# Competitive Pricing for Cloud Computing in an Evolutionary Market

# (Extended Abstract)

Bolei Xu
The University of Nottingham
Ningbo China
bolei.xu@nottingham.edu.cn

Tao Qin
Microsoft Research Asia
taoqin@microsoft.com

Guoping Qiu
The University of Nottingham
Ningbo China
guoping.qiu@nottingham.edu.cn

Tie-Yan Liu
Microsoft Research Asia
tyliu@microsoft.com

## ABSTRACT

We study the problem of how to optimize a cloud service provider's pricing policy so as to better compete with other providers. Different from previous work, we take both the evolution of the market and the competition between multiple cloud providers into consideration while optimizing the pricing strategy for the provider. In order to compute the optimal pricing policy, we decompose the optimization problem into two steps: (1) When the market finally becomes saturated, we use Q-learning, a method of reinforcement learning, to derive an optimal pricing policy for the stationary market; (2) Based on the optimal policy for the stationary market, we use backward induction to derive an optimal pricing policy for the situation of competition in an evolutionary market. Numerical simulations demonstrate the effectiveness of our proposed approach.

## Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Social and Behavioral Sciences**]: Economics; K.1 [**The Computer Industry**]: Markets

## General Terms

Economics, Algorithms, Experimentation

## Keywords

Cloud computing, market competition, dynamic pricing, reinforcement learning

## 1. INTRODUCTION

In this work, we study the problem of pricing policy optimization in the cloud market by considering both the competition between providers and the evolution of the market. Specifically, we first model an evolutionary cloud computing market where the growth rate of the number of users is changing over different market stages and the demand of individual users is affected by the price. Then we formulate the market competition as a multiple-stage game in which we assume an active provider is competing with other reactive competitors who adopt the follow-up pricing policy. When performing pricing optimization for the active provider under the above settings, to handle the evolution of the cloud market, we decompose the optimization problem into two steps. In the first step, we consider the situation when the market evolves to saturation (stationary market) and apply a well-established method of reinforcement learning — Q-learning, to find out an optimal policy for the provider. In the second step, based on the optimal policy obtained for the stationary market, we use backward induction to find an optimal pricing policy for the situation when the market is still evolving and the environment is non-stationary. Numerical simulations demonstrate the effectiveness of our proposed approach.

**Appears in:** *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*


## 2. PROBLEM MODELING

We first use the classical logistic growth function [2] to model the growth of the number of cloud service users: $N(t) = \frac{N_0 N_\infty}{N_0+(N_\infty-N_0)e^{-\kappa t}}$, where $N_0$ is the initial population, $N(t)$ denotes the number of cloud users in the market at stage $t$, $N_\infty$ is the saturated population in the market, and $\kappa$ is the temporal evolution rate of the market. We then assume the demand of an individual user negatively correlates with the market price. Let $d_{j,t}$ denote the demand of user $j$ at stage $t$, which is a random variable that follows an exponential distribution with expectation$\lambda_t = \frac{\Lambda}{\frac{1}{K}\sum_{i=1}^{K} p_{i,t-1}}$, where $p_{i,t-1}$ is the price set by provider $i$ at stage $t-1$, $\Lambda$ is a constant parameter. When the demand of a cloud user is satisfied, he/she can extract value from the cloud service and need to pay the cloud provider. Let $\theta_j$ denote the marginal value that the user can extract from per-unit of the cloud service. The utility $u^t_{j,i}$ of the user $j$ by choosing provider $i$ is defined as$u^t_{j,i} = d_{j,t}(\theta_j - p_{i,t} + \tau_{j,i})$, where $\tau_{j,i}$ denotes user $j$'s preference towards provider $i$. We model the marginal operational cost for a provider as $c_{i,t} = c_{i,0}(\sum_{j\in\mathcal{N}_{i,t}} d_{j,t})^{-\beta} e^{-\eta t}$, where $c_{i,0}$ is the initial cost of provider $i$, $\mathcal{N}_{i,t}$ is the set of users choosing provider $i$ at stage $t$, and $\beta > 0$ and $\eta > 0$ are two parameters. The immediate profit of provider $i$ at stage $t$ can be written as $r_{i,t} = \sum_{j\in\mathcal{N}_{i,t}} d_{j,t}(p_{i,t} - c_{i,t})$.

# 3. PRICING POLICY OPTIMIZATION

Because we are considering an evolutionary market, we cannot directly apply reinforcement learning as done by [3] to our problem. In particular, we notice that as the cloud market evolves, it will eventually become mature and stationary. On this basis, we can decompose the solution of an optimal pricing policy into two steps: applying reinforcement learning techniques to find an optimal pricing policy when the market becomes stationary, and then using backward induction to find an optimal pricing policy when the market is still evolving.

For the Q-learning, the state $s$ is the price of the proactive provider in the previous stage and we denote $Q(s,a)$ as the discounted long-run expected profit when the proactive provider takes action $a$ in state $s$. We use a look-up table to represent the $Q(s,a)$ function [1]. Then we repeat the following update procedure. For each state $s$, we randomly take an action $a$ with probability $\epsilon$ and take the action $a$ maximizing $Q(s,a)$ with probability $1-\epsilon$. After taking action $a$ in state $s$, the users choose providers and the proactive provider receives an immediate reward $r(s,a)$ and the new state is denoted as $s'$. Then we update $Q(s,a)$ in the table by adding $\Delta Q(s,a) = \alpha[r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$, where $\gamma$ is the discount factor and $\alpha$ is the learning rate. We terminate the update procedure if the table converges. When we obtain the convergent table, the optimal policy can be easily found by choosing the action with the highest expected profit $Q(s,a)$ in any state $s$.

After obtaining the optimal pricing policy when the market is stationary and the optimal expected profit for each state-action pair, we adopt backward induction to find the optimal pricing policy when the market is still evolving by leveraging the convergent table $Q$. Let $T$ denote a stationary stage of the market, $r(p,a,t)$ denote the immediate profit of the proactive provider at stage $t$ by taking action $a$ (setting new price as $a$) given the price $p$ of the previous stage $t-1$, $R(p,t)$ denotes the optimal expected profit that the proactive provider can achieve at stage $t$ given the price $p$ of the previous stage, and $A(p,t)$ denote the optimal action (price) that the proactive provider should take at stage $t$ given the price $p$ of the previous stage. Algorithm 1 shows how backward induction can find an optimal pricing policy for an evolutionary market.

**Algorithm 1** Finding an optimal pricing policy by backward induction when the market is evolving

**Input:**
Stationary stage $T$ and the convergent table $Q$ outputted by Q-learning;
The initial price $p_{i,0}$ and initial cost $c_{i,0}$ of each provider $i$ before stage 1;

**Output:**
An expected profit table $R(p,t), \forall p \in \mathcal{P}, \forall t \in [T]$ and an optimal action table $A(p,t), \forall p \in \mathcal{P}, \forall t \in [T]$;

```
Set R(p,T) = max_{a∈P, a≤p} Q(p,a);
Set A(p,T) = arg max_{a∈P, a≤p} Q(p,a);
for t = T−1, T−2, ··· , 2, 1 do
    for each p ∈ P do
        Set R(p,t) = max_{a∈P, a≤p} {r(p,a,t) + γR(a,t+1)};
        Set A(p,t) = arg max_{a∈P, a≤p} {r(p,a,t) + γR(a,t+1)};
    end for
end for
```

# 4. NUMERICAL SIMULATIONS

For the simulations in this subsection, we consider a market with three providers. Suppose Provider 1 is the proactive one and other two reactive ones are named as Provider 2 and Provider 3 respectively. We consider Providers 2 and 3 applying either absolute follow-up policy (ABS) that cut the price down by an absolute number or relative follow-up policy (REL) that cut the price down by an relative number. As this paper is the first one to model the evolutionary cloud computing market, there are no existing baselines, and we create two simple price reduction policies to serve as baselines. The first pricing policy called "Exp Reduction" that asks the proactive provider to exponentially cut his/her price over stages $p_t = p_o e^{-0.01t}$, and the second policy asks the proactive provider to drop the price down by 5% for every 10 steps until it approaches the threshold price. We call it "Linear Reduction". We simulate Provider 1's performance according to three different initial market scenarios that Provider 1 has the highest, medium and lowest initial price, marginal cost and preference level from users. According to the Table 1, using the pricing policy outputted by our algorithms, the proactive provider can achieve the largest profit than using the other two policies for all the scenarios.

| | | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|---|
| ABS | Our Policy | **1488** | **932** | **703** |
| | Exp | 980 | 674 | 453 |
| | Linear | 1100 | 772 | 487 |
| REL | Our Policy | **1450** | **916** | **691** |
| | Exp | 919 | 650 | 432 |
| | Linear | 1089 | 751 | 468 |

**Table 1: Profit comparison of different price reduction policies**


## Acknowledgments

This work is partially supported by Ningbo Science and Technology Bureau (Project No 2012B10055 and 2013D10008) and by the International Doctoral Innovation Centre (IDIC) at the University of Nottingham Ningbo China.


## REFERENCES


[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[2] R. Pearl and L. J. Reed. On the rate of growth of the population of the united states since 1790 and its mathematical representation. *Proceedings of the National Academy of Sciences of the United States of America*, 6(6):275, 1920.

[3] T. Truong-Huu and C.-K. Tham. A game-theoretic model for dynamic pricing and competition among cloud providers. In *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*, pages 235–238. IEEE, 2013.