



Pricing in the Competing Auction-Based Cloud Market: A Multi-agent Deep Deterministic Policy Gradient Approach

Bing Shi^{1,2(✉)}, Lianzhen Huang¹, and Rongjian Shi¹

¹ Wuhan University of Technology, Wuhan 430070, China
bingshi@whut.edu.cn

² Shenzhen Research Institute of Wuhan University of Technology,
Shenzhen 518000, China

Abstract. In the cloud market, there exist multiple cloud providers adopting auction-based mechanisms to offer cloud services to users. These auction-based cloud providers need to compete against each other to maximize their profits by setting cloud resource prices based on their pricing strategies. In this paper, we analyze how an auction-based cloud provider sets the auction price effectively when competing against other cloud providers in the evolutionary market where the amount of participated cloud users is changing. The pricing strategy is affected by many factors such as the auction prices of its opponents, the price set in the previous round, the bidding behavior of cloud users, and so on. Therefore, we model this problem as a Partially Observable Markov Game and adopt a gradient-based Multi-agent deep reinforcement learning algorithm to generate the pricing strategy. Furthermore, we run extensive experiments to evaluate our pricing strategy against the other four benchmark pricing strategies in the auction-based cloud market. The experimental results show that our generated pricing strategy can beat other pricing strategies in terms of long-term profits and the amount of participated users, and it can also learn cloud users' marginal values and users' choices of cloud providers effectively.

Keywords: Auction-based cloud market · Pricing strategy · Markov games · Multi-agent deep reinforcement learning

1 Introduction

Because of economical, scalable, and elastic access to computing resources, the development of cloud computing has achieved significant success in the industry. More and more companies and individuals prefer using computing services over the Internet. This contributes to the vigorous development of the cloud computing market. In the cloud market, there exist different types of cloud resource transaction mechanisms, such as pay as you go, subscription-based transaction. Furthermore, some cloud providers may run auction-based mechanisms to sell

resources to users, such as *Amazon's* Spot Instance. In such a context, cloud providers need to set proper transaction prices for the sale of resources. Moreover, there usually exist multiple cloud providers offering cloud resources, where cloud users can choose to participate in one of the auctions to bid for the resources. In this situation, the resource transaction prices will affect the cloud users' choices of cloud providers and bidding behavior significantly, and in turn, affect the cloud providers' profits. Furthermore, the competition among providers usually lasts for a long time, i.e. the providers compete against each other repeatedly. Therefore, in this paper, we intend to analyze how the cloud provider sets the auction price effectively in order to maximize long-term profits.

In more detail, in the environment with multiple auction-based cloud providers, each cloud user needs to determine which auction mechanism to participate in according to the choice model and then submits the bid to the cloud provider. The auction mechanism then determines the auction price. Users whose bids are not less than the auction prices obtain the resources and pay for it according to the auction prices, not their bids. In this paper, we analyze how to design an appropriate pricing strategy to set the auction price to maximize the cloud provider's profits in the environment with two cloud providers. First, we consider the evolution of the market, where the numbers and the preferences of cloud users are changing. In addition, how cloud users choose the providers and bidding, and how providers set the auction prices are affected by each other, and it is a sequential decision problem. Reinforcement learning is an effective way to solve such problems. Furthermore, this problem involves multiple providers competing against each other. This is a Markov game, which can be solved by Multi-Agent Reinforcement Learning. Specifically, we use a multi-agent deep deterministic policy gradient, named MADDPG to generate the cloud provider's pricing strategy [10]. Finally, we run experiments to evaluate our pricing strategy against four typical pricing strategies. The experimental results show that the pricing strategy generated by our algorithm can not only respond to the opponents' changing prices in time but also learn the marginal values of cloud users and users' choices on providers. Moreover, the pricing strategy generated by our algorithm can beat other strategies in terms of long-term profits.

The structure of this paper is as follows. In Sect. 3, we introduce the basic settings of cloud users and cloud providers. In Sect. 4, we describe how to use the MADDPG algorithm to generate a pricing strategy. We run extensive experiments to evaluate the pricing strategy in different situations in Sect. 5. Finally, we conclude in Sect. 6.

2 Related Work

Since cloud computing involves resource provision and consumption, auction-based mechanisms have been widely used by cloud providers for sale of resources, such as *AmazonEC2's* Spot Instance [8]. In [15], *AmazonEC2* Spot Instance mechanism was investigated from a statistical perspective. The researchers also considered the proportion of idle time for cloud service instances and proposed

an elastic Spot Instance method to ensure stable reliability revenues for providers [3]. In [6], a demand-based dynamic pricing model for Spot Instance was proposed by adopting a genetic algorithm. There also exist some works predicting the auction prices of *AmazonEC2* Spot Instances [2, 7]. In [12, 14], the authors analyzed how cloud providers using “pay as you go” set prices in the competing environment, but did not take into account the auction-based cloud market and the evolution of cloud users. In [4], the authors proposed a non-cooperative competing model which analyzed the equilibrium price of a one-shot game, but ignored the long-term profits and did not consider the auction-based mechanism as well. Actually, to the best of our knowledge, few works have considered how to set auction prices effectively in the competing environment with multiple auction-based cloud providers.

3 Basic Settings

In this section, we introduce the basic settings of cloud users and cloud providers. We assume that there are two cloud providers P_1 and P_2 in the cloud market, where they compete with each other to maximize their long-term profits. This market is constantly evolving, and we use t to denote the time stage. At the beginning of each stage, each provider publishes its auction price of the last stage. Then each user chooses to be served by a provider based on its choice model of the provider (see Sect. 3.1). However, if the user’s expected profit in both providers is negative, it may not enter any providers. After users select the cloud providers, they submit their bids. Now two providers determine the auction prices and obtain the corresponding immediate reward (see Sect. 3.2). The competition enters into the next stage.

3.1 Cloud Users

In this section, we describe the basic settings of cloud users. The amount of cloud users participating in the cloud market varies as the market evolves. Therefore, we model it as a classical logical growth function [11], which is:

$$N(t) = \frac{N_0 N_\infty}{N_0 + (N_\infty - N_0) e^{-\delta t}} \quad (1)$$

where $N(t)$ is the number of cloud users at stage t , δ is the temporal evolution rate of the market, and the initial number of cloud users is N_0 , the market is saturated when the amount of cloud users entering the market becomes stabilized, then the number of cloud users is N_∞ .

Users’ Choices of Providers. Cloud users’ choices of providers are mainly dependent on their expected utilities in the selected provider. The expected utility of cloud user j choosing to be served by provider i at stage t is:

$$u_{j,i}^t = m_j - p_{i,t} + \eta_{j,i} = v_{j,i}^t + \eta_{j,i} \quad (2)$$

where m_j is the marginal value that user j can receive from per-unit requested resource, $p_{i,t}$ is the auction price set by provider i , and we use $v_{j,i}^t = m_j - p_{i,t}$ to represent the profit that cloud user j can make when choosing provider i at stage t . $\eta_{j,i}$ means that user j has an implicit preference on provider i , which is an independently, identically distributed extreme value, and the density function is $f(\eta_{j,i}) = e^{-\eta_{j,i}} e^{-e^{-\eta_{j,i}}}$.

According to the user's expected utility in Eq. 2 and the density function, user j will choose to be served by provider i ($i' \neq i$) only if its utility is maximized. The probability of cloud user j choosing to be served by provider i at stage t is denoted as $P_{j,i}^t$:

$$P_{j,i}^t = \frac{e^{v_{j,i}^t}}{\sum_{i'} e^{v_{j,i'}^t}} \tag{3}$$

Users' Bidding Model. After each user chooses a provider, it needs to bid for the cloud resource. We adopt a bidding algorithm based on a feedback control system, where cloud users utilize a feedback loop to automatically adjust the submitted bids [1], which is shown in Fig. 1.

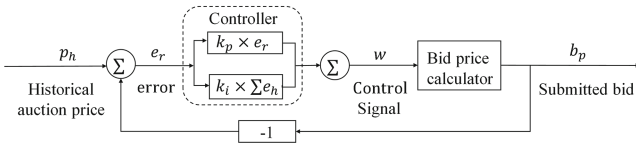


Fig. 1. Cloud users' bidding algorithm

The user's submitted bid for the next stage is b_p :

$$b_p = p_l + \frac{p_u - p_l}{\pi} \times \text{arccot}(w) \tag{4}$$

where p_l and p_u are the lower and upper bound of the cloud service instance respectively, w is a control signal to adjust the user's bid appropriately. The range of $\text{arccot}(w)$ is $(0, \pi)$, and thus the user's bid b_p is constrained in (p_l, p_u) .

Note that w consists of two parts, which is the current proportional error w_p , and the historical accumulated errors $w_i(t)$:

$$w_p = k_p \times e_r \quad (k_p < 0, p_l - p_u < e_r < p_u - p_l) \tag{5}$$

where k_p is the proportional gain of the control signal. e_r is defined as the difference between the submitted bid at stage t and the auction price of the cloud service instance at stage t , i.e. $e_r = p_h - b_p|_{h=t}$. Therefore e_r is in $(p_l - p_u, p_u - p_l)$.

Since historical errors contain more information to help users to improve their bidding behavior and win bids, we decide to use an integral controller to further study the historical errors, which can be expressed as:

$$w_i(t) = k_i \times \int_0^t e(\tau) d\tau \approx k_i \times \sum_{h=0}^t e_h \quad (k_i < 0, p_l - p_u < e_h < p_u - p_l) \quad (6)$$

where k_i is the integral gain of the control signal, e_h is the historical error at stage h ($0 \leq h \leq t$). Based on Eq. 5 and Eq. 6, we can calculate the control signal w , that is: $w = w_p + w_i \approx k_p \times e_r + k_i \times \sum_{h=0}^t e_h$.

3.2 Cloud Providers

In this section, we introduce the basic settings of cloud providers. Cloud providers incur costs when providing cloud services. Similar to the work in [5], the marginal cost of provider i in a per-unit cloud service at stage t is:

$$c_{i,t} = c_{i,0} \times \left(\sum_{j \in N_{i,t}} d_{j,t} \right)^{-\beta e^{-\rho t}} \quad (7)$$

This equation indicates that the marginal cost of provider i will decrease when the number of cloud users $N_{i,t}$ in demands of cloud services $d_{j,t}$ increase at stage t , where $c_{i,0}$ is the initial cost of cloud provider i , $\beta > 0$ and $\rho > 0$ are two parameters to control the decreased marginal cost when users' demands increase. We then compute the provider's immediate payoff(reward), which is:

$$r_{i,t} = \sum_{j \in N_{i,t}} d_{j,t} \times (p_{i,t} - c_{i,t}) \quad (8)$$

Its long-term profits, which are the discounted cumulative profits over all stages, is calculated as: $R_i = \sum_{t=0}^T \gamma^t r_{i,t}$.

4 MADDPG Algorithm

In this section, we describe how to model the issue as a Partially Observable Markov Game and use MADDPG to solve it to generate a pricing strategy.

4.1 Partially Observable Markov Game

In this paper, two cloud providers repeatedly competing with each other to maximize their profits, which is a sequential-decision problem. Furthermore, since cloud providers and users cannot perceive all information of the world, it is a partially observable Markov game [9].

In more detail, this Markov game consists of a set of states S describing the cloud market, a set of pricing actions A_1, A_2 and a set of the observed states

O_1, O_2 for each provider. We use $s = (p_{avg}^1, p_{sd}^1, p_{avg}^2, p_{sd}^2, b_{avg}^1, b_{sd}^1, b_{max}^1, b_{min}^1, b_{mid}^1, b_{avg}^2, b_{sd}^2, b_{max}^2, b_{min}^2, b_{mid}^2, n_{1,t}, n_{2,t}, c_{1,t}, c_{2,t}) \in S$ to denote a state. For cloud provider P_i , the average and standard deviation of its auction prices over a period of time are p_{avg}^i, p_{sd}^i respectively. From the cloud users' bids, we can compute the average, standard deviation, maximum, minimum, and median value of the bids, which are $b_{avg}^i, b_{sd}^i, b_{max}^i, b_{min}^i, b_{mid}^i$ respectively. We use $n_{i,t}$ to represent the number of cloud users choosing to be served by provider i at stage t , and use $c_{i,t}$ to denote the marginal cost of provider i at stage t . Note that in the realistic cloud market, the cloud providers' auction prices over a period of time are usually accessible to users. However, the number of cloud users choosing to be served by provider i and users' bids are usually not public. That is, $p_{avg}^1, p_{sd}^1, p_{avg}^2, p_{sd}^2$ are shared public information of all providers, but $b_{avg}^i, b_{sd}^i, b_{max}^i, b_{min}^i, b_{mid}^i, n_{i,t}, c_{i,t}$ are private information hidden to the other cloud provider. Therefore, the observation of provider P_i is $o_i = (p_{avg}^1, p_{sd}^1, p_{avg}^2, p_{sd}^2, b_{avg}^i, b_{sd}^i, b_{max}^i, b_{min}^i, b_{mid}^i, n_{i,t}, c_{i,t}) \in O_i$.

Then we use $\pi_{\theta_i} : O_i \times A_i \rightarrow [0, 1]$ to present the pricing strategy of provider i . After providers take pricing actions, the state transfers to the next state according to the state transfer function $\Delta : S \times A_1 \times A_2 \rightarrow S'$, then each provider can obtain the immediate reward $r_{i,t} : S \times A_i \rightarrow \mathbb{R}$, and obtain the corresponding observation $o_i : S \rightarrow O_i$ of the next state. Given the immediate reward made at stage t , the cloud provider can maximize the long-term profits through an efficient pricing strategy.

4.2 Multi-Agent Deep Deterministic Policy Gradient

In this section, we introduce how to use MADDPG to generate a pricing strategy in the competing environment with two cloud providers. MADDPG is a multi-agent reinforcement learning algorithm based on the Actor-Critic framework proposed by *OpenAI*, where Actor is a probability-based actuator, while Critic evaluates every action of Actor to modify the weight of Actor. When the critic of MADDPG evaluates the actors' actions, it not only considers themselves but also the rest of the agents [10].

Specifically, the two cloud providers whose strategies $\pi_{\theta} = \{\pi_1, \pi_2\}$ are parameterized by $\theta = \{\theta_1, \theta_2\}$. Then the gradient of expected return $J(\theta_i) = E[R_i]$ of cloud provider i is:

$$\nabla_{\theta_i} J(\theta_i) = E_{s \sim p^{\mu}, a_i \sim \pi_i} [\nabla_{\theta_i} \log \pi_i(a_i | o_i) Q_i^{\pi_{\theta}}(x, a_1, a_2)] \quad (9)$$

where p^{μ} is the state distribution, $x = (o_1, o_2)$ is the observed value of all cloud providers. $Q_i^{\pi_{\theta}}(x, a_1, a_2)$ is a centralized value function and its input contains not only some observed information x , but also all providers' actions a_1, a_2 . When Eq.9 is extended to a deterministic policy, we use μ_{θ_i} w.r.t. parameter θ_i to represent the provider's strategy. Then its gradient can be written as:

$$\nabla_{\theta_i} J(\mu_{\theta_i}) = E_{s, a \sim D} [\nabla_{\theta_i} \mu_{\theta_i}(a_i | o_i) \nabla_{a_i} Q_i^{\mu}(x, a_1, a_2) |_{a_i = \mu_{\theta_i}(o_i)}] \quad (10)$$

where D is the experience replay buffer contains tuple $(x, a_1, a_2, r_1, r_2, x')$, in which r_1, r_2 are the immediate rewards, and x' is the two providers' observations in the next stage. The centralized action-value function Q_i^μ is updated as:

$$L(\theta_i) = E_{x,a,r,x'} \left[(Q_i^\mu(x, a_1, a_2) - y)^2 \right], y = r_i + \gamma Q_i^{\mu'}(x', a'_1, a'_2) \Big|_{a'_j = \mu'_{\theta'_j}(o_j)} \quad (11)$$

where $\mu' = \{\mu_{\theta'_1}, \mu_{\theta'_2}\}$ is the set of target policies with the delayed parameter θ'_i .

Updating the value function in Eq. 11 requires the pricing strategy of the opponent provider. However, the opponent's pricing strategy is usually private in the realistic environment, and thus hard to be known. Therefore each cloud provider can only estimate the opponent j 's pricing strategy $\hat{\pi}_{\varphi_i^j}$ with φ parameter instead. This approximated strategy is learned by maximizing the log probability of provider j 's actions with an entropy regularizer, which is:

$$L(\varphi_i^j) = -E_{o_j, a_j} \left[\log \hat{\pi}_{\varphi_i^j}(a_j | o_j) + \lambda H(\hat{\pi}_{\varphi_i^j}) \right] \quad (12)$$

where $H(\hat{\pi}_{\varphi_i^j})$ is the entropy of the policy distribution. Now y in Eq. 11 can be replaced by the approximated value \hat{y} :

$$\hat{y} = r_i + \gamma Q_i^{\mu'}(x', \pi'_{\varphi_i}(o_i), \hat{\pi}'_{\varphi_i^j}(o_j)), \quad i \neq j \quad (13)$$

where $\hat{\pi}'_{\varphi_i^j}(o_j)$ is the target network of the approximate policy $\hat{\pi}_{\varphi_i^j}$.

To improve the robustness of agents' strategies, sub-strategy will be used to enhance the adaptability of agents. Therefore, in each round of a game, the cloud provider randomly selects a sub-strategy to execute from a set that contains K different sub-strategies. For cloud provider i , the goal is to maximize the ensemble objective, which is:

$$J_e(\mu_{\theta_i}) = E_{k \sim \text{unif}(1, K), s \sim p^\mu, a \sim \mu_{\theta_i^{(k)}}} [R_i(s, a)] \quad (14)$$

where μ_{θ_i} is a set of K different sub-strategies, and $\mu_{\theta_i^{(k)}}$ represents an element in this set. Consequently, the gradient of ensemble objective w.r.t $\theta_i^{(k)}$ is:

$$\nabla_{\theta_i^{(k)}} J_e(\mu_{\theta_i}) = \frac{1}{K} E_{x,a \sim D_i^{(k)}} \left[\nabla_{\theta_i^{(k)}} \mu_{\theta_i^{(k)}}(a_i | o_i) \nabla_{a_i} Q^{\mu_{\theta_i}}(x, a_1, a_2) \Big|_{a_i = \mu_{\theta_i^{(k)}}(o_i)} \right] \quad (15)$$

where $D_i^{(k)}$ is the replay buffer for each sub-strategy $\mu_{\theta_i^{(k)}}$ of agent i .

5 Experiments

5.1 Parameter Settings

In this paper, two cloud providers P_1 and P_2 can set the auction prices in the range of $[10, 100]$. Each round has 200 stages. We set the number of cloud users

at initial stage $t = 0$ is $N_0 = 100$, at saturation stage $t = 200$ is $N_\infty = 1000$, the temporal evolution rate of the market is $\delta = 0.07$. The users' marginal values follows a uniform distribution within $[40, 70]$. Then we use two queues $Queue_1, Queue_2$ to store the two providers' historical auction prices, and the length of the queue is $len = 10$. The lower bound price of the cloud service instance equals to the lowest price that the provider can set, i.e. $p_l = 10$, and the upper bound price of the cloud service instance equals to the marginal value that cloud user j can obtain from per-unit requested resource, i.e. $p_u = m_j$. k_p and k_i in the users' bidding model follow a uniform distribution within $(-0.1, 0)$. We set $\beta = 0.01$, $\rho = 0.02$ and $c_{i,0} = 8.0$, and the users' demands for cloud resources follow a uniform distribution of $d_{j,t} \sim U[1, 3]$.

5.2 Training

In this section, we generate a pricing strategy that can maximize the cloud provider's long-term profits in the competing cloud market. The same as the work done in [13], we consider the fictitious self-playing which can learn the optimal pricing strategy from scratch. Therefore, we use MADDPG with fictitious self-playing to train our agents. After training, a pricing strategy based on MADDPG is shown in Fig. 2. From this figure, we find that the prices set by the two cloud providers P_1, P_2 at each stage converge in $[10, 30]$, which is less than the highest auction price range $[40, 70]$ that cloud users can accept. It further indicates that the MADDPG algorithm can learn the marginal values of cloud users, and set the prices a bit lower than the marginal values of most users. By doing this, the cloud provider can maximize its profits while keeping cloud users.

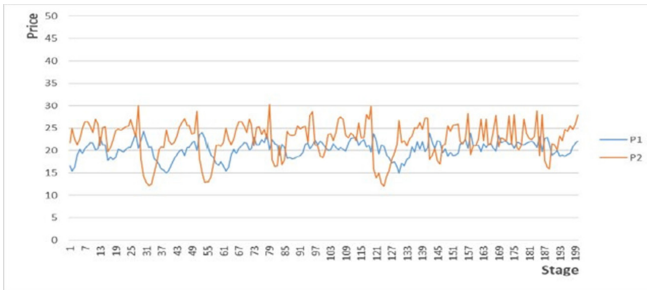


Fig. 2. MADDPG's pricing strategy

5.3 Strategy Evaluation

In this section, we run experiments to evaluate our pricing strategy against four typical pricing strategies, and we evaluate the pricing strategy by using these metrics: auction price set by the pricing strategy, cloud user ratio which is the ratio of the number of cloud users entering in the provider to the total number of users, and cumulative profits which is the long-term profits made by the provider across all stages.

Vs. Random Pricing Strategy. In the market, there exist some fresh competing cloud providers who may explore the market by adopting a random pricing strategy to obtain more information. Therefore, we first evaluate our pricing strategy against the competing provider adopting a random pricing strategy of uniform distribution. The results are shown in Fig. 3, we find that the provider using the MADDPG pricing strategy can attract more cloud users and obtain more cumulative profits than the opponent using a random pricing strategy.

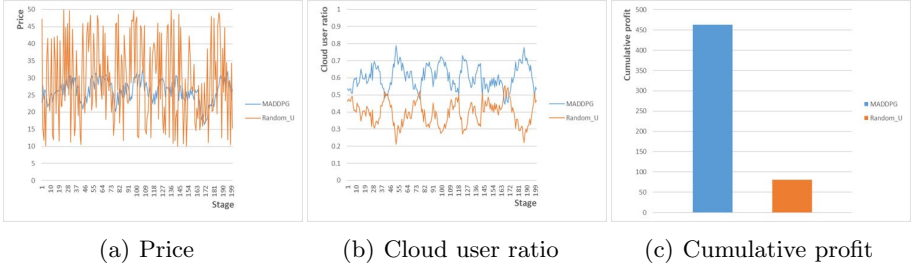


Fig. 3. MADDPG vs. Random (uniform distribution)

Vs. Price Reduction Strategy. Some cloud providers may keep reducing the prices to attract cloud users in the cloud market. We consider two kinds of price reduction strategies, named Linear Reduction strategy (RecL) and Exponential Reduction strategy (RecE) where RecL decreases the price linearly while RecE decreases the price rapidly in the initial stages and then becoming smooth when approaching the threshold price. The results are shown in Fig. 4 and Fig. 5. From the experiments, we find that our provider using MADDPG can adjust the price in time to adapt to the changes of the opponent, and thus make the cumulative profits at a higher level.

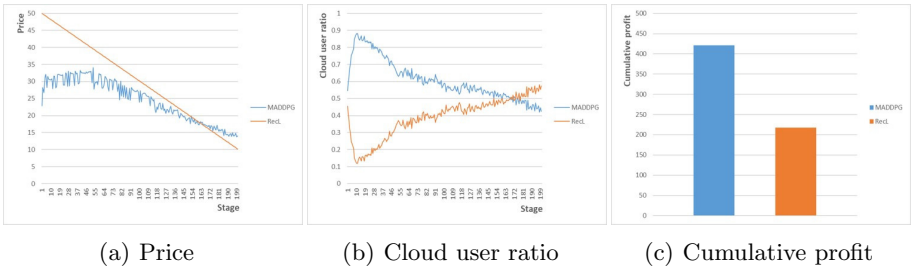


Fig. 4. MADDPG vs. Linear Reduction

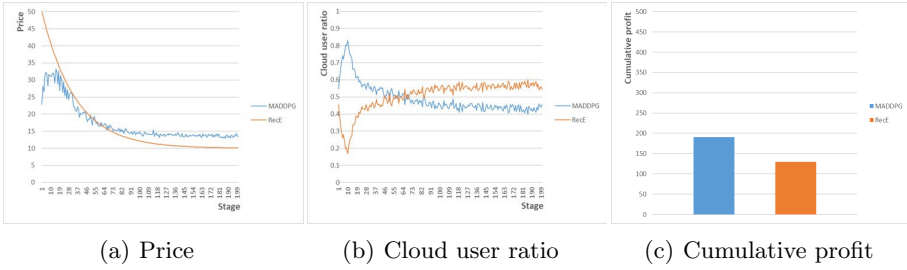


Fig. 5. MDDPG vs. Exp reduction

Vs. Greedy Pricing Strategy. Similarly, some cloud providers may adopt a greedy pricing strategy, which only focuses on the immediate reward of each stage, regardless of long-term profits. Therefore, we set the discount factor γ to 0 in MDDPG. The results are shown in Fig. 6. We find that the price of the greedy strategy is slightly higher, so the number of cloud users attracted by the provider using the MDDPG pricing strategy is higher. Again our pricing strategy can beat the greedy pricing strategy in terms of cumulative profits.

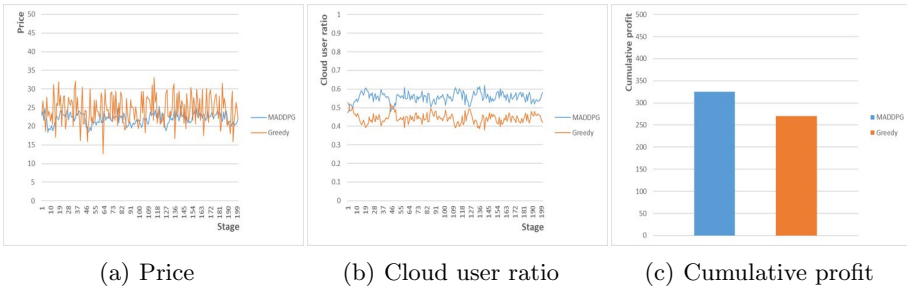


Fig. 6. MDDPG vs. Greedy

Vs. M-MDDPG Pricing Strategy. To further demonstrate the effectiveness of the pricing strategy generated by MDDPG algorithm, we train a new pricing strategy against itself, and we name it as M-MDDPG. The results are shown in Fig. 7. We can see that the provider using the M-MDDPG pricing strategy has almost the same cumulative profits as that in the MDDPG pricing strategy. This means that even though the opponent can train a particular pricing strategy against the MDDPG pricing strategy, it still cannot beat our pricing strategy.

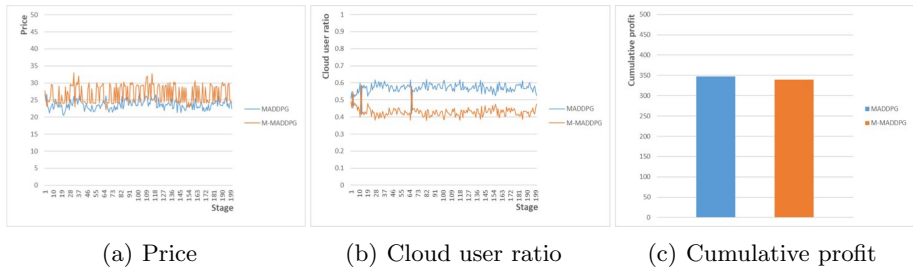


Fig. 7. MADDPG vs. M-MADDPG

6 Conclusion

In this paper, we use the gradient-based multi-agent deep reinforcement learning algorithm to generate a pricing strategy for the competing cloud provider. We also run extensive experiments to evaluate our pricing strategy against the other four typical pricing strategies in terms of long-term profits. Experimental results show that MADDPG based pricing strategy can not only beat the opponent's pricing strategy effectively but also learn the marginal values of cloud users and users' choices of providers. Our work can be used to provide useful insights on designing practical pricing strategies for competing cloud providers.

Acknowledgement. This paper was funded by the Humanity and Social Science Youth Research Foundation of Ministry of Education (Grant No. 19YJC790111), the Philosophy and Social Science Post-Foundation of Ministry of Education (Grant No. 18JHQ060) and Shenzhen Fundamental Research Program (Grant No. JCYJ20190809175613332).

References

1. Åström, K.J., Murray, R.M.: *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, Princeton (2010)
2. Cai, Z., Li, X., Ruiz, R., Li, Q.: Price forecasting for spot instances in cloud computing. *Future Gener. Comput. Syst.* **79**, 38–53 (2018)
3. Dawoud, W., Takouna, I., Meinel, C.: Reliable approach to sell the spare capacity in the cloud. In: *Cloud Computing*, pp. 229–236 (2012)
4. Feng, Y., Li, B., Li, B.: Price competition in an oligopoly market with multiple IaaS cloud providers. *IEEE Trans. Comput.* **63**(1), 59–73 (2013)
5. Jung, H., Klein, C.M.: Optimal inventory policies under decreasing cost functions via geometric programming. *Eur. J. Oper. Res.* **132**(3), 628–642 (2001)
6. Kansal, S., Kumar, H., Kaushal, S., Sangaiah, A.K.: Genetic algorithm-based cost minimization pricing model for on-demand IaaS cloud service. *J. Supercomput.* **76**, 1–26 (2018)
7. Khandelwal, V., Gupta, C.P., Chaturvedi, A.K.: Perceptive bidding strategy for Amazon EC2 spot instance market. *Multiagent Grid Syst.* **14**(1), 83–102 (2018)

8. Kumar, D., Baranwal, G., Raza, Z., Vidyarthi, D.P.: A survey on spot pricing in cloud computing. *J. Netw. Syst. Manage.* **26**(4), 809–856 (2018)
9. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: *Proceedings of the Eleventh International Conference on Machine Learning (ML 1994)* (1994)
10. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: *Advances in Neural Information Processing Systems*, pp. 6379–6390 (2017)
11. Pearl, R., Reed, L.J.: On the rate of growth of the population of the united states since 1790 and its mathematical representation. *Proc. Nat. Acad. Sci. U.S.A.* **6**(6), 275 (1920)
12. Shi, B., Zhu, H., Wang, J., Sun, B.: Optimize pricing policy in evolutionary market with multiple proactive competing cloud providers. In: *Proceedings of the IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI 2017)*, pp. 202–209. IEEE (2017)
13. Silver, D., et al.: Mastering the game of go without human knowledge. *Nature* **550**(7676), 354–359 (2017)
14. Truong-Huu, T., Tham, C.K.: A game-theoretic model for dynamic pricing and competition among cloud providers. In: *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pp. 235–238. IEEE (2013)
15. Zheng, L., Joe-Wong, C., Tan, C.W., Chiang, M., Wang, X.: How to bid the cloud. *ACM SIGCOMM Comput. Commun. Rev.* **45**(4), 71–84 (2015)