computing resources for each user called *quota* and flexibly adjust this limit according to users' historical usage and current demand. A quota management system is therefore adopted to manage the quota for all these users to properly plan and stabilize the actual resource usage of the platform. As the total computing resource is limited for the cloud computing platform, the allocated quota should also be limited with careful management. A reasonable quota management system is supposed to 1) satisfy the expansion requests for users with growing demands, 2) avoid large quota amount allocated to users with low demands on computing resources, and 3) maintain a reasonably fair balance among all the users on the quota management. In this paper, we investigate the quota management problem in cloud computing platforms, as illustrated in Figure 1. When users have growing demands on computing resources, they will submit requests with specific amounts of required quota. Then the system can respond to users by approving the request, rejecting the request, or partially admitting the request. Besides, when users have a decreasing demand on the computing resources, they can also request to decrease their quota, and these requests will take effect automatically by default.

2 CHALLENGES AND MOTIVATION

Generally, cloud quota management problem can be regarded as a special type of resource allocation problem. Many previous works have investigated the resource allocation problem. For instance, efficiency and fairness are optimized to solve resource allocation problem in cloud computing platform [30, 31]. However, these works formulate the resource allocation problem as a static optimization problem without considering the dynamics of the system.

As the system evolves with complex user dynamics, the allocation plan should also be dynamically adjusted. Therefore, many endeavors have been made to design deep reinforcement learning methods for solving resource allocation problems due to its natural dynamic optimization setting [1, 5, 15]. However, most existing reinforcement learning-based works solve the resource allocation problem by Q-learning method, which needs to enumerate all possible actions given current state. Due to the high-dimensional and continuous action space of cloud quota management problem, it is impossible to enumerate all actions, making many of these methods not applicable for this problem. Moreover, one of the remarkable differences between cloud quota management problem and other resource allocation problem is that system grant quota as resource qualification to users instead of allocating resources directly to users, giving more usage freedom to users as users can adjust deployment according to varying requirements as long as the number does not surpass quota. This problem setting makes the allocation more complex and difficult to achieve efficiency and fairness at the same time.

For a cloud computing platform, it is challenging to manage quota among end users with reinforcement learning for the following three reasons. Firstly, the action space in such a quota management can be complex, thus difficult to handle [6]. In quota management problem, quota management agent needs to take two different kinds of actions at the same time: user-level satisfaction ratio and system-level relax factor, making the action space heterogeneous. Secondly, traditional reinforcement learning techniques exhibit low efficiency on quota management problem with high dimensional continuous spaces, making effective exploration difficult [15]. Finally, it is difficult to balance the system efficiency and fairness among users at the same time for a cloud computing platform. To resolve the three challenges aforementioned, we propose a novel reinforcement learning framework called **M**ulti-**A**gent reinforcement learning with **S**hared **P**olicy (MASP) and adopt a reward communication mechanism. Extensive experiments have been conducted on a cloud simulation environment developed based on data from a number of first-party cloud computing users in Microsoft.

3 METHODOLOGY

3.1 Preliminaries and Definitions

In the quota management system, there are mainly two roles: users and the platform. For a user in quota management problem, we consider the following three aspects : 1) *Quota* is the upper limit of resource one user could deploy and is updated by request-and-response interaction between user and platform; 2) *Deployment* is the actual deployed computing resources determined by smaller value between user's quota and expected deployment which is determined by user's demand; 3) *Credit score* is a concept originally developed in the banking industry, which is used to provide an assessment for the circumstances of lender and borrower, and reflects the lender's view of the likely future economic scenarios [28]. In cloud computing quota management problem, the credit score encodes the historical information of both deployment and quota for a user to evaluate the user's tendency to adequately utilize quota. The platform generally makes two types of decisions (actions) at each time step: *satisfaction ratio* for each user as the approved proportion of the requested amount and *relax factor* as the ratio of the total amount of quota over the total capacity for the system, which is usually larger than one considering that the actual total deployment will not reach the quota at the same time for all users.

Specifically, we formulate cloud quota management problem as a discrete-time, infinite Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is a mixed state space with both discrete and continuous components, $\mathcal{A}$ is a continuous action space containing two kinds of sub-actions, R is the reward function, $\mathcal{P}$ is the transition probability and $\gamma \in [0, 1]$ is the discount factor. The state space, action space and reward function are detailed as follows:

State space $\mathcal{S}$. State s_t at time step t is denoted as $(\mathbf{r}_t; \mathbf{c}_t; \mathbf{d}_t; \mathbf{x}_t; h_t, q_t)$ consisting of the request vector $\mathbf{r}_t = (r_t^1 \cdots r_t^N)$, credit score vector $\mathbf{c}_t = (c_t^1 \cdots c_t^N)$, deployment vector $\mathbf{d}_t = (d_t^1 \cdots d_t^N)$, quota vector $\mathbf{x}_t = (x_t^1 \cdots x_t^N)$, platform available quota h_t and total capacity q_t where N is the number of users.

Action space $\mathcal{A}$. Action a_t at time step t is denoted as $a_t = (\mathbf{b}_t; C_t)$ consisting of satisfaction ratio vector $\mathbf{b}_t = (b_t^1 \dots b_t^N)$ and relax factor C_t.

Reward R: At each round, the reward of quota management agent is calculated based on the system's allocation and users' feedback. The reward consists of four components: system deployment ratio R_D, structural allocation fairness R_F, deployment confined by quota R_H, allocation over available quota R_O, which can be written as weighted sum of these four components: $R = w_D R_D + w_F R_F + w_H R_H + w_O R_O$ where weights w_D, w_F, w_H, w_O are calibrated to balance the four components.

Specifically, $R_D = \sum_{i=1}^N d_t^i / q_t$ indicates resource utilization ratio where d_t^i is deployment for i-th user, q_t is total capacity. In a cloud computing platform, resource allocation has to be fair to attract a wide range of subscribers in order to optimize profit [11]. Usually a fairness metric should be defined as a divergence metric indicating the deviation of current allocation from a preferred one [11, 24]. In cloud quota management problem, we follow the principle that a fair allocation should have matching ranks between demand and supply for all users. Hence, structural allocation fairness $R_F = -L_p\big(g(\mathbf{c}_t) - g(\mathbf{b}_t)\big) - L_p\big(g(\mathbf{d}_t) - g(\mathbf{b}_t)\big)$ is defined as divergence between satisfaction ratio and deployment where g: $\mathbb{R}^N \to \mathbb{R}^N, \mathbf{v} \mapsto \mathbf{v}'$ is rank operator extracting the rank of each element in any vector $\mathbf{v}$. In addition to deployment and fairness, there are two malfunction cases we are concerned about. Firstly, if a user is planning to increase deployment over current quota, the actual deployment can only be the current quota of the user, in which case the user's demand is confined by current quota. This case is considered as a penalty term in the reward because such unsatisfied demand for deployment might degrade users' experience and bring potential loss for the platform. Secondly, the total allocated amount of quota could exceed the total amount of remaining quota for the whole platform after generating satisfaction ratio by a policy. In this case, the actual allocated quota will be adjusted by down-scaling the allocated quota for all users proportionally to the original output allocation at the same time. Based on these two cases, deployment confined by quota $R_H = -\sum_{i=1}^N \max(0, (d_t^i)' - x_t^i)$ and allocation over available quota $R_O = \max(\text{NAQ} - \text{PA}, 0)$ are defined as reward penalties to prevent agent from malfunction

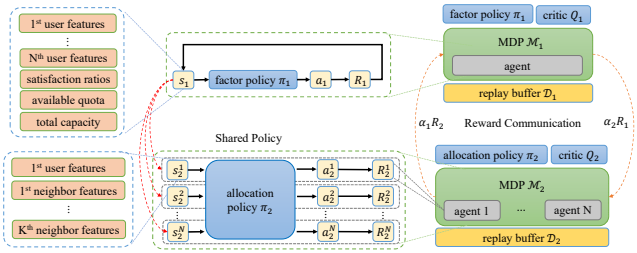

Figure 2: MASP framework with reward communication.

where $(d_t^i)'$ means the expected deployment for i-th user at time step t, PA $= \sum_{i=1}^{N} \max(0, r_t^i \cdot b_t^i)$ means positive allocation and NAQ $= C_t \cdot q_t - \sum_{i=1}^{N} \min(0, r_t^i \cdot b_t^i)$ means next available quota.

With the preliminary notations shown above, the goal of solving cloud quota management problem is to learn a quota management policy $\pi : \mathcal{S} \to \mathcal{A}, s \mapsto a$ given MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma)$ by maximizing the expected discounted cumulative reward $\mathbb{E}\left(\sum_{i=1}^{\infty} \gamma^i R\right)$. According to the application scenario of platform-level quota management, the performance of quota management policy is evaluated on deployment ratio (DR) and allocation fairness (AF).

3.2 Multi-Agent with Shared Policy and Reward Communication

For cloud quota management problem, we break the original MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma)$ into MDP $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{A}_1, R_1, \mathcal{P}_1, \gamma)$ with global information to make decision on the relax factor and MDP $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{A}_2, R_2, \mathcal{P}_2, \gamma)$ with local information to make decision on the user-level satisfaction ratio. This paradigm resolves the heterogeneity of original action space $\mathcal{A}$ by separating different actions. Moreover, for all the users we use a shared policy to ensure the consistency of the quota allocation mechanism among different users. This design could mitigate the dimension explosion problem incurred by original action space $\mathcal{A}$.

In previous works, multi-agent system with cooperation mechanism has been investigated in [16, 18, 27]. However, most of them focus on the design of state communication instead of reward communication. Under the multi-agent with shared policy paradigm, the original reward R is split into two rewards R_1 and R_2 where R_1 contains allocation fairness and R_2 contains deployment ratio. Based on this architecture, we propose a reward-communication mechanism with two hyperparameters α_1 and α_2: $R_1' = R_1 + \alpha_1 R_2$ and $R_2' = R_2 + \alpha_2 R_1$ where R_1' and R_2' are modified rewards in order to achieve the arbitrary control and customization of training process in reinforcement learning. This communication mechanism aims to achieve balance between system efficiency and fairness. Instead of picking primary and auxiliary scalers empirically, we propose an adaptive weighting method to automatically adjust primary and auxiliary scalers during the training process. The adaptive weighting technique consists of two steps: 1) update deployment ratio momentum m_1 and allocation fairness momentum m_2 by $m_t^{(i)} = 1 - \mathrm{DR}_t / \mathrm{DR}_{t-1}$ where $i = 1, 2$ where t denotes the training epoch; 2) update hyperparameters $\alpha_t^{(i)} = \alpha_{t-1}^{(i)} + \beta m_{t-1}^{(i)}$ where $i = 1, 2$ and β is a constant.

Summing up all techniques aforementioned, we propose the MASP framework with reward communication, as illustrated in Figure 2. The training of MASP is elaborated in Algorithm 1. The factor policy π_1 and allocation policy π_2 are trained using Actor Critic methods. K-Nearest Neighbor (KNN) method is applied to construct user-level state $s_2^i \in \mathcal{S}_2$ by extracting information of i-th user and its neighborhood from state $s_1 \in \mathcal{S}_1$.

Algorithm 1 MASP training

Input: initial factor policy and allocation policy, Q-functions and empty replay buffers $\mathcal{D}_1, \mathcal{D}_2$

```
Reset cloud environment at state s_1.
Extract sub-states {s_2^i}_{i=1}^N from state s_1.
repeat
    Compute relax factor a_1 and satisfaction ratios a_2^i.
    Compute rewards R_1 and R_2 (average over users).
    Obtain R_1' and R_2' by reward communication.
    Augment state s_1 with satisfaction ratios.
    Transit into next state and extract sub-states.
    Append replay buffers D_1 and D_2.
    if it is time to update then
        Update policies and Q functions.
    end if
until convergence
```

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setup

The experiments are conducted on three Linux servers with 6 cores, 110G memory and a Tesla P100 GPU. We model user's behavior by probability distributions in the cloud simulation environment and the distribution parameters of users are set up by fitting data traces, which are collected from a group of first-party users of cloud computing in Microsoft. Each model used in this section is trained up to two million interactions with the environment and evaluation of each policy is conducted using ten random seeds to ensure statistical significance. For the comparison between different policies, we focus on two evaluation metrics: deployment ratio DR $= \frac{1}{T}\sum_{t=1}^{T} R_D$ and allocation fairness AF $= \frac{1}{T}\sum_{t=1}^{T} R_F$.

4.2 Baselines

The baselines used in the experiments include: random, fixed, greedy, DDPG [14] and SAC [8]. Actions made by random policy is uniformly distributed over the whole action space independent of the state. Likewise, actions made by fixed policy is constant independent of the state. The greedy policy generates actions matching perfectly with the incoming deployment of users while other policy doesn't have the exact information of incoming deployment for decision making. Hence, greedy policy is expected to have preeminent performance on deployment ratio and could be regarded as the approximate upper bound for other methods on the performance of DR. The single-agent methods DDPG and SAC optimize the original MDP $\mathcal{M}$ directly without multi-agent paradigm and reward communication.

Policy	DR	AF
random	0.43±0.07	-866.58±10.21
fixed	0.50±0.06	-892.86±5.65
greedy	**0.87±0.05**	-986.13±22.58
DDPG	0.50±0.05	-882.54±9.12
SAC	0.25±0.02	-816.08±17.99
MASP-DDPG	0.83±0.03	**-374.36±2.84**
MASP-SAC	0.79±0.03	-385.36±3.57

Table 1: Performance evaluation by different policies on deployment ratio (DR) and allocation fairness (AF).

α_1	α_2	DR(DDPG)	AF(DDPG)	DR(SAC)	AF (SAC)
1	1	0.83±0.03	-374.36±2.84	0.79±0.03	-385.36±3.57
5	1	0.10±0.01	**-352.22±4.08**	0.51±0.04	-385.48±2.62
10	1	0.16±0.02	-373.93±6.04	0.43±0.07	-375.33±4.15
15	1	0.25±0.02	-377.36±5.95	0.41±0.03	-371.12±6.06
20	1	0.44±0.04	-365.59±5.23	0.80±0.04	-391.42±3.51
1	5	0.30±0.01	-363.95±4.14	0.79±0.02	-382.26±3.78
1	10	0.33±0.02	-366.39±5.03	0.31±0.04	**-364.73±2.95**
1	15	0.27±0.02	-361.48±6.34	0.40±0.04	-373.97±3.62
1	20	0.42±0.03	-371.09±4.92	0.45±0.04	-378.99±1.67
adaptive		**0.84±0.02**	-363.11±5.77	**0.82±0.02**	-374.17±3.16

Table 2: Performance evaluation by MASP-DDPG and MASP-SAC with different reward communication hyperparameter settings.

4.3 Experimental Results

In this section, we verify the effectiveness of MASP framework for cloud quota management problem by addressing two research questions:

- RQ1: how much does MASP improve performance on both efficiency and fairness?

In order to investigate the performance of multi-agent paradigm and shared policy technique, we implement both multi-agent paradigms combined with DDPG (MASP-DDPG) and SAC (MASP-SAC) respectively. The results of comprehensive performance evaluation are listed in Table 1 where we highlight the best performance on DR and AF respectively. The performances on DR of MASP-DDPG and MASP-SAC are significantly higher than naive policies (random, fixed) and single-agent policies (DDPG, SAC). Besides, the performances on DR of MASP-DDPG and MASP-SAC are almost equal to the greedy policy, indicating approximate optimality. Furthermore, MASP methods outperform other baselines on AF dominantly.

- RQ2: how much does adaptive reward communication improve performance on both efficiency and fairness?

We observe that setting the hyperparameters α_1 and α_2 might not be easy without any domain knowledge. According to evaluation results in Table 2, the performance on deployment ratio and allocation fairness are sensitive to the choice of hyperparameters α_1 and α_2. Inappropriate choices may undermine the performance of MASP framework. By applying the adaptive weighting method, MASP-DDPG and MASP-SAC improve further deployment ratio and allocation fairness over default setting $(\alpha_1, \alpha_2) = (1, 1)$ while arbitrary settings fail to do so. Although AF (DDPG) of the setting $(5, 1)$ and AF (SAC) of setting $(1, 10)$ in Table 2 are higher than adaptive weighting method, they perform poorly on DR.

5 APPLICATION IN PRACTICE

In the internal first-party cloud computing platform of Microsoft, a number of internal service teams are deploying their workload. Their deployment is restricted according to the approved quota of each team. The proposed quota management framework based on reinforcement learning has been applied to such a cloud computing platform to replace the human labor of checking each quota request and make decisions based on domain knowledge. After the deployment, there's a significant drop in the amount of "applied but not deployed" quota, as service teams are encouraged to apply for quota only when necessary in such a system.

6 RELATED WORK

Generally, previous works relating to quota management problem can be categorized into three groups. The first group consists of methods for general resource allocation problem [3, 7, 12, 13, 19, 32]. These works mainly focus on traditional methods like solving min-max problem or bipartite matching problem without considering the properties of cloud computing platform. The second group focuses on either resource allocation problem or VM provisioning problem in cloud computing platforms [2, 4, 9, 10, 17, 22, 26, 29–31]. Game theoretic approach and integer linear programming method are used to obtain an optimal resource allocation plan. However, these works are not applicable if dynamics of system are changing over time. The third group applies reinforcement learning technique in the resource allocation problem [1, 5, 15, 20], including Q-learning and hierarchical framework. Even though reinforcement learning algorithm could mitigate the problem incurred by shifting dynamic, these works fail to include both allocation fairness and properties of cloud computing platform.

7 CONCLUSION

In this paper, we propose an efficient reinforcement learning framework called MASP with reward communication mechanism for cloud quota management problem. The proposed multi-agent method with shared policy can lower the learning barrier induced by heterogeneous action space and mitigate the curse of dimensionality of action space in cloud quota management problem. Moreover, the reward communication mechanism enables MASP to balance the relative priority among multiple objectives and to achieve a great trade-off between efficiency and fairness. We evaluate the framework over three different kinds of competitors on a cloud simulation environment, parameterized by data traces collected from internal first-party cloud computing users in Microsoft. Experimental results show that the proposed MASP method can perform better on both efficiency and fairness metrics and is adaptable for efficiency and fairness trade-off by applying reward communication mechanism.

REFERENCES

[1] Enda Barrett, Enda Howley, and Jim Duggan. 2013. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Concurrency and Computation: Practice and Experience* 25 (2013).

[2] Maria Carla Calzarossa, Marco L. Della Vedova, and Daniele Tessera. 2019. A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty. *Future Generation Computer Systems* 93 (2019), 212–223. https://doi.org/10.1016/j.future.2018.10.037

[3] Karan N. Chadha, Ankur A. Kulkarni, and Jayakrishnan Nair. 2021. Efficiency fairness tradeoff in battery sharing. *Operations Research Letters* 49, 3 (2021), 377–384. https://doi.org/10.1016/j.orl.2021.04.002

[4] Hang Dong, Boshi Wang, Bo Qiao, Wenqian Xing, Chuan Luo, Si Qin, Qingwei Lin, Dongmei Zhang, Gurpreet Virdi, and Thomas Moscibroda. 2021. Predictive Job Scheduling under Uncertain Constraints in Cloud Computing. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 3627–3634. https://doi.org/10.24963/ijcai.2021/499

[5] Xavier Dutreilh, Sergey Kirgizov, Olga Melekhova, Jacques Malenfant, Nicolas Rivierre, and Isis Truck. 2011. Using Reinforcement Learning for Autonomic Resource Allocation in Clouds: towards a fully automated workflow. In *7th International Conference on Autonomic and Autonomous Systems (ICAS'2011)*. Venice, Italy, 67–74. https://hal-univ-paris8.archives-ouvertes.fr/hal-01122123

[6] Zhou Fan, Rui Su, Weinan Zhang, and Yong Yu. 2019. Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 2279–2285. https://doi.org/10.24963/ijcai.2019/316

[7] Mahendra Bhatu Gawali and Subhash K. Shinde. 2018. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing* 7, 1 (2018), 4. https://doi.org/10.1186/s13677-018-0105-8

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 1856–1865. http://proceedings.mlr.press/v80/haarnoja18b.html

[9] Noha Hamdy, Amal Elsayed Aboutabl, Nahla ElHaggar, and Mostafa-Sami M. Mostafa. 2017. Resource Allocation Strategies in Cloud Computing: Overview. *International Journal of Computer Applications* 177, 4 (2017), 18–22. https://doi.org/10.5120/ijca2017915699

[10] Giang D. Jalaparti, Virajith; Nguyen. 2010. Cloud Resource Allocation Games. In *Research and Tech Reports - Computer Science*. http://hdl.handle.net/2142/17427

[11] Jiechuan Jiang and Zongqing Lu. 2019. Learning Fairness in Multi-Agent Systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 13854–13865. https://proceedings.neurips.cc/paper/2019/hash/10493aa88605cad5ab4752b04a63d172-Abstract.html

[12] Naoki Katoh and Toshihide Ibaraki. 1998. *Resource Allocation Problems*. Springer US, Boston, MA, 905–1006. https://doi.org/10.1007/978-1-4613-0303-9_14

[13] A. Kumar and J. Kleinberg. 2000. Fairness measures for resource allocation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 75–85. https://doi.org/10.1109/SFCS.2000.892067

[14] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1509.02971

[15] Ning Liu, Zhe Li, Jielong Xu, Zhiyuan Xu, Sheng Lin, Qinru Qiu, Jian Tang, and Yanzhi Wang. 2017. A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 372–382. https://doi.org/10.1109/ICDCS.2017.123

[16] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 6379–6390. https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html

[17] Chuan Luo, Bo Qiao, Xin Chen, Pu Zhao, Randolph Yao, Hongyu Zhang, Wei Wu, Andrew Zhou, and Qingwei Lin. 2020. Intelligent Virtual Machine Provisioning in Cloud Computing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 1495–1502. https://doi.org/10.24963/ijcai.2020/208

[18] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. 2012. Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems. *The Knowledge Engineering Review* 27, 1 (2012), 1–31. https://doi.org/10.1017/S0269888912000057

[19] Quynh C. Nguyen and Richard E. Stone. 1993. A Multiperiod Minimax Resource Allocation Problem with Substitutable Resources. *Management Science* 39, 8 (1993), 964–974. http://www.jstor.org/stable/2632696

[20] Konstantinos D. Polyzos, Qin Lu, Alireza Sadeghi, and Georgios B. Giannakis. 2021. On-Policy Reinforcement Learning via Ensemble Gaussian Processes with Application to Resource Allocation. In *55th Asilomar Conference on Signals, Systems, and Computers*. 1018–1022. https://doi.org/10.1109/IEEECONF53345.2021.9723135

[21] Anupama Prasanth. 2012. Cloud Computing Services: A Survey. *International Journal of Computer Applications* 46 (2012), 975–8887.

[22] Arkaitz Ruiz-Alvarez and Marty Humphrey. 2014. Toward Optimal Resource Provisioning for Cloud MapReduce and Hybrid Cloud Applications. In *Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing*. 74–82. https://doi.org/10.1109/BDC.2014.14

[23] Karima Saidi, Ouassila Hioual, and Abderrahim Siam. 2020. Resources Allocation in Cloud Computing: A Survey. In *Smart Energy Empowerment in Smart and Resilient Cities*, Mustapha Hatti (Ed.). Springer International Publishing, Cham, 356–364.

[24] Sean R. Sinclair, Gauri Jain, Siddhartha Banerjee, and Christina Lee Yu. 2020. Sequential Fair Allocation of Limited Resources under Stochastic Demands. https://arxiv.org/abs/2011.14382

[25] Jain Anurag Singh Palvinder. 2014. Survey Paper on Cloud Computing. *International Journal of Innovations in Engineering and Technology (IJIET)* 3 (2014), 84–89.

[26] K G Srinivasa, S Srinidhi, K Sharath Kumar, Vignesh Shenvi, U Shashank Kaushik, and Kushagra Mishra. 2014. Game theoretic resource allocation in cloud computing. In *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*. 36–42. https://doi.org/10.1109/ICADIWT.2014.6814667

[27] Gerald Tesauro. 2003. Extending Q-Learning to General Adaptive Multi-Agent Systems. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf (Eds.). MIT Press, 871–878. https://proceedings.neurips.cc/paper/2003/hash/e71e5cd119bbc5797164fb0cd7fd94a4-Abstract.html

[28] Lyn C. Thomas, David B. Edelman, and Jonathan N. Crook. 2002. *Credit Scoring and Its Applications*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718317 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9780898718317

[29] Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. 2010. Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In *2010 IEEE 3rd International Conference on Cloud Computing*. 228–235. https://doi.org/10.1109/CLOUD.2010.58

[30] Zhen Xiao, Weijia Song, and Qi Chen. 2013. Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems* 24, 6 (2013), 1107–1117. https://doi.org/10.1109/TPDS.2012.283

[31] Xin Xu and Huiqun Yu. 2014. A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing. *Mathematical Problems in Engineering* 2014 (2014), 1–14. https://doi.org/10.1155/2014/915878

[32] Moshe Zukerman, Hanwu Wang, and Iradj Ouveysi. 2005. Efficiency-fairness tradeoff in telecommunications networks. *Communications Letters, IEEE* 9 (2005), 643 – 645. https://doi.org/10.1109/LCOMM.2005.1461691