Maximizing Revenue with Dynamic Cloud Pricing: The Infinite Horizon Case

Hong Xu, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto

***Abstract*—We study the infinite horizon dynamic pricing problem for an infrastructure cloud provider in the emerging cloud computing paradigm. The cloud provider, such as Amazon, provides computing capacity in the form of virtual instances and charges customers a time-varying price for the period they use the instances. The provider's problem is then to find an optimal pricing policy, in face of stochastic demand arrivals and departures, so that the average expected revenue is maximized in the long run. We adopt a revenue management framework to tackle the problem. Optimality conditions and structural results are obtained for our stochastic formulation, which yield insights on the optimal pricing strategy. Numerical results verify our analysis and reveal additional properties of optimal pricing policies for the infinite horizon case.**

I. Introduction

Cloud computing has recently attracted much attention from both industry and academia. Beyond technological advances, cloud computing also holds promises to change the economic landscape of computing. Pricing of the cloud resources is both a fundamental component of the cloud economy and a crucial system parameter for the cloud operator, because it directly impacts customer usage pattern and the utilization level of the infrastructure.

Static pricing remains the dominant form of pricing today. However, it is intuitive to adopt a *dynamic* pricing policy in order to strategically influence demand in order to better utilize unused cloud capacity, and to generate more revenue. Dynamic pricing emerges as an attractive strategy to better cope with unpredictable customer demand. A recent such example is Amazon, whose elastic computing cloud platform (EC2) [1] introduced a "spot pricing" feature for its instances, with the spot price dynamically updated to match supply and demand [2] as in Fig. 1.

Given the flexibility to change the price on the spot, a natural question is, what is the *optimal* dynamic pricing policy that a cloud provider — such as Amazon EC2 — can adopt, in terms of maximizing its expected revenue amid fluctuating demand? On one hand, a cloud provider has incentive to *price to the present*, i.e. to set price as high as possible to extract more profit from current customers. On the other hand, doing so increases the risk of negatively affecting future demand especially from low valuation customers. An important observation in cloud computing is that computing resources, such as CPU cycles and bandwidth, are inherently *perishable*: if at some point in time they are not utilized they are of no value. That is, we should also consider *pricing to the future*. It is non-trivial to balance the intrinsic tradeoff with perishable capacity and stochastic demand.

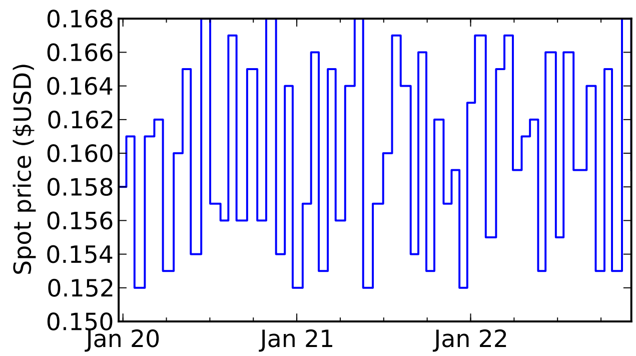


Fig. 1. The history of spot prices on Amazon EC2's large linux instances (API name m1.large) in US-West-1 availability zone, from Jan 20 2011 to Jan 23 2011. Data are obtained via the Amazon EC2 API.

In trying to address this fundamental challenge, we adopt a *revenue management* framework from economics that deals with the problem of selling perishable resources, such as airline seats and hotel reservations, in order to maximize the expected revenue from a population of price sensitive customers [3]. Dynamic pricing has become an active field of the revenue management literature, with successful real-world applications in travel and fashion industries [4], [5].

In our previous work [6], we addressed the problem in a finite horizon setting. In this work we extend the analysis to consider the infinite horizon setting, where the optimal price is only a function of the system utilization and not a function of time. The objective is to maximize the average expected revenue rate in the limit as time goes to infinity.

Our contributions in this paper are two-fold. First, we present an infinite horizon stochastic dynamic program for the revenue maximization problem in the cloud, with stochastic demand arrivals and departures. We characterize optimality conditions for the problem and prove important structural results, such as the monotonicity of optimal price and relative rewards. Second, we conduct numerical studies to verify our analysis. The results also reveal several interesting observations regarding the interplay between the degree of demand dynamics and the optimal pricing policy. Dynamic pricing is more important and rewarding when the expected dynamics is significant compared with the system capacity.

The remainder of the paper is structured as follows. In Sec. II we summarize related work. In Sec. III we present our formulation, and in Sec. IV we present analysis of the stochastic revenue maximization problem in the infinite horizon setting. Sec. V presents numerical results with different demand dynamics. Finally, Sec. VI concludes the paper.

II. Related Work

Research on pricing in data communication networks started arguably from the work of Kelly [7], while there is certainly much prior work on pricing in telecommunication networks [8]. The book by Courcoubetis and Weber [9] provides a useful overview of the field. [10] studies congestion-based pricing in cellular networks, and shows that static pricing achieves good performance in general. This result is extended in [11] and [12]. The focus of these works is on the use of pricing to prevent congestion in large network asymptotics, and the problem formulation assumes a single charge at the time a call is admitted, while our formulation differs by charging on the usage time basis.

Our study is also related to another line of research on Internet bandwidth pricing for ISPs. The main theme here is to understand the tradeoff among resource allocation, performance, and social welfare implied by using flat-rate or usage-based pricing schemes, or a combination of the two called Paris Metro Pricing [13]–[15]. In essence, a static price is used no matter if it is charged on a flat-rate or a usage basis, whereas we consider dynamic prices that vary over time.

There have been some recent studies on pricing of cloud resources. [16] argues for the importance of pricing in the cloud computing context for distributed systems design. [17] proposes a computationally efficient pricing scheme based on mechanism design, and [18] adopts a genetic algorithm to iteratively optimize the pricing policy. The most related work to ours is [19] where a pricing strategy is developed by solving an optimization problem for a cloud cache. These approaches are primarily of a one-shot nature without considering the effect of pricing on future demand and revenue.

III. Problem Formulation

A. Model and Assumptions

We start by introducing our model and assumptions. We focus on an infrastructure cloud provider that sells its computing resources packaged in the form of virtual machines, or *instances*, with a price that is dynamically changing over time. We assume that the spot price p can take any value from the interval $[0, p^{\max}]$. Without loss of generality, we let $p^{\max} = 1$ throughout the paper.

We assume that the cloud operator can influence demand by varying its price p. Demand is determined by two independent Poisson processes, namely the arrival process that corresponds to the birth of new instances, and the departure process that models the death of existing ones (when customers shut down the instances). Here we assume that a demand arrival function $\lambda(p)$ determines the Poisson arrival rate (number of new instances requested per time unit). Intuitively, as price p increases, arrival rate strictly decreases, and thus $\lambda'(p) < 0$. The demand departure process is also a Poisson process, whose rate is modulated by the departure function $\mu(p)$. Clearly, $\mu'(p) > 0$ which means that customers are more likely to leave when spot price is higher.

Price is a function of system state, and it is charged on a usage time basis. This means that, the provider collects p_n from each of the active instances every unit of time, if there are n active instances running in the system. $0 \leq n \leq C$ and C denotes the system capacity. A pricing policy is a rule that determines which price should be advertised at any given time as a function of the current state n. Since we are looking for a stationary pricing policy, it is not a function of time as in our previous work [6].

B. Formulation

The revenue maximization problem can be formulated as follows. The goal of the cloud provider is to maximize the average revenue per unit of time in the infinite horizon. This quantity is denoted by J. We are interested in finding a pricing policy $\boldsymbol{p} = \{p_1, p_2, \ldots, p_C\}$ that achieves this goal, from the set $\mathcal{P} = \{\boldsymbol{p} | 0 \leq p_n \leq 1, \forall n\}$ of all possible pricing policies.

Under the above assumption, the system behavior follows the dynamics of a continuous-time birth-death Markov process, and explicit expression for the average profit J can be provided as follows. First, we denote the steady-state probability of state n given a pricing policy $\boldsymbol{p}$ as $\pi_n(\boldsymbol{p})$. The arrival and departure rates in any state given $\boldsymbol{p}$ can be readily obtained, and the calculation of the steady-state probabilities $\pi_n(\boldsymbol{p})$ is straightforward. Due to the Poisson Arrivals See Time Averages property, the average revenue rate is

$$J(\boldsymbol{p}) = \sum_{n=0}^{C} \pi_n(\boldsymbol{p}) \cdot n p_n, \tag{1}$$

since price is charged on a usage time basis in cloud computing.

The provider's problem is to find a pricing policy $\boldsymbol{p}^{\boldsymbol{*}}$ that maximizes the average revenue rate denoted by J^*. Equivalently,

$$J^* \doteq \sup_{\boldsymbol{p} \in \mathcal{P}} J(\boldsymbol{p}). \tag{2}$$

This is a finite-state, continuous-time, average reward dynamic programming problem. Note that the set $\mathcal{P}$ is compact and all states communicate, so there always exists a policy with which we can eventually reach an arbitrary state n' from any state n. The demand arrival and departure functions $\lambda(p)$ and $\mu(p)$ are all continuous, and thus the transition rates and average reward rate J are continuous in the decision variables p_n. Moreover, the reward rate and the expected holding time at each state n, and the total transition rate out of any state are all bounded functions of $\boldsymbol{p}$. Under these assumptions, the standard DP theory applies [20], and there exists a stationary policy which is optimal.

IV. Optimal Dynamic Policies

A. Optimality Conditions

(2) is a complex stochastic dynamic programming problem. To solve it, we can consider the so-called Bellman's equations since all the states in the Markov chain are recurrent. Bellman's equations are formulated for discrete-time Markov chains. Thus, we need to discretize our Markov chain by a procedure called *uniformization*, where the transition rates out of each state are normalized by the maximum possible transition rate v, which in our case is given by

$$v = \max_{p} \ \lambda(p) + \mu(p). \tag{3}$$

The uniformized Markov chain for our problem is illustrated in Fig. 2.

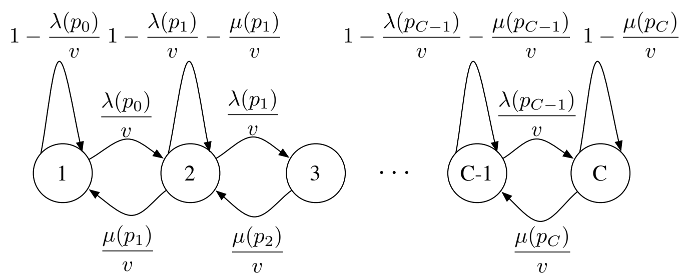


Fig. 2. The uniformized Markov chain for our problem.

With the above setup, we can obtain the Bellman's equations of the following form:

$$J^* + h(n) = \max_{p_n \in \mathcal{P}} \Bigg[np_n + \frac{\lambda(p_n)}{v} h(n+1) + \frac{\mu(p_n)}{v} h(n-1) + \left(1 - \frac{\lambda(p_n)}{v} - \frac{\mu(p_n)}{v}\right) h(n) \Bigg]. \tag{4}$$

for $n = 1, 2, \ldots, C-1$. The first term in the right-hand side is the revenue rate at state n. The second, third, and fourth terms are contributions to the revenue if the next transition is an arrival, a departure, or a recurrence, respectively. The unknowns in the above equations are $h(n)$ and J^*. J^* represents the optimal expected revenue per unit time as discussed, and $h(n)$ denotes the *relative reward* in state n. In particular, consider an optimal pricing policy that attains the maximum in (4) for every state n. If we follow this policy starting from state n' or state n, the expectation of the difference in average rewards over the infinite horizon is equal to $(h(n') - h(n))/v$.

Note that (4) holds only for $n \in [1, C-1]$. To deal with cases when $n = 0$ and C, i.e. when there is no demand departure and arrival, respectively, we can simply let $h(-1) = 0$, and $h(C+1) = h(C)$.

The solution to the Bellman's equations can be computed using classical DP algorithms such as the policy iterations or relative value iteration [20]. The resulted prices that maximize the right-hand side of (4) at each state n constitutes the optimal pricing policy π^*.

B. Structural Results

Although we have found the Bellman's equations and can numerically solve them, obtaining a closed form solution is quite difficult for arbitrary demand arrival and departure functions. However, we are able to characterize several important structural properties of the optimal solution to the dynamic program (4).

Our first result is the monotonicity of the relative rewards. It corresponds to the fact that it is more desirable to have more active instances and thus a higher utilization of the capacity, as they lead to higher revenue rate in the long run even though they imply fewer revenue opportunities in the future.

Theorem 1: Monotonicity of $h(n)$. $h(n) \geq h(n-1)$ for all $n = 1, 2, \ldots, C$.

Proof: The proof can be carried out in similar methods as used in the proof of Theorem 1 in [10]. Due to limited space the details are omitted. ■

We can further show that the relative rewards exhibit diminishing marginal returns with respect to utilization.

Theorem 2: Concavity of $h(n)$. $h(n)$ is concave in n for all $n = 0, 1, \ldots, C$.

Proof: It suffices to show that

$$h(n) - h(n-1) \geq h(n+1) + h(n), \forall n \in [1, C-1]. \tag{5}$$

We prove by constructing a feasible pricing policy that satisfies the above inequality at equality.

Suppose $\boldsymbol{p^*}$ is the optimal pricing policy that achieve the maximum in the right-hand side of (4) for each state. Consider two copies of the system, which we refer to as System A and B, respectively. System A starts from state $n-1$ and follows the optimal policy. System B starts from state $n+1$ and also follows the optimal policy. Now consider a third copy of the system, System C. It starts with state n, and at any point in time, it sets the price as half of the sum of System A's and B's prices. Let $h'(n)$ denote the relative reward obtained from this pricing policy. Now by construction of the policy and the definition of $h(n)$, we have

$$h'(n) = \frac{1}{2}(h(n-1) + h(n+1)).$$

Since $h(n)$ corresponds to the optimal relative reward, $h(n) \geq h'(n)$, and thus the proof. ■

We can now prove our main result, which is the monotonicity of the optimal price at each state.

Theorem 3: Price Monotonicity. $p_n^* \geq p_{n-1}^*$ for $n = 1, 2, \ldots, C$.

Proof: For convenience, let us denote $g_n = h(n+1) - h(n)$. From Theorem 2 we know that $g_n \leq g_{n-1}$. Rearranging the terms in (4) we have

$$J^* = np_n^* + g_n \frac{\lambda(p_n^*)}{v} - g_{n-1} \frac{\mu(p_n^*)}{v}. \tag{6}$$

By definition of the optimal price p_n^* for state n, any other price cannot make the right-hand side of (6) larger, i.e.

$$J^* \geq np_{n-1}^* + g_n \frac{\lambda(p_{n-1}^*)}{v} - g_{n-1} \frac{\mu(p_{n-1}^*)}{v},$$

which gives us

$$p_n^* - p_{n-1}^* \geq$$
$$\frac{1}{nv}\left[g_n\big(\lambda(p_{n-1}^*) - \lambda(p_n^*)\big) + g_{n-1}\big(\mu(p_n^*) - \mu(p_{n-1}^*)\big)\right]$$

Similarly we can write

$$\begin{aligned} J^* &= (n-1)p_{n-1}^* + g_{n-1}\frac{\lambda(p_{n-1}^*)}{v} - g_{n-2}\frac{\mu(p_{n-1}^*)}{v} \\ &\geq (n-1)p_n^* + g_{n-1}\frac{\lambda(p_n^*)}{v} - g_{n-2}\frac{\mu(p_n^*)}{v} \\ \Rightarrow p_n^* - p_{n-1}^* &\leq \end{aligned}$$
$$\frac{1}{(n-1)v}\left[g_{n-1}\big(\lambda(p_{n-1}^*) - \lambda(p_n^*)\big) + g_{n-2}\big(\mu(p_n^*) - \mu(p_{n-1}^*)\big)\right]$$

Thus for the two inequalities to hold, we must have

$$\frac{1}{nv}\left[g_n\big(\lambda(p_{n-1}^*) - \lambda(p_n^*)\big) + g_{n-1}\big(\mu(p_n^*) - \mu(p_{n-1}^*)\big)\right] \leq$$
$$\frac{1}{(n-1)v}\left[g_{n-1}\big(\lambda(p_{n-1}^*) - \lambda(p_n^*)\big) + g_{n-2}\big(\mu(p_n^*) - \mu(p_{n-1}^*)\big)\right],$$

which holds only when $p_n^* \geq p_{n-1}^*$ since $\lambda'(p) < 0, \mu'(p) > 0$, and $0 \leq g_n \leq g_{n-1} \leq g_{n-2}$. ■

Theorem 3 has natural economic interpretations. When the system is heavily loaded, it is in the interest of the provider to set a higher price to obtain a higher revenue from customers, as well as to discourage future demand in order to prevent the system from overloading. When the system is lightly utilized, the provider can afford to adopt a lower price to attract more customers. This key insight is also consistent with results in our previous work [6] and related work [10].

V. Numerical Studies

In this section, we conduct numerical studies to evaluate the properties of the optimal pricing policy. The system capacity C is set to 10000. This corresponds to a moderate-scale data center, such as a single availability zone in Amazon EC2, with several tens of thousands of virtual instances according to anecdotal evidences [21]. We adopt the demand arrival and departure functions of the form $\lambda(p) = a(1-p^2)$ and $\mu(p) = bp^2$, respectively. In this case, the optimal price for each state n has a closed form solution $p^*(n) = vn/2(g_n a + g_{n-1} b)$, where $g_n = h_{n+1} - h_n$ as seen in Sec. IV-B.

A. Weak dynamics scenario

We first consider a relatively weak dynamics scenario, where the maximum expected demand arrivals and departures a and b are orders of magnitude less than the system capacity C. We set one hour to be the unit of time, and assume that the cloud is expected to launch and close several hundreds of instances per hour in the weak dynamics scenario. Thus we set a and b to 100, which is much smaller than the system capacity $C = 10000$. $v = 100$ in this case. The results are shown in Fig. 3. The optimal price increases with n (number of active instances) as seen from Fig. 3(a). The relative reward h_n clearly grows with n as seen from Fig. 3(b). These observations validate our analysis in Sec. IV-B.

Note that the optimal price does not change much when n increases from 2500 to 10000, and is close to 1 for the interval. This is due to the effect of small demand dynamics compared to even a lightly loaded system (when $n = 0.25C$). The expected long-term revenue can be maximized without considering much about the future demand, i.e. setting price close to 1 to obtain a higher revenue rate at present. To facilitate the understanding, Fig. 3(c) shows a sample path of the optimal price, with the corresponding system state $n(t)$ starting from $n = 5000$ and last for 0.25 unit time, i.e. 15 minutes. We can see that in this time period, $p^*(n(t))$ decreases only marginally from 0.982 to around 0.98, when $n(t)$ slowly decreases from 5000 to about 4930.

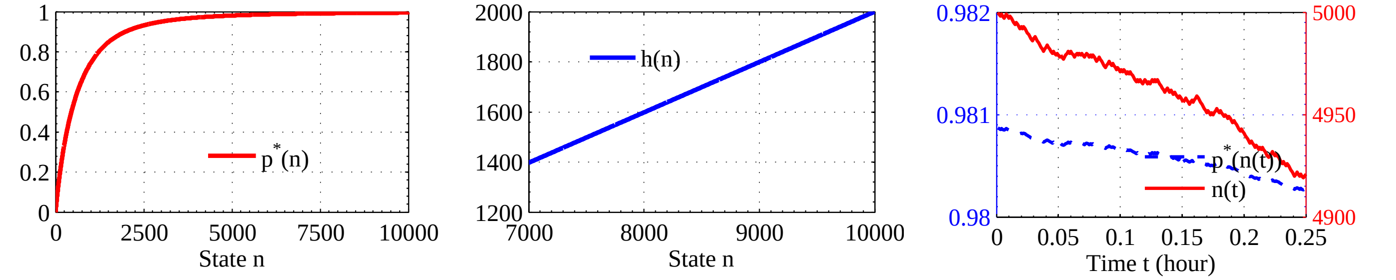
Another observation is that the relative reward is *almost* linear in n as seen in Fig. 3(a). This also can be explained by the small dynamics the system faces. Over time, the system state n is not expected to change much as seen in Fig. 3(c) and price $p^*(n)$ does not vary much with n, implying that the long-term revenue rate is proportional to n. When the system utilization is quite low (below $x = 1000 = 0.1C$), revenue generated from future demand becomes more important, and $p^*(n)$ is much lower and varies with n as in Fig. 3(a).

B. Strong dynamics scenario

Now we examine the optimal pricing policy when the problem embraces a significant degree of demand dynamics. We let a and b equal to 10000, which means the maximum number of new instances (and vanished instances) the cloud can expect is equal to its capacity. $v = 10000$ in this case. Other parameters remain the same as in the previous experiment. The results are shown in Fig. 4.

The optimal price and relative rewards again exhibit monotonicity as expected from our analysis. Compared to the small dynamics case, the first observation is that optimal price varies significantly with state n for the entire range of n, and increases much slower than it does in the small dynamics case in Fig. 3(a). As seen from Fig. 4(a), $p^*(9000)$ is much higher than $p^*(5000)$ which is in turn much higher than $p^*(1000)$. The reason is that when the dynamics is strong, revenue from future demand is crucial in maximizing the long-term revenue rate. The operator cannot set a price close to 1 in the hope of maximizing the present revenue rate, because doing so has detrimental effect on future revenue. The optimal policy thus is to slowly and steadily increase the price as n grows. This also explains the stronger concavity of $h(n)$ as seen in Fig. 4(b), because the marginal benefit of increasing the utilization is diminishing, causing the revenue curve to be bent downwards.

Therefore, dynamic pricing becomes more critical in a strong dynamics setting. We can expect the optimal price fluctuating with the utilization, since the number of instances in the system is expected to fluctuate quickly over time. This is demonstrated in the sample path of the system state and price in Fig. 4(a). Compared with Fig. 3(a), the price grows from 0.3 to around 0.5 as n also grows. It is thus reasonable to conclude that dynamic pricing plays an important role in the strong dynamics setting.

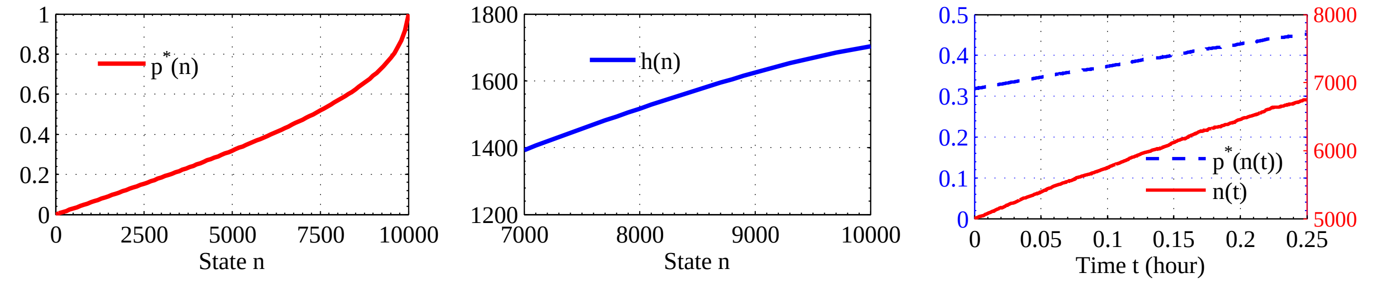


(a) Optimal price.

(b) Relative reward.

(c) A sample path of $n(t)$ and the corresponding optimal price $p^*(n(t))$.

Fig. 3. Numerical results with $C = 10000, \lambda(p) = 100(1 - p^2), \mu(p) = 100p^2$.



(a) Optimal price.

(b) Relative reward.

(c) A sample path of $n(t)$ and the corresponding optimal price $p^*(n(t))$.

Fig. 4. Numerical results with $C = 10000, \lambda(p) = 10000(1 - p^2), \mu(p) = 10000p^2$, and $N = 10^5$ time intervals.

VI. Concluding Remarks

In this paper, we presented an infinite horizon revenue maximization framework to tackle the dynamic pricing problem in an infrastructure cloud. The technical challenge compared to previous pricing work is that prices are charged on a usage time basis, and as a result the demand departure process has to be explicitly modelled. An average reward dynamic program is formulated for the infinite horizon case. Its optimality conditions and structural results on optimal pricing policies were presented. We showed that the relative rewards as well as the optimal price exhibit monotonicity, which is resonant with previous results [6], [10]. We also conducted numerical studies to verify the analysis, and illustrated the importance of dynamic pricing especially in the strong demand dynamics scenarios.

References

[1] "Amazon EC2," http://aws.amazon.com/ec2/.

[2] "Amazon EC2 spot instances," http://aws.amazon.com/ec2/spot-instances.

[3] L. Weatherford and S. Bodily, "A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking, and pricing," *Operations Research*, vol. 40, no. 5, pp. 831–844, September 1992.

[4] M. K. Geraghty and E. Johnson, "Revenue management saves national car rental," *Interfaces*, vol. 27, no. 1, pp. 107–127, 1997.

[5] B. Smith, R. J. Leimkuhler, and J. S. Darrow, "Yield management at American Airlines," *Interfaces*, vol. 22, pp. 8–31, 1992.

[6] H. Xu and B. Li, "Revenue maximization with dynamic cloud pricing," 2011, in submission.

[7] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Operat. Res. Soc.*, vol. 49, no. 3, pp. 237–252, March 1998.

[8] B. M. Mitchell and I. Vogelsang, *Telecommunications Pricing: Theory and Practice*. Combridge University Press, 1991.

[9] C. Courcoubetis and R. Weber, *Pricing Communications Networks: Economics, Technology and Modelling*. John Wiley & Sons, Ltd., 2003.

[10] I. C. Paschalidis and J. N. Tsitsiklis, "Congestion-dependent pricing of network services," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 171–184, April 2000.

[11] I. C. Paschalidis and Y. Liu, "Pricing in multiservice loss networks: Static pricing, asymptotic optimality, and demand substitution effects," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 425–438, June 2002.

[12] X. Lin and N. Shroff, "Simplification of Network Dynamics in Large Systems," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 813–826, August 2005.

[13] A. Odlyzko, "Paris Metro Pricing for the Internet," in *Proc. ACM EC*, 1999.

[14] X. R. Cao, H. X. Shen, R. Milito, and P. Wirth, "Internet pricing with a game theoretical approach: Concepts and examples," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 208–216, April 2002.

[15] C.-K. Chau, Q. Wang, and D.-M. Chiu, "On the viability of Paris Metro Pricing for communication and service networks," in *Proc. IEEE INFOCOM*, 2010.

[16] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, "Distributed systems meet economics: Pricing in the cloud," in *Proc. HotCloud'10: 2nd USENIX Conf. Hot Topics in Cloud Computing*, 2010.

[17] M. Mihailescu and Y. M. Teo, "On economic and computational-efficient resource pricing in large distributed systems," in *Proc. 10th IEEE/ACM Intl. Symp. Cluster, Cloud and Grid Computing*, 2010.

[18] M. Macías and J. Guitart, "A genetic model for pricing in cloud computing markets," in *Proc. 26th Symp. Applied Computing*, 2011.

[19] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki, "Optimal service pricing for a cloud cache," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1345–1358, February 2011.

[20] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Academic Press, 1976.

[21] "Amazon's EC2 Generating 220M+ Annually," blog-post, http://cloudscaling.com/blog/cloud-computing/amazons-ec2-generating-220m-annually.