# **Evolutionary Stable Resource Pricing Strategies**

Bo An Dept. of Computer Science University of Massachusetts, Amherst, USA ban@cs.umass.edu Athanasios V. Vasilakos Dept. of Computer and Telecom. Eng., University of Western Macedonia, Greece vasilako@ath.forthnet.gr Victor Lesser Dept. of Computer Science University of Massachusetts, Amherst, USA lesser@cs.umass.edu

# ABSTRACT

Enterprises currently employ Cloud services to improve the scalability of their services and resource providers strategically price resources to maximize their utilities. While Nash equilibrium is the dominant concept for studying such kind of interaction, evolutionary game theory seems more appropriate for modeling agents' strategic interactions as it relaxes many strong assumptions. This paper applies evolutionary dynamics to generate resource providers' evolutionary stable strategies. We present a sequential monte carlo approach for simulation of multi-population evolutionary dynamics in which each agent's strategy space is continuous. We use resampling and Gaussian smoothing to prevent degeneration of particle samples. Simulation results show that the proposed approach always converges to evolutionary stable strategies. Our approach is general in that it can be used to generate agents' evolutionary stable strategies for other resource allocation games.

## Keywords

Cloud computing, resource allocation, evolutionary game theory

# 1. INTRODUCTION

Cloud computing is a new paradigm of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. As resource consumers rely on Cloud providers to supply their resource needs, Cloud providers will need to consider and meet different resource requirements of each individual consumer. To achieve this, Cloud providers can no longer continue to deploy traditional system-centric resource management architecture that do not provide incentives for them to share their resources and still regard all service requests to be of equal importance [1]. Instead, market-oriented mechanisms are necessary to regulate the supply and demand of resources. Recently, many research projects have proposed a variety of market structures for Cloud resource allocation.

This paper develops new techniques for generating evolutionary stable strategies for resource allocation problems in Cloud markets consisting of selfish resource consumers and resource providers. Resource consumers (buyers) have tasks to finish and each buyer has multiple functionally equivalent plans for its high level task and each plan consists of a set of resources. Resource providers (sellers) sell resources to buyers following a simple protocol: sellers announce their prices first and then buyers determine the set of resources to buy. One immediate question arising from the resource allocation game is to finding agents' equilibrium strategies, which is important for predicting agents' selfish behavior and optimizing system level objectives. We apply evolutionary dynamics to generate agents' evolutionary stable strategies. Since agents' strategy space is continuous, we present a sequential monte carlo approach for simulation of multi-population evolutionary dynamics.

#### 2. RESOURCE ALLOCATION PROBLEM

We consider a resource allocation game among agents  $\mathcal{A} = \mathcal{B} \cup \mathcal{S}$  where  $\mathcal{B}$  is the set of buyers and  $\mathcal{S}$  is the set of sellers. For ease of analysis, we assume that each seller  $\mathbf{s} \in \mathcal{S}$  can only provide one resource  $r_{\mathbf{s}}$  and  $\mathbf{s}$  can serve multiple buyers simultaneously. Let  $c_{\mathbf{s}}$  be seller  $\mathbf{s}$ 's (marginal) cost for providing resource  $r_{\mathbf{s}}$  to a buyer and the cost is the same for different resource consumers. Let  $\mathcal{R} = \bigcup_{\mathbf{s} \in \mathcal{S}} r_{\mathbf{s}}$  be the set of resources provided by all sellers  $\mathcal{S}$ . Each buyer  $\mathbf{b} \in \mathcal{B}$  has a high level task and there are multiple plans  $\mathcal{P}_{\mathbf{b}}$  for the task. Each plan  $P \in \mathcal{P}_{\mathbf{b}}$  consists of a set of resources  $\mathcal{R}(P) \subseteq \mathcal{R}$ . Let  $v_{\mathbf{b}}$  be buyer  $\mathbf{b}$ 's value of finishing its task, which is also the highest price  $\mathbf{b}$  can pay to buy resources (services).

Buyers and sellers interact with each other following a simple contracting protocol: Sellers announce their prices first. Then each buyer will decide whether to buy a seller's resource or not. Given sellers' prices  $\theta$ , buyer b's utility  $u_{\mathbf{b}}(\theta)$  is the difference between its gain of completing its high level task and its cost of buying resources. Similarly, seller s's utility  $u_{\mathbf{s}}(\theta)$  is the difference between its received payment and its cost of providing resources. Since sellers announce their prices simultaneously, there exists no optimal strategy for each agent and it leads us to study agents' equilibrium strategies. Given the two stage protocol, we can represent each buyer's optimal action as a function of sellers' strategies. Therefore, we only need to focus on sellers' equilibrium strategies.

An evolutionary stable strategy (ESS) is a strategy which, if adopted by a population of players, cannot be invaded by any alternative strategy that is initially rare. An ESS is a Nash equilibrium which is "evolutionary" stable meaning that once it is fixed in a population, natural selection alone is sufficient to prevent alternative (mutant) strategies from successfully invading [3].

# 3. COMPUTING EVOLUTIONARY STABLE STRATEGIES

Here we present a design of evolutionary dynamics using the sequential Monte Carlo methods. The replicator dynamics is conducted in discrete time indexed by  $\{0, 1, \ldots\}$ . There are multiple populations, each representing a seller. Let the population for seller s at time t be  $\delta_{\mathbf{s}}(t)$ . Each population consists of N samples, i.e.,  $\delta_{\mathbf{s}}(t) = \{\delta^{1}_{\mathbf{s}}(t), \ldots, \delta^{N}_{\mathbf{s}}(t)\}$  where  $\delta^{i}_{\mathbf{s}}(t)$  represents a price in the range  $[c_{\mathbf{s}}, \infty)$ . The state of populations at time t is  $\delta(t) = \bigcup_{\mathbf{s} \in S} \delta_{\mathbf{s}}(t)$ . The population  $\delta_{\mathbf{s}}(t)$  for each seller s is evolved from the population state  $\delta_{\mathbf{s}}(t-1)$  at time t-1 following the procedure outlined in Algorithm 1.

Initialization: The initial population  $\delta_{\mathbf{s}}(0)$  for seller  $\mathbf{s}$  can be

Initialization: At time t = 0, randomly choose N samples δ<sub>s</sub>(0) from the initial distribution.
Importance Weight Evaluation: Evaluate the importance weight

 $w_{\mathbf{s}}^{i}(t)$  of each sample  $\delta_{\mathbf{s}}^{i}(t)$ .

3) **Resampling** : Resample N samples  $\delta_{\mathbf{s}}^{i}(t+1)$  from set  $\delta_{\mathbf{s}}(t)$ 

according to the normalized weight of all samples in  $\delta_{\mathbf{s}}(t)$ .

**4) Smoothing**: Add a small smoothing term sampled from a Gaussian distribution to each sample  $\delta_{s}^{i}(t+1)$ . Set t = t+1 and go to step 2).

Algorithm 1: SMC Model of Replicator Dynamics

randomly chosen from its strategy space  $(c_s, \varpi_s]$  where  $\varpi_s$  is the maximum price seller s can propose.  $\varpi_s$  can be set as the maximum value of all buyers.

Importance Weight Evaluation: After generating a new population  $\delta_{\mathbf{s}}(t)$  at time t, the weight  $w_{\mathbf{s}}^{i}(t)$  of each sample  $\delta_{\mathbf{s}}^{i}(t)$  is evaluated based on its interaction with other populations. Let  $\delta_{-\mathbf{s}}(t) = \prod_{\mathbf{s}' \in \{S-\mathbf{s}\}} \delta_{\mathbf{s}'}(t)$  be the all strategy profiles of all sellers other than  $\mathbf{s}$ . The weight  $w_{\mathbf{s}}^{i}(t)$  is averaged over all utilities it can get with all possible strategy combination of other sellers. Formally,

$$w_{\mathbf{s}}^{i}(t) = \frac{\sum_{\theta_{-s} \in \delta_{-\mathbf{s}}(t)} u_{\mathbf{s}}(\delta_{\mathbf{s}}^{i}(t), \theta_{-s})}{|\delta_{-\mathbf{s}}(t)|}$$

where  $|\delta_{-s}(t)|$  is the number of strategy combinations for all sellers other than s.

Resampling: At step 3, the weights of all samples  $\delta_{\mathbf{s}}(t)$  will be normalized to satisfy the condition  $\sum_{i} w_{\mathbf{s}}^{i}(t) = 1$ . The normalized weight  $w_{\mathbf{s}}^{i}(t)$  represents the probability that sample  $\delta_{\mathbf{s}}^{i}(t)$  will be selected during the resampling process. Resampling corresponds to selection operator in genetic algorithms. That is, the sample with a higher weight (i.e., utility) has a higher surviving probability.

Smoothing: After the resampling process, a number of the samples could be identical, i.e., they are of the same price. When one sample has a large weight, many samples chosen in the resampling process could be the same and the sample degeneration problem become dominant. To overcome the degeneration problem, we add some random noise to each sample to smooth the sampling space. If there is not the smoothing step, all samples in different time steps use the fixed set of strategies, which are only a small part of the whole strategy space. The discretized strategies result in information loss due to using a small number of strategies and thus the evolutionary dynamics may not converge. In contrast, if there is too much perturbation, over-dispersion problem occurs.

To overcome the degeneration problem and information loss, we use the a smoothing step to adjust the strategies of the samples generated in the resampling process. To overcome the over-dispersion problem in the sequential simulation, we use the Gaussian kernel with a shrinkage rule suggested by Liu and West [2]. Specifically, let  $\overline{\delta}_{\mathbf{s}}(t)$  be the weighted mean of all samples  $\delta_{\mathbf{s}}(t)$  and let  $\widetilde{\delta}_{\mathbf{s}}(t)$  be the weighted variance of all samples  $\delta_{\mathbf{s}}(t)$ . Then each sample  $\delta_{\mathbf{s}}^{i}(t+1)$  is replaced by a new sample generated from the smoothing Gaussian distribution  $G(\delta_{\mathbf{s}}^{i}(t+1))$  with mean  $M_{t}^{i} = \alpha \delta_{\mathbf{s}}^{i}(t+1) + (1-\alpha) \overline{\delta}_{\mathbf{s}}(t)$  and variance  $V_{t}^{i} = (1-\alpha^{2}) \widetilde{\delta}_{\mathbf{s}}(t)$ . We can find that all the samples generated in Step 4 have the same sample mean and weighted sample variance as that of samples at step 2. Thus, over-dispersion is corrected. The smoothing step can be treated as mutation in genetic evolution. The parameter  $\alpha \in [0, 1]$  is a smoothing parameter.

#### 4. PRELIMINARY SIMULATION RESULTS

We implemented a testbed to evaluate our approach in a variety of test environments. For each scenario, we measured average con-



Figure 2: Population size, convergence, and social welfare: example 2

vergence time and the ratio of the social welfare of the allocation given by the evolved evolutionary stable strategies and the optimal social welfare. Figs. 1 and 2 show the simulation results in two different scenarios.

The main findings from simulation are: 1) the proposed approach always converges to evolutionary stable strategies; 2) evolutionary dynamics converges with a small population size; 3) evolutionary stable strategies achieved very good social welfare; and 4) the scale of resource allocation problems does not affect the convergence speed of the evolutionary dynamics.

## 5. CONCLUSION

The contributions of this paper include: 1) We adopt the realistic concept of evolutionary stability and introduce replicator dynamics to compute agents' evolutionary stable strategies. 2) Since agents' strategy space is continuous, we present a sequential monte carlo approach for computing evolutionary stable strategies. Experimental results suggest that the proposed evolutionary dynamics always converges to evolutionary stable strategies. Our approach is general and can be used in many related applications. Our approach can be easily extended to handle incomplete information. For example, consider that there are two sellers s and s'. Assume that the cost of seller s could be either 1)  $c_{s}^{h}$  with probability  $\omega_{s}^{h}$  or 2)  $c_{s}^{l}$  with probability  $1 - \omega_{s}^{h}$ . Then for seller s, we create two populations: one for cost  $c_{s}^{h}$  and the other for cost  $c_{s}^{l}$ . The utility of an individual in the population for  $c^h_{\mathbf{s}}$  (or  $c^l_{\mathbf{s}}$ ) is based on its competition with the population for s'. However, the utility of an individual  $\delta_{s'}^i(t)$  in the population for s' is  $w_{\mathbf{s}'}^i(t) = \omega_{\mathbf{s}}^h w_{\mathbf{s}'}^i(c_{\mathbf{s}}^h, t) + (1 - \omega_{\mathbf{s}}^h) w_{\mathbf{s}'}^i(c_{\mathbf{s}}^l, t)$ where  $w_{\mathbf{s}'}^i(c_{\mathbf{s}}^h,t)$   $(w_{\mathbf{s}'}^i(c_{\mathbf{s}}^l,t))$  is the utility  $w_{\mathbf{s}'}^i(t)$  of  $\delta_{\mathbf{s}'}^i(t)$  when it competes with the population for s with cost  $c_s^h$  ( $c_s^l$ ).

## 6. **REFERENCES**

- R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities," in *Proc.* of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC), 2008, pp. 5–13.
- [2] J. Liu and M. West, Sequential Monte Carlo in Practice. Springer-Verlag, 2001, ch. Combined parameter and state estimation in simulation-based filtering, pp. 197–223.
- [3] P. Taylor and L. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical Biosciences*, vol. 5, pp. 455–484, 1978.