

# Model Selection in an Information Economy: Choosing what to learn

Christopher H. Brooks  
Artificial Intelligence Lab  
University of Michigan  
Ann Arbor, MI 48109-2110  
chbrooks@umich.edu

Rajarshi Das  
Institute For Advanced Commerce  
IBM Research  
PO Box 704  
Yorktown Heights, NY 10598  
rajarshi@us.ibm.com

Jeffrey K. MacKie-Mason  
School of Information and  
Department of Economics  
University of Michigan  
Ann Arbor, MI 48109  
jmm@umich.edu

Robert S. Gazzale  
Department of Economics  
University of Michigan  
Ann Arbor, MI 48019  
rgazzale@umich.edu

Jeffrey O. Kephart  
Institute For Advanced Commerce  
IBM Research  
PO Box 704  
Yorktown Heights, NY 10598  
kephart@us.ibm.com

Edmund H. Durfee  
Artificial Intelligence Lab  
University of Michigan  
Ann Arbor, MI 48109-2110  
durfee@umich.edu

## Abstract

In an economy in which a producer must learn the preferences of a consumer population, it is faced with a classic decision problem: when to explore and when to exploit. If the producer has a limited number of chances to experiment, it must explicitly consider the cost of learning (in terms of foregone profit) against the value of the information acquired. Information goods add an additional dimension to this problem; due to their flexibility, they can be bundled and priced according to a number of different price schedules. An optimizing producer should consider the profit each price schedule can extract, as well as the difficulty of learning of this schedule.

In this paper, we demonstrate the tradeoff between complexity and profitability for a number of common price schedules. We begin with a one-shot decision as to which schedule to learn. Schedules with moderate complexity are preferred in the short and medium term, as they are learned quickly, yet extract a significant fraction of the available profit. We then turn to the repeated version of this one-shot decision and show that moderate complexity schedules, in particular two-part tariff, perform well when the producer must adapt to nonstationarity in the consumer population. When a producer can dynamically change schedules as it learns, it can use an explicit decision-theoretic formulation to greedily select the schedule which appears to yield the greatest profit in the next period.

Keywords: Online learning, information economics, model selection

# 1 Introduction

Electronic information goods are very flexible. In contrast to physical goods, marginal costs are negligible and nearly limitless bundling and unbundling of these items is possible. Consequently, producers can offer complex pricing and bundling schemes that would be infeasible for traditional commerce in physical goods. When articles have a very low marginal cost, it is almost always advantageous to include them in a bundle if any consumer values them positively. Considering only pricing structures that are based on the number of items in a bundle, and not on the identity of the items or the identity of the consumer, there are families of such pricing schedules with one free parameter, two parameters, and so forth. In the limit, the most general pricing function for this problem has  $N$  parameters, where  $N$  is the total number of different information goods under consideration. These different families of schedules approximate consumer demand with different degrees of precision, thereby yielding different profits.

It would seem reasonable to assume that a producer with more free (and independent) pricing parameters to control will want to use all of those parameters, since it will be able to fit the consumer demand curve more precisely. However, when a producer faces uncertainty about the nature of consumer demand, the situation is not so straightforward. Optimal pricing under more complex schemes requires more knowledge about consumer preferences than pricing under simple schemes requires. In addition, different schedules with the same number of parameters may have different learning curves. Learning about consumer preferences takes time; meanwhile, the firm is earning less than the optimal profit. In a world in which a producer has a limited number of chances to interact with a consumer population, the cost of learning becomes very important; learning a complex schedule is not useful if the producer does not have sufficient opportunity to recoup learning costs.

Aspects of our problem have been studied by prior authors. There is an extensive economic literature on how a firm can use multi-parameter pricing schedules to extract greater surplus when the distribution of consumers is known, but the firm cannot tailor individual-dependent prices; see Wilson [22] for a thorough overview. Maskin and Riley [14] present a method for deriving the most profitable unconstrained nonlinear pricing scheme when consumers are differentiated by a single taste parameter for which the distribution is known. Economists and others have studied how agents optimally choose between competing opportunities of unknown reward, often referred to as multi-armed bandit problems [21]. In bandit problems, an agent weighs the benefit of gaining information by experimenting against the cost of experimentation (manifested as foregone short-run profits). Rothschild has studied how a firm chooses a one-parameter linear price when it faces uncertain consumer preferences [17]. Braden and Oren [2] have found that when the distribution of consumer demands is unknown, the desire to learn leads to lower prices in early periods.

The machine learning community has also considered the problem of efficient learning. The problem of how to determine an optimal sequence of actions, when there is a tradeoff between exploration and exploitation, is a primary focus of the reinforcement learning literature. Sutton and Barto [18] provide an excellent overview.

Problems in which a learner acquires data incrementally and must interleave learning with acting in the environment (as opposed to processing a large dataset offline and then selecting an action) are referred to as on-line learning problems. Work in both machine learning [1] and optimal control [7] has examined this problem, typically asking questions such as how to learn a model with the smallest number of queries. Thus, they do not consider an economically-sensible objective function, such as our objective of maximizing aggregate payoff net of any costs of learning. When a learner is able to select data points to examine she has a decision problem in which there is a trade-off between the opportunity cost of further exploration and the value of the information acquired. This is known as a problem with active data selection. Thrun and Møller [19] survey common techniques, most of which assume that the problem is stationary.

In our research, a producer of information goods selects parameter values that define a price schedule offered to an unknown consumer population. This agent can select from schedules of differing complexity to represent the world. As it gathers evidence, the agent updates its beliefs about the optimal parameter values. We point out that a price schedule can be thought of as a model of consumer demand. If a consumer's valuation for each good in a bundle were known, it could be expressed as a nonlinear price schedule charging that consumer's valuation for each good. Other price schedules are less precise, but still approximate consumer demand.

Various assumptions distinguish our on-line learning problem with active data selection from prior research. First, we assume that the producer's goal is economically sensible: to maximize aggregate profits. The producer is concerned not only with the profits it can expect once it finds a good schedule, but also with the lower profits earned while uncovering this good schedule. Second, we allow the environment to change over time. Thus the producer must also consider the degree to which profits are affected by changes in its environment and how quickly it can "relearn" a schedule. Finally, we endow our firm with no knowledge of its environment. These features render optimal control

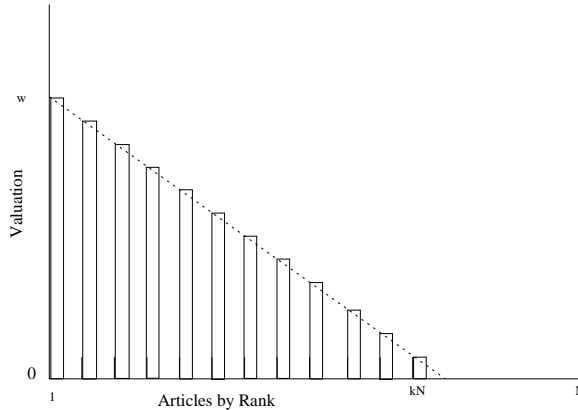


Figure 1: Consumer Valuation under the Chuang and Sirbu model.

methods intractable, and point to simulation as the appropriate tool for studying this problem.

We begin with a fixed-schedule problem in which a producer chooses a single schedule, then learns and updates beliefs about the optimal value of the parameters for that schedule. Simpler schedules outperform those that are more complex over the short and medium run, due to the higher costs of learning more parameters. The two-part tariff performs especially well. We then allow consumer demand to change over time and evaluate schedule performance over a range of change frequencies. Finally, we demonstrate how a producer can do better by dynamically changing between schedules as it learns so as to combine the learnability of simpler schedules with the higher profits of more complex schedules.

## 2 Modeling

In this section we present our model of producer and consumer behavior, describe the different price schedules, and summarize the profits they extract under complete information assumptions.

### 2.1 Consumer Model

In each period, a monopolist offers consumers a new set of  $N$  articles priced according to a price schedule. Consumers are rational utility maximizers. That is, they know their preferences and, on every iteration, will act so as to maximize their expected utility. We assume consumer preferences follow a simple two-parameter model originally introduced by Chuang and Sirbu [5], and described in our previous work [4, 12]. This model has the advantage of being analytically tractable while still providing significant nonlinearities in consumer demand when consumers are heterogeneous. The model consists of two parameters:  $w$ , which indicates a consumer's value for its most-preferred article, and  $k$ , which indicates the fraction of the  $N$  articles available for which the consumer has a positive valuation. The valuation  $V_j(i)$  of the  $i$ th most-preferred article by consumer  $j$  is a linear function of these variables, expressed by:

$$(1) \quad V(j) = \begin{cases} w_j(1 - \frac{i-1}{k_j N}) & \text{if } i - 1 \leq k_j N \\ 0 & \text{if } i - 1 > k_j N \end{cases}$$

This equation produces a downward-sloping line representing consumer valuations, as shown in Figure 1. In practice, consumers can only purchase integer quantities of articles, as indicated by the solid rectangles. It is worth noting that consumers can choose which articles that they wish to consume; the producer only prices articles according to the order in which they are consumed. For example, the first article consumed might cost \$5, regardless of which article the consumer chooses to read.

As we assume that the the producer can produce an additional copy of any article without additional cost, the socially efficient outcome would be for each person to consume  $k_i N$  items. This would yield a surplus of  $\frac{w k_i N}{2}$  to

each person for a total value of  $\frac{w\bar{k}N}{4}$  for the entire population. A firm that could observe each consumer’s  $k_i$  could perfectly price discriminate by making a take-it-or-leave-it offer tailored to each individual and extract this entire surplus. This serves as an upper bound for the maximum profit that a monopolist could earn. (Of course, since a producer cannot actually price discriminate between consumers, it is unlikely to reach this bound when interacting with a heterogeneous consumer population.)

The consumer population can change over time. There are several ways in which this could happen: for example, consumer preferences could change, the population could evolve as consumers leave and are replaced by new consumers, or the span of market facing the producer could change due to the introduction of other producers. In section 4 we consider the case in which the extant consumer population periodically is removed and replaced by new consumers drawn from a static distribution.<sup>1</sup>

## 2.2 Producer Model

We focus on the producer and how its decision as to what it should learn leads to differences in performance. The producer’s goal is to maximize its profit aggregated over time.<sup>2</sup> This depends on the profit from the current schedule, the cost of learning better price parameters, and the value of learning as realized through improved future profitability.

We focus on a monopolist producer interacting with a consumer population. There are two possible explanations for the monopolist assumption: the first is that many information goods, such as journal articles and audio files, are protected by copyright, thereby preventing other producers from selling them. Secondly, the economy we are considering may be part of a larger system in which the producers have subdivided the consumer market into separate niches. Within each niche, a single producer acts as a local monopolist. We have studied conditions sufficient for this to occur elsewhere [3].

We assume that the producer can offer exactly one price schedule in a given period. The producer offers a pricing schedule, receives feedback in the form of profit earned, and uses this to select new prices. In section 5, we allow the producer to select a new schedule. The producer does not have knowledge of or tight beliefs about the distribution of consumer preferences, and thus tries to learn only the direct relationship between price schedule parameters and profit.

We are studying the question of *what* price parameters (i.e. which schedule) to learn, rather than how to optimally learn a given schedule. Therefore, we selected an off-the-shelf optimization algorithm to use in learning prices.<sup>3</sup> We use amoeba [16], a variant of the Nelder-Mead direct search method for nonlinear optimization [15, 20] as a learning algorithm. Amoeba is a hillclimbing algorithm particularly well-suited to the sorts of surfaces we are studying. It works by initially choosing and evaluating  $n + 1$  points (where  $n$  is the number of parameters in the price schedule). The worst point is reflected through the centroid of the convex hull formed by the other  $n$  points. If this produces an improvement, the process is repeated. Otherwise, the worst point is contracted toward the centroid of the hull and the new worst point is reflected. This produces a back-and-forth flipping up the gradient of a landscape which gives amoeba its name. Amoeba performs well on discontinuous, multimodal, terraced surfaces such as the ones induced by these price schedules. This is due both to the fact that amoeba does not explicitly use gradient information to select new points to explore and also the fact that amoeba uses a widely separated set of points to choose new points to explore, as opposed to just examining the profit landscape in its immediate neighborhood. This allows amoeba to skip over ridges and local optima.

## 2.3 Price Schedules

The space of schedules that we explore (presented in Table 1) is larger than that typically considered in the sale of physical goods. The reason for this is that digital goods typically have a high fixed cost, but negligible marginal cost. Once a single digital copy of a good, such as a newspaper article, is produced, copies can be made virtually for free. This makes *bundling* of articles into collections an attractive strategy for dealing with consumer heterogeneity [5, 6]. If an article is valued by any consumer, including it in a bundle increases the bundle’s aggregate value.

The calculation of optimal profit with perfect information for each schedule is described in more detail in our previous work [4]; we simply summarize those results here to provide a benchmark for the best possible profit that

<sup>1</sup>We also allowed the parameters of the underlying distribution to follow a random walk and obtained qualitatively similar results.

<sup>2</sup>In some cases, a producer may want to discount the value of future profit. Although we do not treat it explicitly, it is straightforward to incorporate a discounting factor into our results.

<sup>3</sup>It is possible that the question of what to learn might be sensitive to the choice of learning algorithm. For example, different price schedules might have different algorithms which perform optimally on them, and the improvement from moving to a better algorithm might be sufficient for some methods as to change the ranking of payoffs from the various schedules. This remains an open question for our problem.

Pricing Schedule	Parameters	Optimal Profit	Description
Pure Bundling	$b$	.875	Consumers pay a fixed price $b$ for access to all $N$ articles.
Linear Pricing	$p$	.875	Consumers pay a fixed price $p$ for each article purchased.
Two Part Tariff	$f, p$	1.037	Consumers pay a subscription fee $f$ , along with a fixed price $p$ for each article
Mixed Bundling	$b, p$	1.037	Consumers have a choice between a per-article price $p$ and a bundle price $b$
Block Pricing	$p_1, p_2, j$	1.094	Consumers pay a price $p_1$ for the first $j$ articles ( $j < N$ ), and a price $p_2$ for remaining articles.
Nonlinear Pricing	$p_1, \dots, p_N$	1.152	Consumers pay a different price $p_i$ for each article $i$ .

Table 1: This table presents characteristics and optimal profits for a series of price schedules, when applied to the following population:  $w = 10$ ,  $N = 10$ , and  $k$  drawn from  $U[0, 0.7]$ . Column 3 denotes the optimal profit per good sold that can be earned by each schedule.

could be obtained from a schedule once all information is learned. As we can see from table 1, as the number of pricing parameters increases, the potential profit a schedule can extract is increased. We refer to this as a schedule’s *precision*; what fraction of available profit is it capable of capturing? Another factor, which is not reflected in the above table, is a schedule’s *complexity*, which indicates the cost needed to learn the optimal parameters of a price schedule. Complexity allows us to compare the learnability of different schedules by estimating both how well they will perform once learned, and the cost of learning them. This will be further explored in section 5, when we examine the producer’s decision process as it repeatedly chooses price schedules.

### 3 Learning parameters for a one-time schedule choice

In the simplest formulation of our problem, a producer selects a single price schedule and is then locked into this decision. The producer is free to dynamically change its price parameters every period, but the selection of a schedule is fixed.

In order to compare the performance of each schedule as prices are learned, we conducted a set of experiments in which amoeba was run on the landscape induced by a randomly generated consumer population and a price schedule. In these experiments, we generated a population of 1000 consumers.  $k_i$  was drawn from  $U[0, 0.7]$ .  $w_0$  was fixed at 10, so as to remain consistent with the analytical results summarized in table 1. This produces a consumer population in which each consumer has the same value for its most-preferred article, but consumers vary in the number of articles they value. Experiments were conducted for both  $N = 10$  and  $N = 100$  goods. The producer interacted with a fixed population for 10000 iterations. Results are averaged over 100 runs.

Figure 3 shows our results on the six price schedules for  $N = 10$  and  $N = 100$ . (Note that the X axis is log scale.) Each line indicates the average cumulative (per-article, per-customer) profit for a particular schedule. For example, the value of about 1 for mixed bundling at iteration 10 in the left-hand graph in figure 3 means that over the first 10 iterations, profit averaged 1 (so cumulative profit is approximately 10). The total expected profit earned up to time  $t$  is the height of the curve at  $t$  multiplied by  $t$ . Higher lines therefore indicate schedules which have been more profitable to date.

For  $N = 10$ , after an initial adjustment period of about 10 iterations, the two-parameter schemes perform quite well, dominating for the next 100 periods. After this point, the cost of learning the nonlinear schedule has been absorbed, and nonlinear pricing becomes the most effective strategy. Also worth noting is the poor performance of block pricing; even though it is able to extract more profit under complete information than either two-parameter schedule, it performs poorly here. The reasons for this are discussed below.

The experimental results when  $N = 100$  are shown on the right-hand graph in figure 3. Again, two-part tariff performs well in the short and medium run before being overtaken by nonlinear pricing. The single-parameter schedules (pure bundling and linear pricing) are simply not able to keep up due to their low precision.

While the theoretical properties of each price schedule tell us the maximal profit they can achieve, the rate at which that profit is reached is dependent upon the structure of the profit landscape. The experiments in figure 3 help

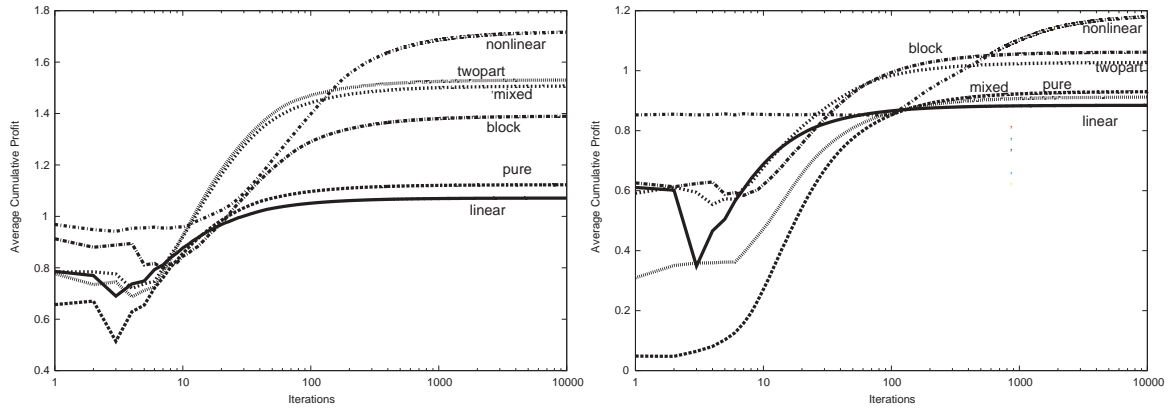


Figure 2: Average cumulative profit per article, per customer for  $N = 10$  (left) and  $N = 100$  (right).

to illustrate some of the subtleties of this problem. First, note that the average cumulative profit per article is lower when  $N = 100$  than when  $N = 10$ . Recall that a consumer is described by the *fraction* of articles she desires. Since consumers can only purchase integer quantities of articles, as the number of articles increases, each consumer is able to target more precisely what she wants, reducing the discontinuity in the landscape. Increasing the number of articles also emphasizes the heterogeneity in the consumer population, making consumers “more different” from each other and thereby reducing the profit (per consumer) that can be extracted using a single schedule.

Nonlinear pricing initially shows a flat line as 101 random samples are taken to fill in amoeba’s simplex when  $N = 100$ . Learning does not actually begin with this schedule until iteration 102.

Also, note the difference in performance for both mixed bundling and block pricing as the number of articles is increased. In the case of mixed bundling, profits fell off, whereas block pricing performed better as the number of articles increased. To understand why the performance of mixed bundling falls off, consider the topographical depiction of the mixed bundling profit landscape shown in figure 3. Since mixed bundling offers consumers a choice between pure bundling and per-article pricing, one solution is to only find the bundle price, setting the per-article price so high that no one will choose it. This places the producer on the large light-gray plateau. In this case, changing the per-article price a little has no change in profit (since no one was buying per-article) and so the producer will remain on that plateau. The larger this plateau is, the more likely a learner will find itself there. Since the size of the bundle price is dependent upon the number of articles offered, as  $N$  increases, a greater fraction of the profit landscape is taken up by this plateau.

In contrast, block pricing performs better as the number of articles offered is increased. Recall that the third parameter of a block pricing schedule ( $j$ ) is the number of articles a consumer must purchase in order to receive a discount. However, when  $N = 10$ , consumers typically value only a few articles, and so most values of  $j$  do not produce any additional profit beyond that of linear pricing. When  $N = 100$ , consumers will value a larger number of articles, and so block pricing is able to effectively find two prices that capture both the high value placed on the first few articles purchased and the low value for the remaining articles. This provides a lesson for designers of price schedules: adding additional schedule parameters requires a producer to consider not only the ability to extract additional profit at a schedule’s optima, but also the way in which the addition of extra parameters will affect the profit landscape.

Two-part tariff becomes a particularly useful schedule given these observations, since it is able to perform well for both  $N = 10$  and  $N = 100$ . Unlike mixed bundling, it contains an easy-to-climb hill, and many non-optimal values have relatively high profit, making exploration less costly. Additionally, there are no large plateaus, making it easy to move between optima. This is discussed further in section 4. Figure 3 shows the landscapes for mixed bundling and two-part tariff.

As we can see, even though they have the same number of parameters and the same optimal profit, the search spaces are quite different. As we discussed previously, the mixed bundling landscape contains the pure bundling and linear pricing landscapes as subspaces, with the pure bundling solution creating a large plateau. Because of this, even though these two-parameter schedules have the same potential optimal profit, the two-part tariff performs better on average after learning is complete, particularly for low  $N$ . Since two-part tariff’s parameters are tightly coupled (as the fee increases a bit, the per-article price falls) the resulting landscape has a nice gradient that a learner can ascend.

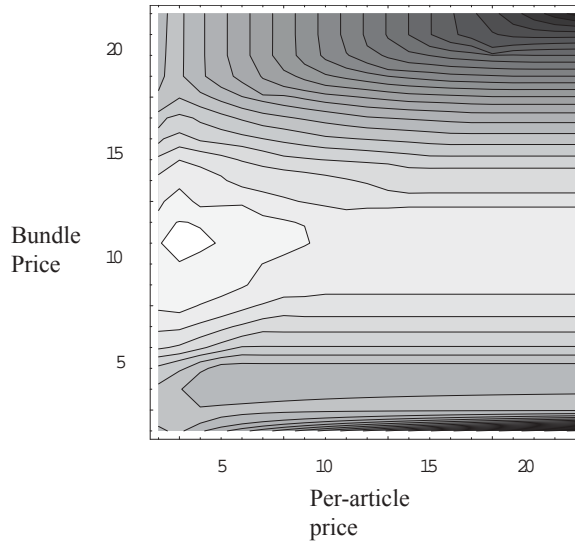


Figure 3: A topographic depiction of the mixed bundling landscape for  $N = 10$ . Lighter regions indicate higher profit.

Recall that the graphs in figure 3 show results averaged over several runs with randomized starting values. The global peak of the mixed bundling landscape was found less often than that of two-part tariff.

These results highlight the exploration-exploitation tradeoff involved in selecting schedules of different complexity. While more highly parameterized schedules can potentially fit consumer demand more precisely, simpler, less precise schedules can be more effective in the short run due to their learnability. Additionally, some schedules may be less robust to a change in model parameters, such as number of articles. Therefore, it is important to consider not only the theoretical solution, but also solutions actually realized in practice.

The analysis in this section provides a recipe for a producer choosing a single schedule: find the point on the  $x$  axis corresponding to the number of iterations it will interact with a consumer population and select the schedule with the highest area under its curve up to that point. (If the ending time is not known with certainty, use an expected utility calculation for each possible ending time.)

While this is an interesting initial step, it assumes that the producer starts with no information about the consumer population. In the following sections, we will relax our assumptions about nonstationarity and allow our producer to track a changing consumer population. We will then allow the producer to dynamically change schedules, using information gained from simpler schedules to “bootstrap” into more complex schedules.

## 4 Examining Nonstationarity Further

One interpretation of the assumptions of the previous section is that it represents a nonstationary environment where the changes are so dramatic that all previous learning is without value to the producer. This is just one end of a spectrum of potential change. In this section, we examine less dramatic nonstationary environments, with an eye toward the question of selecting a schedule that is robust to periodic changes in the consumer population.

In our model, the consumer population is subjected to periodic shocks. When a shock occurs, the existing consumer population is replaced with new consumers drawn from a fixed distribution.

We compared the performance of each schedule to that of a producer who knows the parameters of the consumer distribution and selects a fixed schedule and prices targeted to this distribution. In other words, this is the best that a producer could hope to do (in expectation) without adapting to population shocks. For this experiment, we allowed shocks to occur every 10, 100, 1000 and 10000 iterations. The producer was allowed to learn on a fixed population until it converged on a solution, and then shocks were begun. The experiment was ran for many (typically millions) periods, and results were averaged across shocks. The results from these simulations are presented in table 2.

As we can see from table 2, when the consumer population is changing rapidly, a producer can do better by simply staying in place. The profits from not changing a “good” schedule even as the consumer population changes, depicted

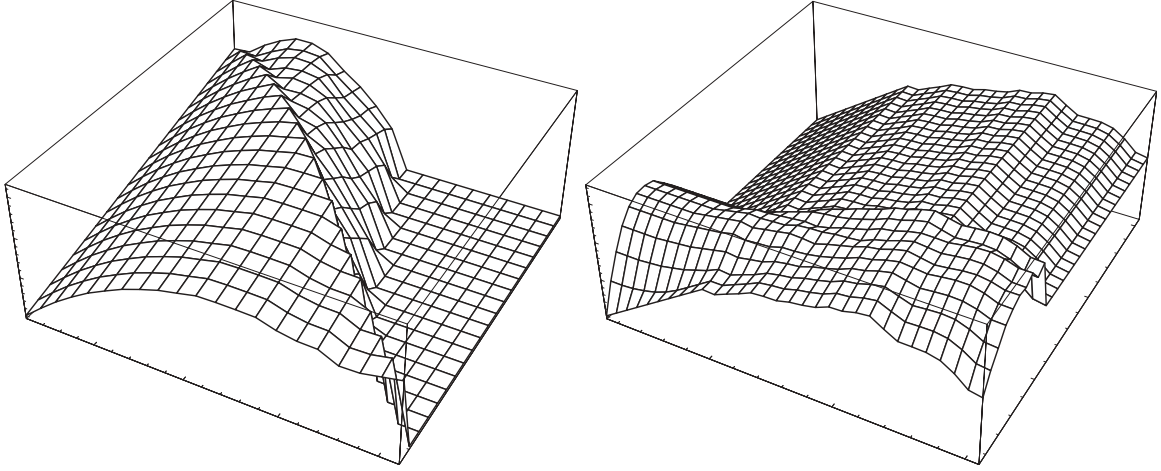


Figure 4: Two-part tariff (left) and mixed bundling (right) landscapes. Even though they have the same number of parameters and the same optimal profit, their landscapes are very different.

Schedule	$T = 10$	$T = 100$	$T = 1000$	$T = 10000$	Optimal	Opt. (fixed)
Pure Bundle	0.847	0.955	1.066	1.079	1.122	0.901
Linear	0.900	0.904	0.905	0.907	0.910	0.900
Two-Part	0.853	1.034	1.144	1.186	1.232	1.071
Mixed	1.016	1.090	1.146	1.154	1.240	1.071
Block	0.855	0.994	1.076	1.253	1.279	1.063
Nonlinear	—	—	1.112	1.312	1.393	1.205

Table 2: Time-averaged profits for each schedule for several values of shock interval  $T$ , using 10 consumers. The second-to-last column gives the optimal profits for  $N = 100$ ; this represents the upper limit on what can be obtained by an adaptive, fully-informed producer. The last column gives the optimal profits for  $N = 100$  for a producer using a fixed price schedule, rather than adapting.

in the final column, are greater than the profits of trying to adapt to frequent changes, depicted in the second column. The producer simply doesn't have enough time to make up profits lost due to learning. (This advantage is dependent upon new consumers being drawn from a stationary distribution; if the distribution of potential consumers changes over time, then adaptation is more likely a successful strategy.) As the shock rate decreases, learning becomes more useful, with the two-parameter schedules again performing well. One result worth noting is the performance of nonlinear pricing. In this experiment, nonlinear pricing did quite well, whereas in the experiments in section 3, it initially performed poorly. This is a direct result of the way in which the experiments were performed. In the experiments depicted in figure 3, the producer was started cold on a stationary population. In the experiment summarized in table 2, the producer was first allowed to learn on a stable population to eliminate the initial transient, *then* the shocks began. This illustrates the huge difference that initial knowledge about the consumer population, or, alternatively, the ability to learn about the population before the experiment begins, can play in selecting a price schedule.

A persistent question in our research is identifying characteristics of different landscapes that makes them easier or harder to learn. For example, two-part tariff appears quite resistant to nonstationarity; the reason for this can be seen by examining a topological depiction of the two-part tariff landscape, shown in figure 5. The landscape consists of a single large hill with a large plateau at the top. The "northeast" face of this hill contains a sharp dropoff. Optima are scattered on this plateau, all with similar profits. Once amoeba finds this plateau, it is easy to move between optima without significant cost. As the consumer population changes, the exact angle and direction of the dropoff changes, moving the plateau somewhat, but the landscape retains its overall structure. Figure 5 shows a contour map of a sample two-part landscape. Superimposed upon this are the results of running amoeba on 100 randomly generated landscapes. As we can see from this figure, optima tend to be located on or near the plateau, regardless of the particular



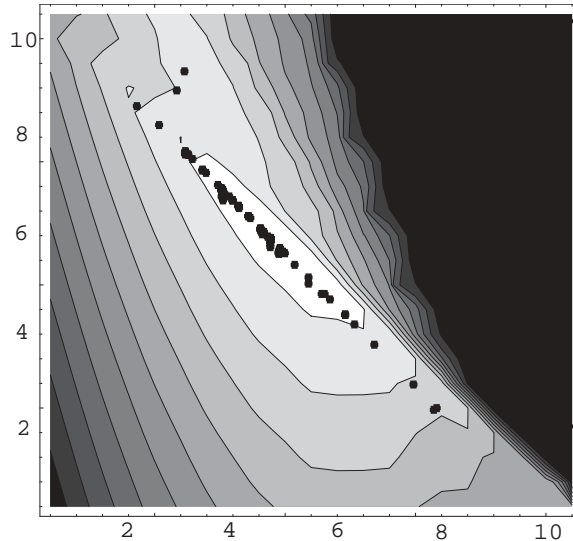


Figure 5: One instance of a two-part tariff landscape, as seen from above. Notice the plateau at the top of the hill, and the sharp drop to the northwest. Dots indicate the global optima from 100 different randomly generated landscapes.

instantiation of the consumer population. This implies that it is easy for a producer to track consumer change using this schedule. Even though the optima move, the landscape’s overall structure is stable enough that, once a producer finds the plateau at the top, it is able to stay on it and easily track optima.

## 5 Dynamic Schedule Selection

In the previous sections, we have assumed that the choice of a pricing schedule is a one-and-for-all decision. However, it may be more advantageous for a producer to instead change the schedule it is offering as information is acquired. Initially, a simple schedule is learned, allowing a producer to accrue some profit quickly. As information about consumer preferences is gathered, the producer eventually switches to a higher-precision schedule, thereby earning higher profits. To get an intuitive idea of how this works, consider two-part tariff, a two-parameter schedule, and linear pricing, a one-parameter schedule. As we saw in section 3, linear pricing is easily learned, but two-part tariff yields higher profit once learned. Now consider that linear pricing is simply two-part tariff with a subscription fee of zero. This means that when a producer has learned a linear schedule, it has learned a slice of the landscape for two-part tariff. It need not start over, since this solution is hopefully somewhat near to the solution in the higher-dimensional space. In this section, we will make this idea more concrete and describe it in terms of a decision procedure that a producer can engage in to select the optimal sequence of schedules to learn with.

### 5.1 Models and Model Families

Recall that a price schedule can be thought of as a model of consumer demand. In this context, we use the term ‘model’ to refer to a set of features of interest and an assignment of values to each of those features. If a consumer’s valuation for each good in a bundle were known, it could be expressed as a nonlinear price schedule charging that consumer’s valuation for each good. This is a model which contains  $N$  features, one for each article, and a numerical valuation associated with each feature. Other price schedules are less precise, but still act as models of consumer demand.

To begin, we introduce some terminology to help with the decision problem of selecting price schedules. We refer to the set of all price schedules of a given type as a *model family*. A model family is the set of all models which share the same set of features; however, there is no binding of value assignments to these features in a model family. For example, the family of linear pricing models consists of all models which use a single parameter to represent per-article price. When a producer decides to use linear pricing, it is actually selecting a model family; it then searches through the space of models in that family for the one that maximizes profit.

Pricing Schedule	Number of Parameters	Profit	Precision
Pure Bundling	1	.875	0.76
Linear Pricing	1	.875	0.76
Two Part Tariffs	2	1.037	0.90
Mixed Bundling	2	1.037	0.90
Block Pricing	3	1.094	0.95
Nonlinear Pricing	$N$	1.152	1.0

Table 3: An example of precision for various pricing schedules for a consumer population with  $w = 10$ ,  $N = \infty$ , and  $k$  drawn from  $U[0.0, 0.7]$

A model family can be characterized by its *precision* and its *complexity*. Precision indicates the fraction of profit that the best model in this family can capture. In other words, precision tells us how well we can hope to do using a model from a given family. For example, we see from table 1 that linear pricing yields a maximum per-consumer, per-article profit of 0.875. If the total possible profit available is 1.152, then the family of linear pricing models has a precision of  $\frac{0.875}{1.152} = 0.76$ . Precision for each family of price schedules is shown in table 5.1.

Within a model family, we can also consider the *accuracy* of a model. A model’s accuracy indicates the fraction of profit it extracts, relative to the optimal model in its family. If  $M^*$  is the optimal model in a given family, then the accuracy of a given model  $M$  is  $\frac{\pi(M)}{\pi(M^*)}$ , where  $\pi(M)$  indicates the profit yielded by a model. For example, if we had a linear pricing schedule which yielded a normalized profit of 0.6, its accuracy would be  $\frac{0.6}{0.875} = 0.68$ .

While precision and accuracy serve to capture the static characteristics of a model and its family, they don’t say anything about how easily a model family can be learned. To capture this, we must characterize a model family’s *complexity*. We propose a PAC-learning style framework in which a model family  $\mathbf{MF}$ ’s complexity  $C(\mathbf{MF}, p, \epsilon)$  indicates the number of samples  $n$  needed to reach an accuracy of  $\epsilon$  with probability  $p$ . The more complex a model family is, the longer a learner will need (on average) to attain a given accuracy. Of course, more complex model families will likely have a higher precision, assuming that the behavior of the agent being modeled is actually complex.

The actual calculation of complexity will depend upon the specifics of the domain at interest. In some cases, either PAC-learning estimates [11] or Monte Carlo estimation [13] can be applied. Unfortunately, our in our model neither are applicable, since the trials (price-profit offerings) are not independent and some schedules yield higher reward than others, rather than there being a set of schedules that are “equally bad.”

Empirically, we have found that a sigmoid function works well for approximating complexity, at least when applied to pricing data. Since profit landscapes tend to be composed of large plateaus, dotted with hills that are easy to locate and ascend initially, but which have multiple peaks, we would like to estimate complexity with a function that increases steeply initially, then tails off as it approaches an asymptote. We have had good results when applying the following formula as an estimate of the solution quality  $Q$  (an inverse of complexity) after  $n$  iterations:

$$(2) \quad Q(MF, n) = \frac{\text{precision}_{MF}}{1 + e^{\frac{-n}{\log_2 N_{d_{MF}}}}}$$

where  $d_{MF}$  is the dimensionality of the model family. An analytical proof of complexity is difficult to construct in a domain-independent manner; our current work involves using landscape theory [10, 9] to characterize theoretical properties of profit landscapes for this purpose. For the decision procedure described in the following section, it does not particularly matter where a producer’s complexity estimate comes from, only that it has one. Of course, more accurate estimates of complexity yield more optimal decisions, and therefore greater profit.

For example, a producer that knew it would have 20 iterations to interact with a consumer population could use this formula to predict its performance using two-part tariff. Two-part tariff has a precision of 0.90, from table 1, and its dimensionality is 2. For example, assume that prices are within the range [0,100] and  $N = 100$ . Plugging in these numbers, we find that  $Q(\text{Twopart}, 20) = 0.736$ . So, in the 20th iteration, a producer using 2-part tariff would extract  $\frac{0.736}{1.152} = 64\%$  of the available profit (in expectation). By summing this calculation over each iteration 0-19, we get the total profit (per customer). A producer using 2-part tariff for 20 iterations can expect to earn a total undiscounted profit of 12.09, or 52% of the available profit. Similarly, a producer using block pricing would have  $Q(\text{Block}, 20) = 0.695$  and a cumulative profit of 11.79, so two-part tariff would be a better model family. At 200 iterations, this has changed. Now two-part tariff has  $Q = 0.899$  and cumulative profit of 171.7, whereas block pricing

has  $Q = 0.949$  and a cumulative profit of 176.8. At this point, both schedules are nearly learned, and block pricing’s higher precision begins to dominate.

## 5.2 Schedule Selection

Now that we have a way for a producer to compare the performance of different price schedules, we are almost ready to propose a decision procedure for sequentially selecting schedules. There is one more detail to consider. Our description of complexity tells how quickly a price schedule is learned when the producer is using *that schedule*. Recall our previous example that, when linear pricing is offered, the producer also learns something about two-part tariff. We must characterize the amount of information learned about more complex schedules from experimentation with simpler ones. We accomplish this using a simple assumption: the inputs of a  $d$ -dimensional schedule are a subset of the inputs of a  $d + 1$ -dimensional schedule. In other words, a simple schedule can be thought of as a more complex schedule with some inputs held constant. As a producer is learning a lower-dimensional schedule, it can also update its model of a more complex schedule. The question then becomes: how quickly is a higher-dimensional schedule being learned, using the information acquired with a lower-dimensional schedule?

Our solution is to weight the number of data points seen by the more complex model by the ratio of dimensionalities of the two models. That is, if a producer collected  $n$  data points from a model of dimensionality  $d$ , we would treat this as  $n \frac{d}{d+k}$  data points for the purposes of evaluating the complexity of a model of dimensionality  $d + k$ .

If we let the total available surplus in a population equal  $S$ , then the most that any model family  $MF$  can extract is  $precision(MF) \times S$ . If we assume that the accuracy of models in a family fits an unskewed distribution so that, in expectation, a randomly chosen model  $M'$  in  $MF$  has an accuracy of 0.5, and we know that  $n = C(MF, pr, \epsilon)$  is the number of iterations needed to learn a model in  $MF$  (from scratch) to within  $\epsilon$  with probability  $p$ , then, for any model  $M^*$  in  $MF$  with an accuracy  $a$ , it will take  $n' = n(0.5a - 1)$  iterations to learn  $M^*$  within  $\epsilon$  with probability  $p$ .

So, if a producer is to learn using a given model  $M_i$  from family  $MF$  for  $m$  iterations, it can expect to receive a profit of:

$$(3) \quad \Pi_{M_i} = \sum_{i=0}^{n'} (precision(\mathbf{M}) * accuracy(M_i) * S) + \sum_{i=0}^{m-n'} (precision(\mathbf{MF}) * S)$$

If we assume that a producer will make this decision once and only once, then this reduces to the problem described in section 3, and the producer selects the schedule with the highest expected profit. If a producer is able to make this decision more than once, it can break the  $m$  iterations into subintervals and use any algorithm for solving a Markov decision process, such as dynamic programming or greedy search, to select the optimal series of schedules. If the number of subintervals is large, the problem will be intractable using an optimal procedure such as dynamic programming, and so heuristic methods will be necessary.

We implemented this procedure within a producer which continually reexamined its selection of price schedule every iteration and used a greedy approach; that is, it continually selected the schedule it believed would yield the greatest aggregate profit in the next iteration, given current information. As in previous sections, the producer used amoeba to search the profit landscapes. A model of each price schedule was maintained, and all were updated in parallel.

A graph comparing the performance of an adaptive producer to producers using a fixed price schedule is shown in figure 6.

As we can see from the graph, the adaptive producer (denoted as ‘Adaptive Pricing’) is able to outperform most of the static schedules for the majority of the experiment. The schedules used by the adaptive producer are indicated at the top of the figure. It begins by trying a number of different schedules to collect data, and then settles on linear pricing for approximately 35 iterations. It then switches to two-part tariff for about 60 iterations before moving on to block pricing. Finally, at around 300 iterations, it switches to nonlinear pricing. Since the graph averages performance over a number of iterations, these boundaries are approximate; in different runs, the adaptive producer changed its schedule at different times. This experiment illustrates how the producer uses the simpler schedules to bootstrap itself into more complex schedules: linear pricing is two-part tariff with a fee of 0, and two-part tariff is block pricing with a threshold of 1.

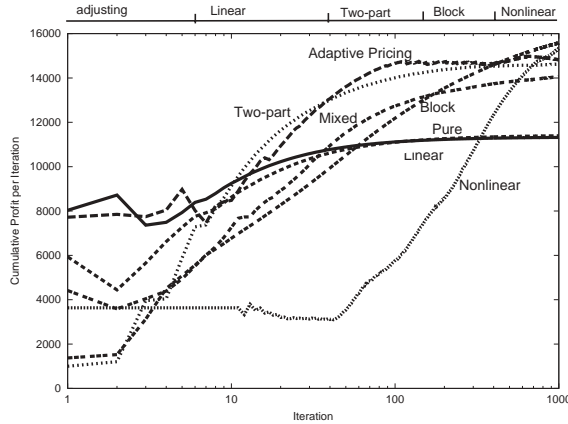


Figure 6: Comparison of an adaptive pricing producer to producers using fixed schedules. The Y axis indicates cumulative profit per article, per period, per customer. ( $N = 10$ ) (averaged over 10 runs)

Also, we note that there is no static schedule that dominates adaptive pricing over the course of the entire experiment. This implies that if a producer is unsure of how long it will have to interact with a population, adaptive pricing is an effective approach. In addition, adaptive pricing performs particularly well in the ‘intermediate’ range of 10-200 iterations. In this range, learning is able to play an important role. If a producer has only a few iterations to interact with a population, there is not enough time to learn much. (Recall the  $N = 100$  experiments shown in figure 3, where nonlinear pricing’s random guesses initially performed well.) Conversely, if a producer has a long time to interact with a given population, nonlinear schedules can be learned exactly and long-run steady-state profits will dominate.

## 6 Discussion

In this paper, we have addressed the problem of a monopolist producer selecting a model of a population to learn, when it must account for the cost of learning. We have demonstrated how, in a single-decision scenario, learning costs can outweigh optimal profits if there are a small number of periods of interaction. In general, moderately complex schedules appear to perform well in terms of the exploitation/exploration tradeoff.

Moderately complex schedules, particularly two-part tariff, also perform well under moderate amounts of nonstationarity. The distribution of their optima allows a learning producer to adapt without too much unnecessary exploration. As the frequency of shocks to the consumer population increases, it becomes more difficult for a producer to adapt its model in time, and schedules which tend to learn the mean of the generating distribution are more effective than adapting to constant change.

Finally, we have illustrated how a producer can continually update its selection of schedules, estimating both the information gained from past experiments with other schedules and the expected future performance of a schedule so as to make an informed, decision-theoretic selection as to the model(s) that will maximize aggregate profit.

There are still plenty of open questions regarding this model. As discussed above, we are working to extend our estimates of model complexity using statistical characteristics of profit landscapes. Our model of nonstationarity could also be extended to an evolutionary approach in which consumer entry and exit is a function of producer behavior. Finally, we plan to more closely reconcile this work with our research involving multiple learning producers [3, 8] and characterize the selection of learning models in a competitive environment.

## 7 Acknowledgments

We thank Scott Fay for his contributions to this research, in particular the complete-information solutions for each price schedule. This work was supported in part by NSF grants IIS-9872057 and IIS-0112669.

## References

- [1] Avrin L. Blum. On-line algorithms in machine learning. Technical Report CMU-CS-97-163, Carnegie Mellon University, 1997.
- [2] David J. Braden and Shmuel S. Oren. Nonlinear pricing to produce information. *Marketing Science*, 13(3):310–326, 1994.
- [3] C. H. Brooks, E. H. Durfee, and R. Das. Price wars and niche discovery in an information economy. In *Proceedings of ACM Conference on Electronic Commerce (EC-00)*, Minneapolis, MN, October 2000.
- [4] C. H. Brooks, S. Fay, R. Das, J. K. MacKie-Mason, J. O. Kephart, and E. H. Durfee. Automated strategy searches in an electronic goods market: Learning and complex price schedules. In *Proceedings of ACM EC-99*, pages 31–40, 1999.
- [5] John Chung-I Chuang and Marvin A. Sirbu. Network delivery of information goods: Optimal pricing of articles and subscriptions. In *Proceedings of the Conference on Economics of Digital Information and Intellectual Property*, Cambridge, MA, January 1997.
- [6] Scott A. Fay and Jeff MacKie-Mason. Competition between firms that bundle information goods. In *Proceedings of the 27th Annual Telecom Policy Research Conference*, Alexandria, VA, September 1999.
- [7] A. A. Fe'ldbaum. *Optimal Control Systems*. Academic Press, New York, 1965.
- [8] Robert S. Gazzale and Jeffrey K. MacKie-Mason. Endogenous differentiation of information goods under uncertainty. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, FL, July 2001.
- [9] Wim Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4):336–366, 1996.
- [10] Stuart Kauffman. *Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York, 1993.
- [11] Michael Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.
- [12] Jeffrey O. Kephart, Christopher H. Brooks, Rajarshi Das, Jeffrey K. MacKie-Mason, Robert S. Gazzale, and Edmund H. Durfee. Pricing information bundles in a dynamic environment. In *Proceedings of the 2001 ACM Conference on Electronic Commerce*, 2001.
- [13] David J.C. MacKay. An introduction to Monte Carlo methods. Appeared in the Proceedings of the 1996 Erice Summer School., 1996.
- [14] Eric Maskin and John Riley. Monopoly with incomplete information. *Rand Journal of Economics*, 15(2):171–196, 1984.
- [15] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
- [16] William H. Press. *Numerical Recipes*. Cambridge University Press, 1992.
- [17] Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9:185–202, 1974.
- [18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [19] Sebastian B. Thrun and Knut Møller. Active exploration in dynamic environments. In J.E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann, San Mateo, CA, 1992.
- [20] Frederick H. Walters, Jr. Lloyd R. Parker, Stephen L. Morgan, and Stanley N. Deming. *Sequential Simplex Optimization: a technique for improving quality and productivity in research, development, and manufacturing*. CRC Press, 1991.

[21] Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979.

[22] Robert B. Wilson. *Nonlinear Pricing*. Oxford University Press, 1993.