CS 594 Project Report

Comparative Analysis of Simple and Complex Cloud Pricing Schemes with Resources

Mayank Raj
(mraj3@uic.edu)

Chinmay Nautiyal
(cnauti2@uic.edu)

Abstract

In this report we document our efforts in exploring simple pricing schemes in the context of different resource types. In most cloud pricing scenarios, users present themselves with jobs which have preferences over resource types, and we intend to find what the effects on revenue and welfare are in the case where resources are priced in a “simple” way as opposed to a setting where each job has a different price for each quantity of the resource. The idea is to have strong welfare and revenue guarantees, which reasonably justify foregoing complex pricing mechanisms. We examine the accumulated welfare and revenue for both types of pricing techniques and consider a stochastic model for jobs coming in with different prices. We intend to find if simple pricing schemes work in this context. The results are inconclusive as to whether simple mechanisms are bounded in this context.

1 Introduction

Cloud pricing is an important consideration with the current market for cloud services. It is not exactly clear how to charge for these services, several models for cloud pricing exist. Some of these include, on-demand pay as you go pricing, dynamic pricing models where prices change in accordance with different parameters. In light of this, simple mechanisms offer an easy to understand pricing structure, for the consumer and architect alike. The fundamental concern with these pricing structures is the question of how much revenue and welfare is at stake? Simpler mechanisms tend to lose on revenue, because they fail to take into account the precise demand.

Kash et al [1] tackle this question in the context of job lengths. They examine welfare and revenue bounds in the situation where prices can be set in accordance with job lengths and the case where a simple price per unit time is set.

2 Literature Review

We looked at several papers in order to inform ourselves of the research within this domain and to extend some older ideas to the setting of cloud prices.

2.1 Simple Pricing Scheme for Cloud

The paper looks at the problem for pricing jobs for cloud allocation, and considers a model where you set a single price for multiple jobs coming in for allocation. Jobs coming in can be of different length, and intuitively the allocation mechanism would like to charge higher for longer jobs, in order to maximise revenue. Several such parameters(arrival time, job length deadline, value) can also be factored in for this allocation, allowing for the mechanism designer to maximise revenue and welfare. However, specifying these parameters is often complicated in itself, and requires time and resources. These parameters may not always be known up front, and it is much simpler to set a single price. The idea is that simplicity is often more desirable than optimality, and at the cost of losing some optimality, one might want a more simplistic model. It turns out that it is in fact reasonable to have this approach.

The paper goes on to prove that in practice, you can achieve at least 50% of the welfare and revenue in the case where you set a single price per time step, relative to the case where you can set a different price in accordance with different parameters. This is done in the setting of a single server, and in the case with multiple servers. In the case with multiple servers, the welfare and revenue with the same price for all servers instead of different prices for different servers is analysed. The paper assumes a stochastic model, where the value per time step is chosen from the same probability distribution independent of job length. This assumption is relaxed in an extension with tighter parameters for job lengths.

In this setting the probability distribution of values is known to the mechanism designer, and is not unknown. The model consists of a setting with discrete time steps. Incoming jobs on completion yield a value to the server. If an incoming job has value greater than a posted price p, it is accepted and otherwise it is rejected. A job of length l when accepted, occupies the server for l steps. They initially consider a simpler setting with a single server and two jobs with lengths 1 and 2, and extend it to the general case where the jobs can be of arbitrary length. The optimal price might be different from the single price chosen, but the simple price chosen from one of the discriminatory prices in the multi-price setting in general achieves at least 50% of the welfare and revenue.

2.2 Approximation Algorithms for Item Pricing

We consider here an instance where the seller is trying to price the products to maximize the profit. This scenario considers the situation when customer has valuation over a pairs of items, and makes a purchase only when the combined price is below their value. They model the problem here as a multi graph, where each edge e has some valuation w_e, and the goal is to set prices $p_i >= 0$ on the vertices of the graph to maximize the total profit: that is,

$$\text{Profit}(\mathbf{p}) = \Sigma\ price(e)\ .\quad where\ (e:\ we\ >=\ price(e))$$

where price(e) = $\sum i \in e\ p_i$, and **p** is the vector of individual prices.

This is also called the graph vertex pricing problem. If customers have valuations over large subsets, this can be modeled as one of pricing vertices in hypergraph, or in standard terminology, the problem of pricing items in an unlimited-supply combinatorial auction with single minded bidders.

Here, they give a 4-approximation for the graph vertex pricing problem, and O(k)-approximation for the case of hypergraph in which each edge has at most size k. Also, there is a discussion about pricing some items below their cost, and it is found that such a pricing can produce an $\Omega(logn)$ factor more profit than possible if all items must be priced above cost.

Incentive Compatibility

Incentive compatibility basically deals with the mechanism being truthful. Here the assumption is made that the seller has understanding of the market i.e., the seller knows how many customers would buy different sets of items and at what price. If this assumption is not made then, there has to be an incentive compatible mechanism . But it could be seen that if there are sufficiently large number of bidders, then this type of mechanism becomes incentive compatible.

It is considered here that we have m customers and n items, and its an unlimited supply setting, and they have zero marginal cost to the seller. Each customer is single minded i.e. each customer is interested in only a single bundle of items and has 0 valuation for any other bundle.

Therefore, the valuations can be summarized by a set of pairs (e,w_e), indicating that a customer is interested in a bundle e valuing it at w_e.

The proved result for the graph vertex pricing problem is that there is a 4-approximation.

$$\frac{1}{2}\left[\mathrm{OPT}_1 + \frac{\mathrm{OPT}_2}{2}\right] \geq \frac{\mathrm{OPT}}{4}.$$

Also, for a k-Hypergraph Vertex Pricing Problem, this generalizes to *O*(k) approximation.

$$\mathbf{E}[X_{i,e}] = \mathbf{Pr}[i \in V_L \text{ and } e \in E'] \geq \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-1}.$$

$$\begin{aligned}
\mathbf{E}\left[\sum_{i \in V_L, e \in E'} \mathrm{OPT}_{i,e}\right] &= \mathbf{E}\left[\sum_{i \in V, e \in E} X_{i,e}\mathrm{OPT}_{i,e}\right] \\
&= \sum_{i \in V, e \in E} \mathbf{E}\left[X_{i,e}\right]\mathrm{OPT}_{i,e} \\
&\geq \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-1}\mathrm{OPT} \\
&\geq \frac{\mathrm{OPT}}{ke}.
\end{aligned}$$

We did not find any ideas that we could reasonably implement from these papers in our setting. The idea of trying to model the multi-resource problem as a k-hypergraph seems promising, but we with our limited understanding it remains a challenge.

3 Model

- We consider the case with jobs coming in with demands over resources
- This is captured by a resource vector which specifies how many types and how much quantity of a resource type the job requires
- Prices are drawn randomly from a distribution

In our model, we consider jobs coming in with demands over resource types. There are certain number of present resources, and the job is allowed to express how much of what resource the job wants. We examine the accumulated welfare and revenue in the following 2 situations:

Complex Setting:

In this setting the prices are set in accordance with different quantities of difference resources. The idea is that for different quantities of the resource you set different prices. We draw these prices from a probability distribution, and new prices are drawn for each incoming job.

Simple Setting:

In this setting a single per unit price is set for each resource, and the cost for each quantity of the resource is just the quantity of the resource multiplied by the per unit price.

The posted price in both the settings just becomes the total sum of the price of the resources in the bundle.

3.1 Assumptions:

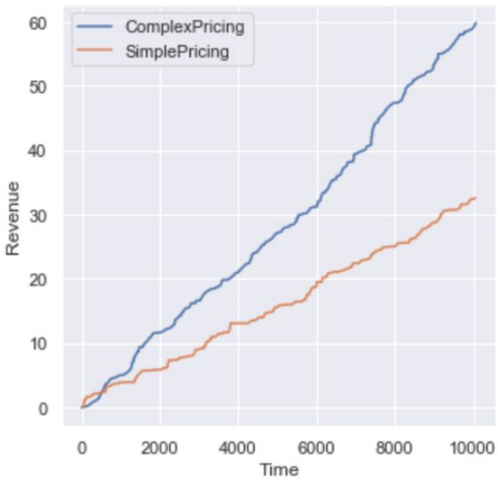
We assume one job shows up at each time step. We assume that each job comes in with a preference of at most 10 resource types. A maximum of 10 resources can be chosen in each run of the job. Each job if accepted, runs for a maximum of 100 units of time. Jobs once accepted can't be preempted, and jobs which are rejected are lost.

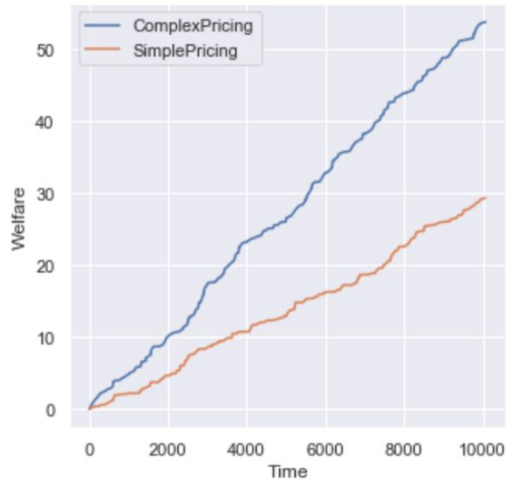
4 Analysis

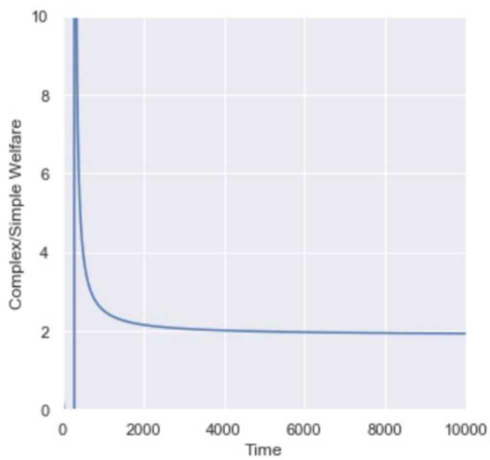
We study the growth of welfare and revenue over time. The graphs are plotted for prices drawn from the following distributions:

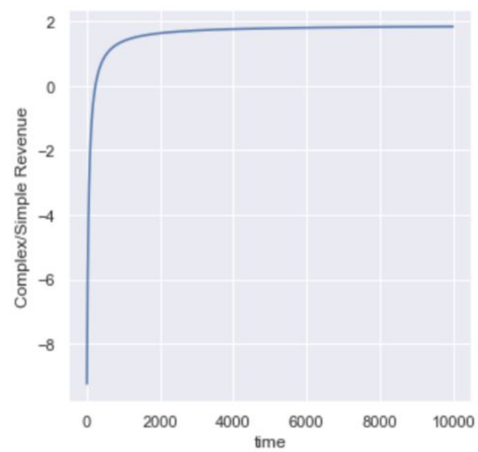
- Uniform distribution over [0,1)
- Beta distributions with parameters
  - 1, 2
  - 0.5, 0.5
  - 2, 2

Uniform distribution over [0,1)

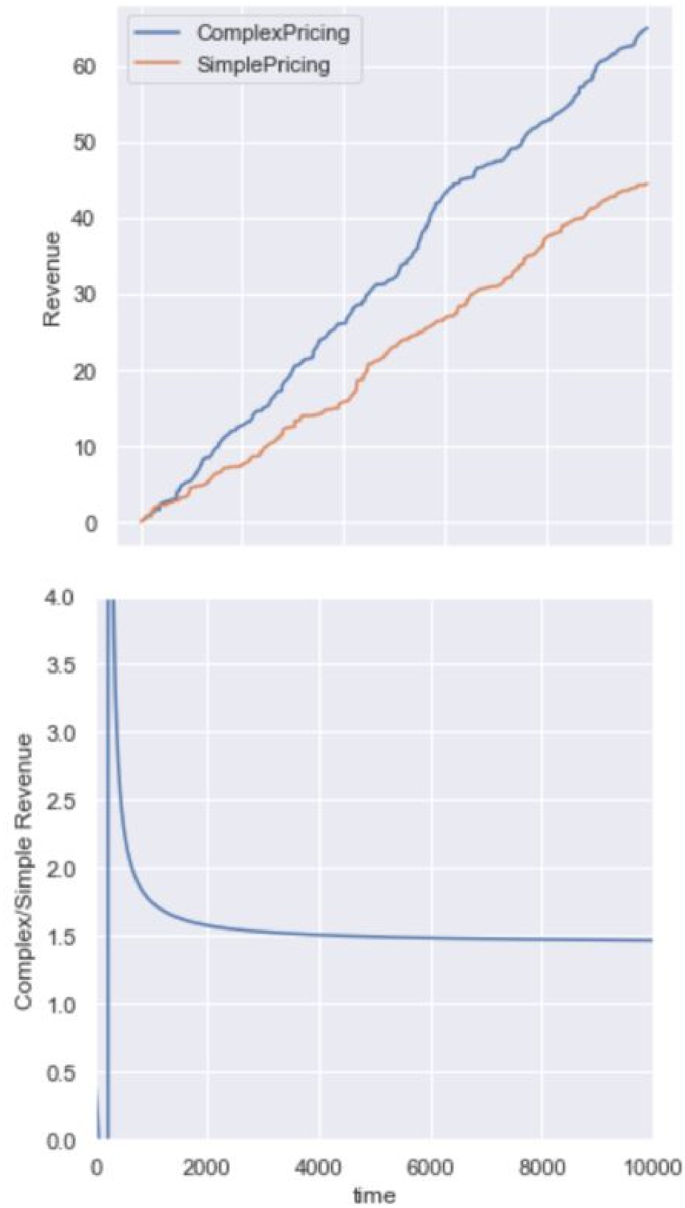


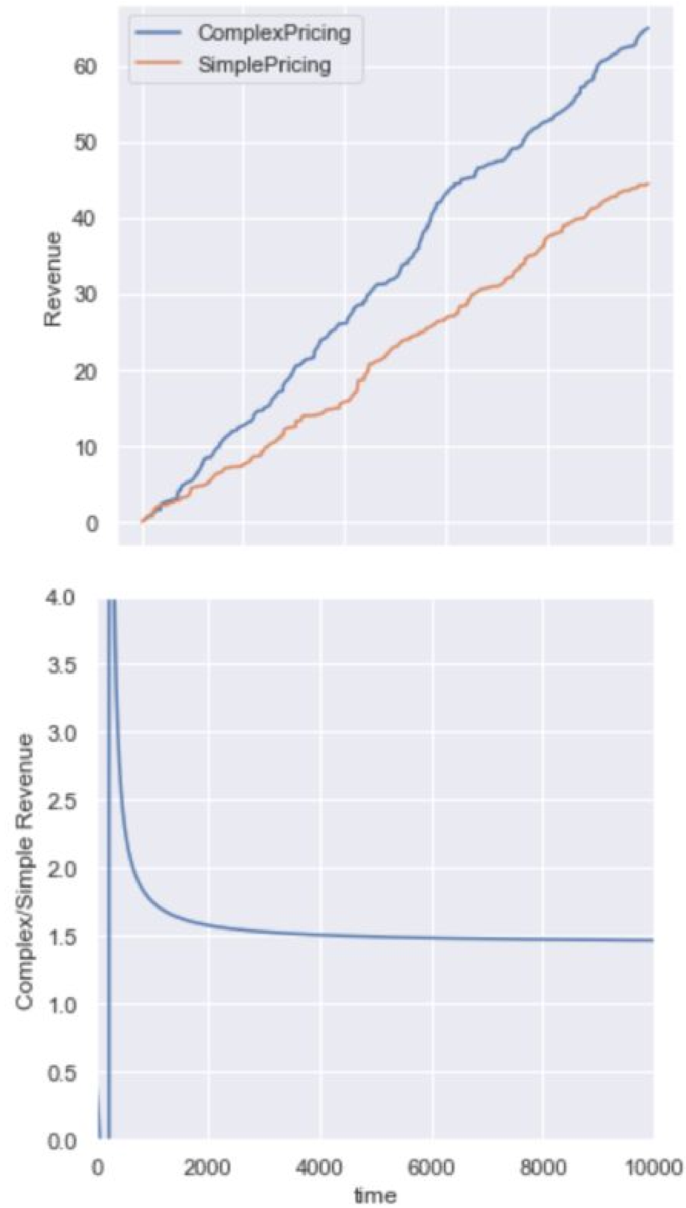




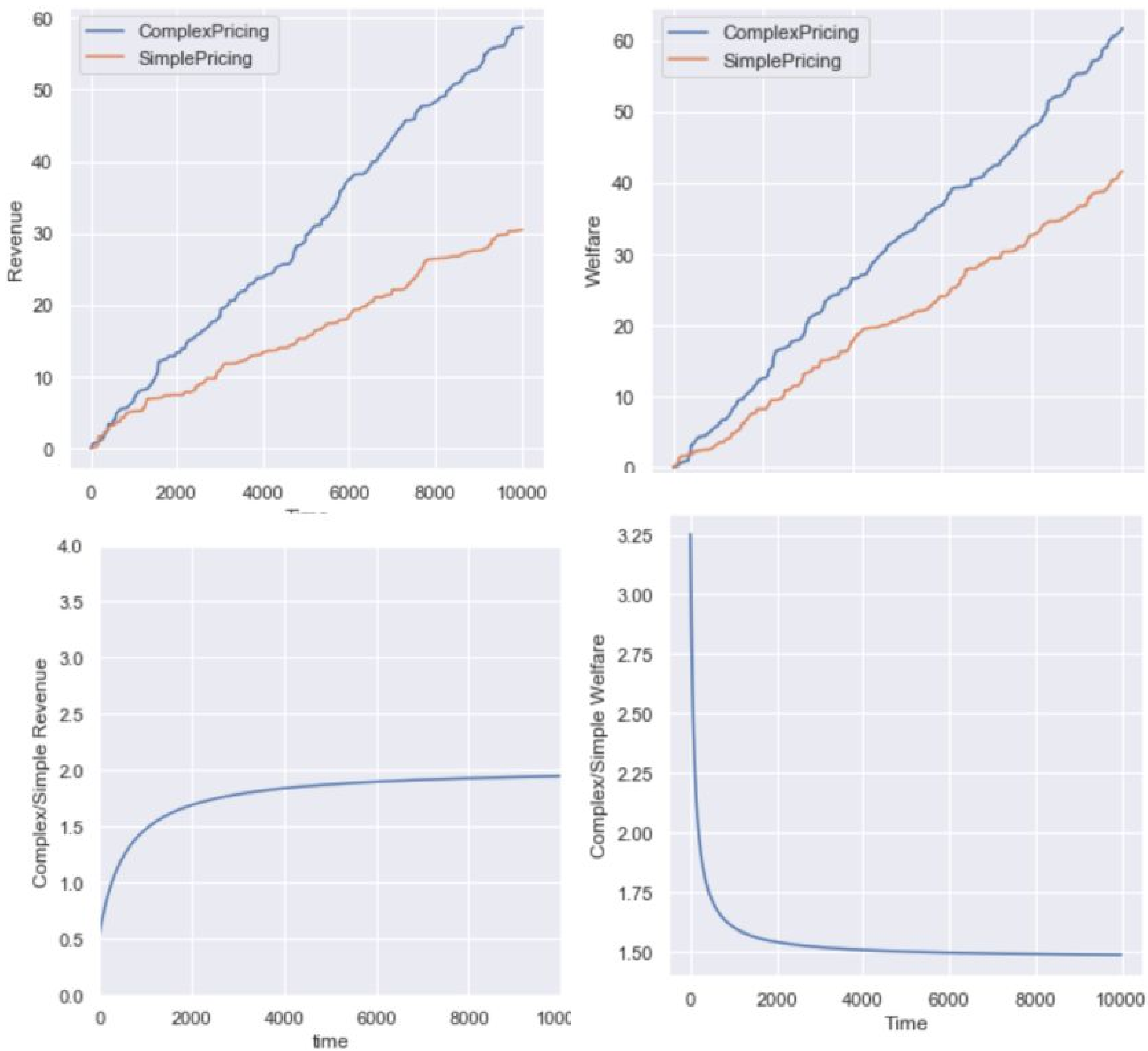


Beta Distribution(1,2)

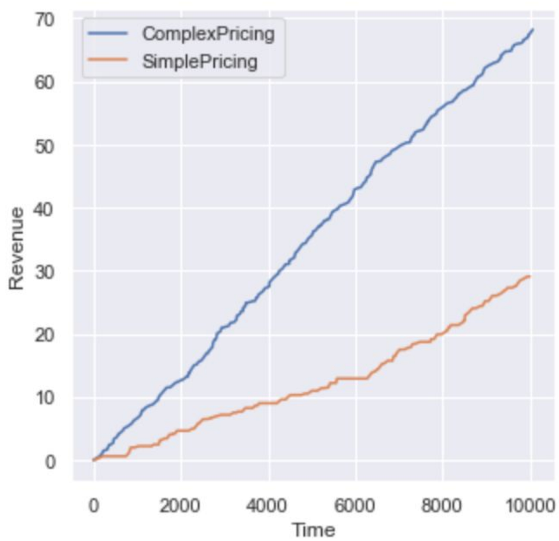


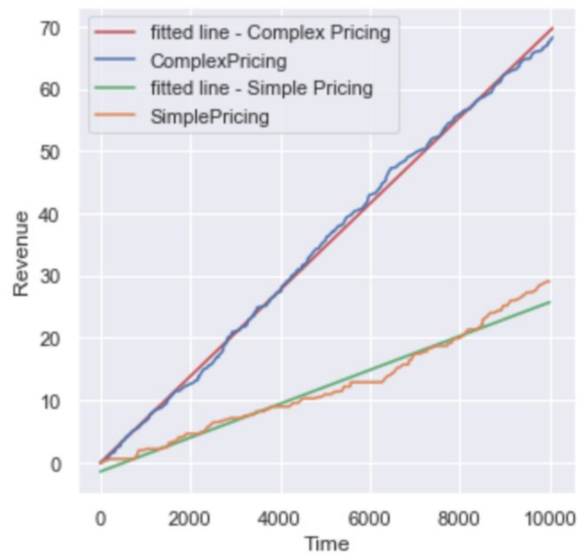


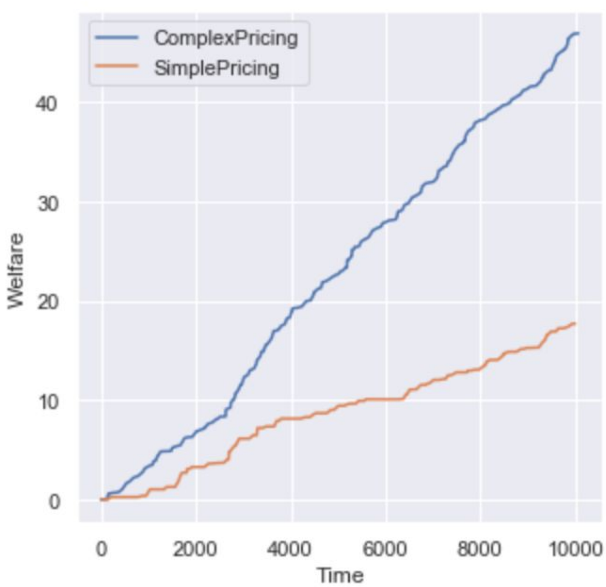
Beta Distribution (0.5, 0.5)

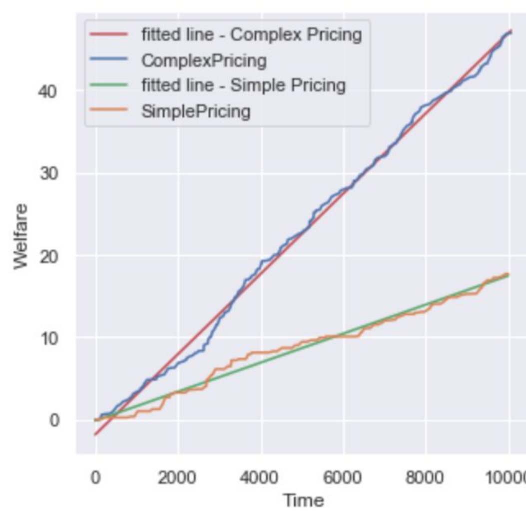


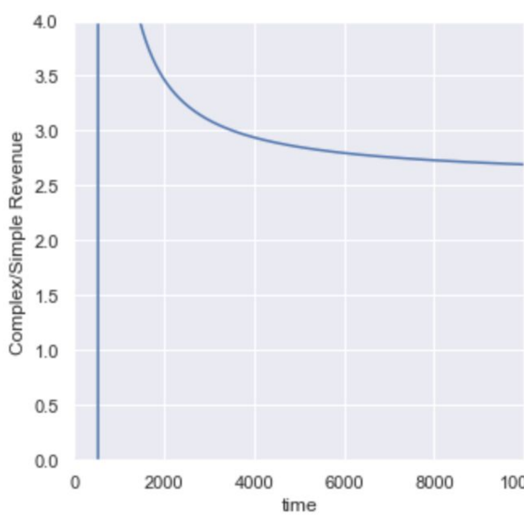
Beta Distribution(2, 2)

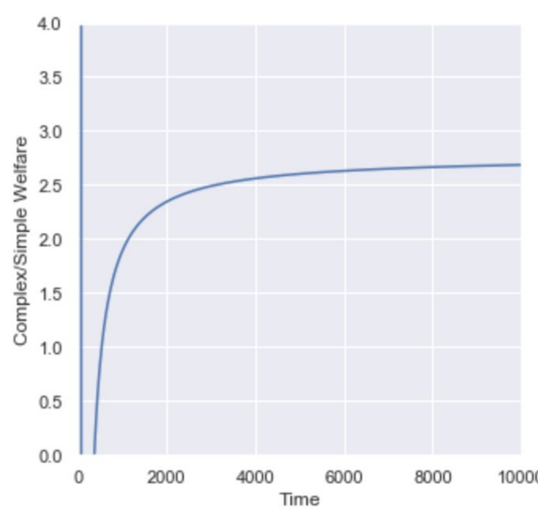












Conclusion

We don't quite understand what the ratios point to. In some situations, complex pricing tends to do way better initially and then converges to a given ratio, while in others simple pricing does way better and until the ratio converges. The convergence values are themselves quite different, which leads us to think something might be missing with the model.

Future work might additionally include looking at models where multiple jobs might come in, but this is a tangential line of work. We additionally want to look at simulations where a single price is fixed for the entire run, and compare welfare/revenue in those situations.

References

[1] Ian A. Kash, Peter Key and Warut Suksompong: Simple Pricing schemes for the cloud

[2] Maria-Florina Balcan and Avrim Blum: Approximation Algorithms and Online Mechanisms for Item Pricing

[3] Avrim Blum and Jason D. Hartline: Near-Optimal Online Auctions