



Lancaster University
Management School

Lancaster University Management School
Working Paper 2018:3

Non-parametric dynamic pricing: a non-adversarial robust optimization approach

Trivikram Dokka, Peter Jacko and Waseem Aslam

*The Department of Management Science
Lancaster University Management School
Lancaster LA1 4YX
UK*

© Trivikram Dokka, Peter Jacko and Waseem Aslam
All rights reserved. Short sections of text, not to exceed
two paragraphs, may be quoted without explicit permission,
provided that full acknowledgment is given.

The LUMS Working Papers series can be accessed at <http://www.lums.lancs.ac.uk/publications>
LUMS home page: <http://www.lums.lancs.ac.uk>

Non-parametric dynamic pricing: a non-adversarial robust optimization approach^{*}

Trivikram Dokka[†] Peter Jacko Waseem Aslam

April 17, 2018

Abstract

We consider a single-product finite-horizon dynamic pricing problem where a seller seeks to maximize the expected revenue while learning how to dynamically adjust the product price, without assuming a shape of the demand function. In this work we propose a non-adversarial robust optimization approach to the non-parametric dynamic pricing, where after the seller announces her price, and for a given level of her conservatism (captured by the statistical confidence level used for estimating the demand function), nature chooses a price distribution for the alternative product which minimizes the expected cost of the customer during that period. Our approach is thus different from the traditional robust optimization approach which assumes an adversarial setting, where nature chooses a price distribution for the alternative product which minimizes the expected revenue of the seller. We show that our algorithm is competitive, using simulated and real data, even without optimizing its parameters compared to the optimized version of the algorithm proposed in Besbes & Zeevi [Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420 (2009)]. We present a sensitivity analysis to show that further improvement in the performance is possible depending on the problem setting and on the decision maker’s level of conservatism.

Keywords: dynamic pricing; learning algorithm; robust optimization

^{*}Working paper

[†]Corresponding author, Department of Management Science, Lancaster University, UK, t.dokka@lancaster.ac.uk

1 Introduction

1.1 Motivation

Setting a price which generates highest revenue is an important problem faced by a seller of a product or service in many industries. The revenue maximizing price is generally not known to the seller and good (near-optimal) prices can often be learned by experimenting with a number of different prices. However, it is important that the seller finds a good balance between revenue maximization and price experimentation, for two main reasons: (1) setting a sub-optimal price means a loss of potential revenue and (2) changing the price too frequently or too dramatically affects the customers' perception of the product/service and/or may not even be feasible due to strategic or contractual constraints. As a result, sometimes it is better for the seller to settle, at least for a certain period of time, at a sub-optimal price instead of continuing price experimentation.

While this problem has been extensively studied in academic literature, the modelling assumptions are often unrealistic for practical use of the results. For instance, some models assume that the demand is deterministic and known as a function of price, other models assume that the demand is stochastic and a distributional form of the demand function is known. In these cases, typically, no experimentation is needed as the optimal price can be derived theoretically or computed numerically using optimization and/or simulation techniques.

Price experimentation (leading to dynamic pricing) becomes important in more advanced models, in which the demand is assumed to be stochastic, but the parameters of the distributional form of the demand function are not known. The seller learns these parameters by price experimentation and eventually fits the model at a sufficiently high precision to find an optimal price. However, in many situations such distributional assumptions may not be correct and both price experimentation and settling at an optimal price under wrong assumptions can be damaging to the seller's revenue. In practice, the distributional form may not be known at all and it is desirable to address this problem without making any assumptions on the distributional form of the demand function, which is the objective of this paper.

A simple such pricing policy first experiments with a subset of prices chosen to cover the entire range of allowable prices (usually known to seller)—*learning phase*—and observes the demand at each of these prices by setting them for some pre-decided periods of time, and then sets the price to one of those that yielded highest revenue—*earning phase*.

In [6], it is shown that a pricing policy like this is asymptotically optimal. Such a policy, however, is often not practical. Moreover, for many classes of products and services, the seller often faces restrictions such as price caps imposed by regulatory authorities, supply chains contracts, and/or seller’s marketing strategy. The reason this simple policy may not be feasible for such products is that seller is not free to change the price whenever she finds it desirable and to whatever price she finds it appropriate. For instance, if there is a cap on the price increase, the seller can end up in a situation with a very low price experimented at the end of the learning phase, being unable to set a higher price which she knows yields a significantly higher revenue. Another drawback of this policy is that it waits for price experimentation to finish with no attempt to use the information collected so far to adapt to in the rest of the learning phase.

In this paper, we propose a *non-parametric dynamic pricing* policy, which balances price experimentation with revenue maximization. It is non-parametric, because we do not assume a particular distributional form of the demand function; we only require the range of allowable prices to be known. We deploy a novel, *non-adversarial robust optimization approach*, which we extend and significantly generalize to a multi-period setting from the recently proposed static formulation in [9]. The approach is robust in two ways. First, in the sense that instead of using means, we use lower confidence bounds, which can be seen as having a risk-averse seller or a seller that takes a conservative stand when facing uncertainty in making estimations from data. It is also robust due to the formulation of a subroutine using a bilevel optimization problem, where the upper objective maximizes the expected revenue. The lower objective minimizes the expected cost of customer(s), which differentiates our approach from the traditional robust optimization approach in which the lower objective is adversarial (i.e., where nature chooses a price distribution for the alternative product which minimizes the expected revenue of the seller).

The idea of our proposed pricing policy is to fit a discrete distribution on the prices with the data collected so far—*estimation step*—and then use the mean and variance of this estimated distribution to find a revenue maximizing price according to a two-point distribution with the same mean and variance by taking price increase cap explicitly into account—*pricing step*. While the estimation step is backed-up by statistical theory, the pricing step is backed-up by optimization theory and balances the price experimentation (avoiding to get stuck to one price) with revenue maximization (sticking to the best price). Thus, our proposed pricing policy tries to set the price close to optimal while at the same time vary the price enough to learn the revenue function. To initialize the policy, there

is also a *forced exploration phase*, which is designed to get out of the range of uninteresting prices (with very small demand) as quickly as possible.

There are several advantages of our proposed pricing policy with respect to the simple policy described above:

1. we allow the price to be adaptively changed whenever there is sufficient evidence from past data that a change is beneficial, i.e. without formally splitting the horizon into learning and earning phases;
2. we can incorporate price caps;
3. we can take into account seller's level of risk-aversion.

1.2 Price Caps and Toll Roads

There is a long history of regulatory price controls used by governments in regulating the prices and usage of various important goods and products. Price controls were used by governments as early as the years following the French Revolution. Historically, there are numerous cases where governments around the world used some sort of price control in regulating markets for products ranging from oil, drugs, toll roads and property rents. One important reason for this is price control is seen by many as an easy political instrument to gain popularity. In many of these instances the private companies involved in selling products with price control have seen heavy losses. Naturally, many economists over the years reasoned against the use of price controls. Despite this, there are many instances where governments impose price controls.[1] notes that rate of return regulation is widely used in sectors like energy, telecommunications and transport. Under such regulation, firms selling products are not allowed to earn above a certain rate of return and price rises are capped. Authors also note that capped increase is usually linked to inflation index.

Our main inspiration to study dynamic pricing comes from toll road pricing. It is a common practice in countries like India and Indonesia that governments regularly intervene in toll road prices and increases in tolls often have unpleasant consequences involving violent opposition from public, opposition parties. A constant toll price on the other hand can result in loss of revenues and may discourage private companies to enter this business which in turn can make road and infrastructure financing next to impossible. It is necessary to allow toll operator some freedom of price increase with restrictions on the

increase. A small or a capped increase on price should be promoted as it can be easier for customers to incorporate it in their spending patterns and gives the toll operators opportunities to better manage their revenues. We quote from [7], “Careful contract negotiations can constrain maximum toll increases. The recent National Surface Transportation Policy and Revenue Commission report recommended capping toll rate increases at the level of the CPI, adjusted by productivity. Tolls on the Indiana Toll Road are scheduled by the Indiana legislature through June 2010. Thereafter, maximum annual increases for all vehicles are capped at the greater of 2%, CPI, or per capita nominal growth in gross domestic product (GDP). Tolls on the Chicago Skyway are scheduled in the lease agreement until 2017, with maximum annual increases capped at the greater of 2%, CPI, or per capita nominal GDP growth beyond 2017. Tolls on the Pocahontas Parkway in Richmond, Virginia, are specified until 2016, and annual increases are capped at the greater of 2.8%, CPI, or per capita real GDP growth thereafter”. Our motivation for considering price control in such problems comes from the public sector, where a third party authority (like the government or a governmental regulation agency) can introduce price controls on the product provided by a revenue-maximizer involved in a public-private partnership or a similar scheme.

With restrictions like price caps seller is very constrained in price experimentation. Price caps require to set the tolls as robustly as possible to avoid any need for a price hike.

1.3 Related Literature on Dynamic Pricing

The *dynamic pricing problem* (DPP), in which a seller dynamically varies the price of a product with the objective of maximizing the expected revenue over a finite time horizon, is a fundamental problem in revenue management. Such modeling framework has seen useful applications in the study of pricing problems which occur in many real world situations, such as for example, in the airline and retail industries (see [16]). In the traditional models, dynamic pricing arises as a consequence of having a time-varying demand. In our case, however, dynamic pricing arises as a consequence of the need to learn about the problem, and in the following discussion we focus on models related to our case.

The key in such problems is understanding and estimating of the customers’ willingness-to-pay for the product. It is important to understand the variability and estimate the price of alternative products available to the customer, which ultimately affect her willingness-to-pay for the product. When the price of the alternative products randomly varies over time following a known distribution, the DPP is well understood ([10]). When the price

distribution of the alternative products is unknown and prices of the alternative products are difficult to observe at any particular time, the seller needs to learn/estimate it in the form of customers' willingness-to-pay by observing the customers' purchasing decisions over time.

Within the existing literature on the dynamic pricing problems (see [17] for a more exhaustive survey), the most popular studies can be categorized as parametric approaches. In these studies, the unknown willingness-to-pay function is in a parameterized family of distributions with possibly unknown parameters. The algorithms using this framework learn these parameters using price experimentation. See, e.g., [2], [12] and [15].

More recently, an increasing number of studies employ a non-parametric approach, that is, they do not assume a parametric functional form for willingness-to-pay function. These approaches generally fall into the robust decision making framework and aim to maximize the worst case revenue. For example, [4] and [5] study robust static pricing problems formulated as minmax regret problems. [14] formulates the robust dynamic pricing problem by modeling an adversarial nature choosing a point distribution characterizing the demand. However, this robust approach suffers from a disadvantage that prices can end up stuck to very low values, which may yield revenue that is far from optimal. [6] observed that such conservativeness precludes real-time learning, and does not lend itself to prescriptive solutions. Instead, [6] established the asymptotic optimality of a simple algorithm, which splits the horizon into a learning and earning phase. See also [19, 13, 18] for similar approaches.

While there is some literature that studies dynamic pricing and learning-vs.-earning problems, to the best of our knowledge, none addresses the case when the seller has limited price revising opportunity due to price caps imposed. Existing dynamic pricing algorithms unfortunately do not extend easily to such a setting.

1.4 Related Literature on Learning vs. Earning

Our work also have close ties with the multi-armed bandit (MAB) problem, which is a fundamental model for resolving the learning vs. earning trade-off (see, e.g. [11] for an operations research approach and [?] for a machine learning approach). There are several variants of MAB problems, including reward-maximizing and adversarial. The analogy is that in every period, one arm needs to be pulled to observe a random reward from that arm (one price needs to be set to observe a random revenue under that price).

Using this analogy, a few papers have focused on extending the results from MAB

literature to the dynamic pricing; for example, [3] gives an algorithm similar to the well-known UCB1 algorithm for solving MABs. These algorithms typically require all prices to be used at least once, and calculate a priority index value for each price at the beginning of every period, when a price with the highest priority index value is chosen to be used for that period.

In summary, there are fundamental differences between the MAB and dynamic pricing problems. In MAB problems, there are usually few arms and (infinitely) many periods, while in our problem there are many allowable prices and relatively few periods, so it may not be possible to try all the allowable prices. Another distinctive feature is that revenues under different prices in our case are not independent, while MAB considers that arms are independent. Finally, we consider existence of restrictions on choosing a price (e.g., caps), while MAB usually do not restrict availability of the arms.

1.5 Our Contributions and Paper Structure

We consider two generic variants of the above described problem, which both lead to the same model, only with a different interpretation of one of the parameters.

Variant 1. Consider a product whose price can be changed occasionally, e.g., monthly or annually, and a single customer that may purchase this product, or an alternative one (including none), on a daily basis. Note that the consideration of a single customer is not restrictive, as she can be taken as a representative, or average, customer.

Variant 2. Consider a product whose price can be changed frequently, e.g., daily, and a known number of customers that may purchase this product, or an alternative one (including none), on a daily basis.

In both variants, the product can be sold at most $N^{(i)}$ times while the price is maintained during the i -th pricing period, where $N^{(i)}$ is interpreted as the number of days per month or year in Variant 1, and as the number of customers per day in Variant 2. Variant 1 is motivated by toll road pricing, which is described in more detail in [Section 2](#), where we formulate the problem mathematically, and to which we refer throughout the paper.

[Section 3](#) describes the approach, design and properties of our proposed non-parametric dynamic pricing algorithm. Our algorithm is developed using a novel non-adversarial robust optimization approach, complemented with a conservative statistical estimation. We

further establish a convergence rate of our proposed algorithm. [Section 4](#) describes an extensive numerical experiments, in which the performance of our proposed algorithm is compared to an algorithm proposed in [6], which is asymptotically optimal, and to which we refer to as the BZ algorithm. We illustrate the performance on both real and simulated data, and show that our proposed algorithm with default parameters performs competitively, even when compared to an optimized version of the BZ algorithm. A sensitivity analysis of a number of parameters is performed in [Section 5](#), based on which we conclude that an optimized version of our algorithm typically dramatically outperforms the BZ algorithm. [Section 6](#) concludes the paper and outlines possible avenues for further research.

2 Problem

In this section we formulate the finite-horizon dynamic pricing problem as considered in this paper. For the sake of concreteness, we use the setting of our motivating example of a road toll pricing. Nevertheless, both the model described in this section and our solution approach described in [Section 3](#) are general and may be applied to many other dynamic pricing contexts.

We further remark that our problem corresponds to the finite-horizon dynamic pricing of a perishable product with no inventory constraints, though the formulation can be extended to the case with limited inventory in a straightforward way.

2.1 Problem Formulation

We consider a parallel road network with two parallel roads connecting the origin and destination. There is a toll road that needs to be dynamically priced and an alternative public (no-toll) road. Besides the *toll price* on the toll road, these two roads are distinguishable by customers, for example in construction quality, weather conditions, speed limits, driving comfort, congestion levels, accidents intensity and severity, obstructive works intensity, services offered, etc. We will include all these parameters in a single (relative) *cost* parameter of the public road, which captures the differences between the two roads, and which possibly varies over time. The toll-setter is, however, unable to observe/measure and correctly predict this cost in a quantifiable manner. For simplicity of exposition, in our model we consider the road choice of a single customer.

Timing. We take a day as a base time unit. The time horizon considered by the toll-setter is T days, labelled by $t \in \mathcal{T} := \{0, 1, \dots, T-1\}$. The horizon is split into I pricing periods. The i -th pricing period, for $i \in \mathcal{I} := \{0, 1, \dots, I-1\}$, starts on day $t^{(i)}$ and has a duration of $N^{(i)} \geq 1$ days. A pricing period is a time interval during which the toll price cannot be changed. The beginnings of the pricing period starting days are called the *decision epochs* and allow the toll price to be changed. That is, the initial toll price decision is taken at the beginning of day 0, which is decision epoch 0. We assume that toll price changes are instantaneous.

Demand. We assume that there is one customer traveling between origin and destination each day, i.e., demand every day is constant. However, our model is not constrained by this assumption since non-constant demand in each pricing period is captured in the model by the (possibly) variable period duration.

Unobserved Variables. On every day there is a cost on the public road, possibly varying from day to day, observed by the customer and kept as a private information of the customer, i.e., it is unobserved by the toll-setter. We assume that the customer's willingness-to-pay is the minimum between the toll price and the public road cost, and thus every day she chooses to use the cheaper road (ties are resolved in favor of the toll road).

Observed Variables. At the end of i -th pricing period, for every i , the toll-setter observes the *pricing period usage* $u^{(i)} \in \{0, 1, \dots, N^{(i)}\}$, i.e., the number of days on which the customer chose the toll road during this pricing period.

Decision Variables. Based on the past observations of the pricing period usage and past toll prices, the toll-setter decides at every time epoch i the toll price $p^{(i)}$, which must be paid by the customer every day she chooses the toll road during the i -th pricing period. We allow for setting the toll prices from a discrete set $\Omega := \{\omega_j : j \in \mathcal{J}\}$, where $\mathcal{J} := \{1, 2, \dots, J\}$ is the allowed price index set and the allowed prices satisfy $0 < \omega_1 < \omega_2 < \dots < \omega_J$. Moreover, we denote by $q := \omega_1$ and by $Q := \omega_J$.

Price Caps. The toll price $p^{(i)}$, for $i \geq 1$, cannot be set higher than

$$Q^{(i)} := \min\{\omega \in \Omega : \omega \geq (1 + \xi)p^{(i-1)}\}, \quad (2.1)$$

where $\xi \geq 0$ is a given *price cap percentage*, though note that it is possible to round up to the next allowed price in order to avoid getting stuck at the previous price. The price cap percentage ξ is set to $+\infty$ when there is no cap on the price increase.

Objective. For a given sequence of toll prices $(p^{(i)})_{i \in \mathcal{I}}$, the *total revenue* collected from toll prices over the time horizon is

$$\sum_{i \in \mathcal{I}} u^{(i)} \cdot p^{(i)}. \quad (2.2)$$

The objective of the toll-setter is to find a non-anticipating sequence of toll prices that maximise (or come close to maximising) the expected total revenue over the time horizon.

2.2 Stochastic Model of Unobserved Variables

The public road cost, possibly varying from day to day, can be modelled by a random variable, which we denote by C and which follows an unknown distribution F . Without loss of generality we assume that F belongs to a set \mathcal{D} of distributions with discrete support given by Ω . The actual cost on day $t \in \mathcal{T}$, c_t , can be seen as an identically distributed random draw from distribution F .

The variations in the public road cost from day to day are in practice likely to be limited, so they may not be independent. In our model we capture this by assuming that the variability of the distribution F is constrained, in the way that the variance-to-mean ratio is bounded by an unknown constant κ . This is reasonable, especially given the fact that we have a bounded support Ω .

Note that we thus view the pricing period usage $u^{(i)}$ in every pricing period i as a random variable, as it depends on the toll price $p^{(i)}$ and on the day-to-day public road costs drawn from distribution F .

3 Non-parametric Dynamic Pricing

3.1 Non-adversarial Robust Optimization Approach

As noted in the introduction, [9] proposed a bilevel distributionally robust approach to overcome the drawback of adversarial nature in traditional robust optimization approach. To briefly review that methodology, let us first consider the case when the toll setter can

only set the toll price once, that is, $I = 1$, which we call the *static pricing problem*. To simplify the notation in this subsection, we drop the dependence of all the variables on the pricing period. In this setting it is assumed that the toll setter has access to historical information about costs on the public road from which toll setter forms her estimate μ of the mean of the public road cost C and about the variability bound κ . She then calculates an optimal toll price by solving the following bilevel program:

$$\max_{p \in \Omega} \sum_{j: \omega_j \geq p} x_j p \quad (3.1)$$

$$\min_{X \in \mathcal{D}} \sum_{j: \omega_j \geq p} x_j p + \sum_{j: \omega_j < p} x_j \omega_j \quad (3.2)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} x_j = 1 \quad (3.3)$$

$$\sum_{j \in \mathcal{J}} x_j \omega_j = \mu \quad (3.4)$$

$$\sum_{j \in \mathcal{J}} x_j \omega_j^2 \leq \kappa \mu + \mu^2 \quad (3.5)$$

$$0 \leq x_j \leq 1 \quad \forall j \in \mathcal{J} \quad (3.6)$$

The lower level optimization is over all the distributions in \mathcal{D} where x_j corresponds to the probability mass put on the price indexed by $j \in \mathcal{J}$. The upper level decision variable is the toll price p . The upper level objective (3.1) is to maximize the expected daily toll revenue, which is obtained by earning p whenever the public road cost is at least as high as p . The lower level objective (3.2) corresponds to nature's problem which acts to favor the customer (minimizing the expected cost of the customer) rather than acting as an adversary to the toll setter (in which case it would minimize the expected revenue of the toll setter).

Nature will choose a distribution over Ω (cf. (3.3) and (3.6)) with the specified mean via (3.4) and its variance-to-mean ratio bounded by κ via (3.5).

A natural way of interpreting the non-adversarial nature (i.e., the lower level objective (3.2)) is by seeing it as an extension of the deterministic case, in which the public road cost is constant and known, and where in the lower level the customer acts to minimize her costs. As argued in [9] the approach is closely connected to worse-case conditional value-at-risk and the principle of maximum entropy.

A simple strategy to solve (3.1)–(3.6) is to solve (3.2)–(3.6) for each value of $p \in \Omega$.

Since the lower level problem is LP with three constraints, the number of basic variables in the optimal solution is at most equal to three. In other words, the optimal solution to the lower level problem is always a distribution with support of at most three points (allowed prices). Note that in [9] it is shown that the optimal solution to the lower level problem is a two-point distribution provided Ω has sufficient granularity.

3.2 Our Algorithm

In this section we introduce an algorithm which extends the non-adversarial robust optimization approach described in [subsection 3.1](#) to the dynamic, multi-period setting. Our proposed algorithm, which we call the *Dokka-Jacko-Aslam (DJA) algorithm*, should be run at every decision epoch. It has two main steps: in the first step (estimation), it estimates the distribution of C ; in the second step (pricing), it calculates an optimal price, using the mean and the variance of the estimated distribution. In the initial few periods, these are overridden so that a minimal exploration in the price set is achieved. We describe the algorithm formally in [Algorithm 3](#). Next we explain each of these steps in detail.

At the i -th decision epoch, where $i \geq 1$, let $\mathcal{K}^{(i)}$ be the discriminative price index set of all the past prices $p^{(0)}, p^{(1)}, \dots, p^{(i-1)}$ (excluding the lowest price ω_1 if used), and let $\Omega^{(i)} := \{\omega_k \in \Omega : k \in \mathcal{K}^{(i)}\}$ be the corresponding discriminative past price set. For every $k \in \mathcal{K}^{(i)}$, let $\bar{u}_k^{(i)}$ be the average daily usage observed at price ω_k so far,

$$\bar{u}_k^{(i)} := \frac{u_k^{(i)}}{N_k^{(i)}}, \quad (3.7)$$

where

$$u_k^{(i)} := \sum_{0 \leq m \leq i-1: p^{(m)} = \omega_k} u^{(m)}, \quad N_k^{(i)} := \sum_{0 \leq m \leq i-1: p^{(m)} = \omega_k} N^{(m)}, \quad (3.8)$$

i.e., if the same price has been used multiple times, then we take the average usage observed at that price over all the pricing periods.

We will make use of $\Gamma_k^{(i)}$, for all $k \in \mathcal{K}^{(i)}$, which is the lower confidence bound of the $100(1 - \alpha)\%$ -confidence interval for the true average daily usage at price ω_k , and which

we calculate using the normal approximation to the binomial distribution, as

$$\Gamma_k^{(i)} := \max \left\{ \bar{u}_k^{(i)} - z_{\alpha/2} \cdot \sqrt{\frac{\bar{u}_k^{(i)} (1 - \bar{u}_k^{(i)})}{N_k^{(i)}}}; 0 \right\}. \quad (3.9)$$

Note that usually it is agreed that the normal approximation is a good approximation to the binomial distribution when $N_k^{(i)} \bar{u}_k^{(i)} \geq 5$ and $N_k^{(i)} (1 - \bar{u}_k^{(i)}) \geq 5$. Finally, note that the extreme case $\alpha = 1$ corresponds to the *liberal toll setter* with $\Gamma_k^{(i)} = \bar{u}_k^{(i)}$, whereas the extreme case $\alpha = 0$ corresponds to the *archconservative toll setter* with $\Gamma_k^{(i)} = 0$.

3.2.1 Estimation

To estimate the distribution function of the public road cost we adopt an approach analogous to that described in [subsection 3.1](#), assuming that nature chooses a distribution which minimizes the expected cost of the customer. However, instead of fitting a distribution to a given mean and variability bound, we use a more conservative approach, in which nature finds a distribution using lower confidence bounds by solving the following linear program

$$\min x_1^{(i)} \omega_1 + \sum_{k \in \mathcal{K}^{(i)}} x_k^{(i)} \omega_k \quad (3.10)$$

$$\text{s.t. } x_1^{(i)} + \sum_{k \in \mathcal{K}^{(i)}} x_k^{(i)} = 1 \quad (3.11)$$

$$\sum_{k \in \mathcal{K}^{(i)}: \omega_k \geq \omega_\ell} x_k^{(i)} \geq \Gamma_\ell^{(i)} \quad \forall \ell \in \mathcal{K}^{(i)} \quad (3.12)$$

$$0 \leq x_k^{(i)} \leq 1 \quad \forall k \in \mathcal{K}^{(i)} \cup \{1\} \quad (3.13)$$

Note that $x_k^{(i)}$ corresponds to the probability mass put on the price indexed by $k \in \mathcal{K}^{(i)}$. Note also that (3.10)–(3.13) is written by defining variables only on prices in $\Omega^{(i)}$. Finally, note that (3.12) actually allows to calculate $x_k^{(i)}$ recursively for k from largest to lowest as long as it is satisfied as equality. Using that observation and the monotonicity of allowed prices, it can be shown that the procedure given in [Algorithm 1](#) solves this problem optimally. Once the estimated distribution of cost C is obtained, we calculate its mean and

Algorithm 1 Estimation

INPUT: Lower confidence bounds for the average daily usage under all used prices:

$\Gamma_k^{(i)} \forall k \in \mathcal{K}^{(i)}$

Initialize $x_k^{(i)} := 0$ for all $k \in \mathcal{K}^{(i)}$

Set $K := \max \mathcal{K}^{(i)}$

Set $x_K^{(i)} := \Gamma_K^{(i)}$

Set $\ell^+ := K$

for $\ell := K - 1$ **downto** 2 **do**

if $\ell \in \mathcal{K}^{(i)}$ and $\Gamma_\ell^{(i)} - \Gamma_{\ell^+}^{(i)} \geq 0$ **then**

 Set $x_\ell^{(i)} := \Gamma_\ell^{(i)} - \Gamma_{\ell^+}^{(i)}$

 Set $\ell^+ := \ell$

end if

end for

Set $x_1^{(i)} := 1 - \sum_{k \in \mathcal{K}^{(i)}} x_k^{(i)}$

OUTPUT: Probability mass $x_k^{(i)} \forall k \in \mathcal{K}^{(i)}$ and $x_1^{(i)}$

variance

$$\mu^{(i)} := x_1^{(i)} \omega_1 + \sum_{k \in \mathcal{K}^{(i)}} x_k^{(i)} \omega_k, \quad (3.14)$$

$$\text{Var}^{(i)} := x_1^{(i)} (\omega_1)^2 + \sum_{k \in \mathcal{K}^{(i)}} x_k^{(i)} (\omega_k)^2 - (\mu^{(i)})^2. \quad (3.15)$$

The interpretation of the key constraint (3.12) is as follows. When the toll price is set to ω_ℓ , then the proportion of days when public road cost was at least as high as ω_ℓ is equal to the average daily usage $\bar{u}_\ell^{(i)}$. Thus, the average daily usage $\bar{u}_\ell^{(i)}$ gives us an estimate of the cumulative probability of the cost being ω_ℓ or higher. In line with the nature's objective of finding a distribution that minimizes the expected cost of the customer, we put all the probability mass of the cost being between ω_ℓ and ω_{ℓ^+} (where $\ell^+ := \min\{k \in \mathcal{K}^{(i)} : \omega_k > \omega_\ell\}$) onto ω_ℓ . Moreover, in order to account for sampling variation, instead of using $\bar{u}_\ell^{(i)}$ we conservatively employ the lower confidence bound $\Gamma_\ell^{(i)}$ defined in (3.9).

Note that for archconservative toll setter (who considers $\alpha = 0$) we get as optimal solution the degenerate distribution $x_1^{(i)} = 1$.

3.2.2 Pricing

To set the price $p^{(i)}$ for the i -th pricing period, where $i \geq 1$, out of the allowed ones in $\underline{\Omega}^{(i)} := \{\omega \in \Omega : \omega \leq Q^{(i)}\}$, we solve the following bilevel problem, a slight modification

of that described in [subsection 3.1](#), using the mean $\mu^{(i)}$ and the variance $\text{Var}^{(i)}$ calculated in the estimation step,

$$\max_{p \in \underline{\Omega}^{(i)}} \sum_{j: \omega_j \geq p} x_j p \quad (3.16)$$

$$\min \sum_{j: \omega_j \geq p} x_j p + \sum_{j: \omega_j < p} x_j \omega_j \quad (3.17)$$

$$\text{s.t. } \sum_{j \in \mathcal{J}} x_j = 1 \quad (3.18)$$

$$\sum_{j \in \mathcal{J}} x_j \omega_j = \mu^{(i)} \quad (3.19)$$

$$\sum_{j \in \mathcal{J}} x_j \omega_j^2 \leq \text{Var}^{(i)} + (\mu^{(i)})^2 \quad (3.20)$$

$$0 \leq x_j \leq 1 \quad \forall j \in \mathcal{J} \quad (3.21)$$

Since κ is unknown, we have modified the problem replacing the original constraint

$$\sum_{j \in \mathcal{J}} x_j \omega_j^2 \leq \kappa \mu^{(i)} + (\mu^{(i)})^2 \quad (3.22)$$

with (3.20). Note that the new constraint strengthens the original formulation, because $\text{Var}^{(i)} \leq \kappa \mu^{(i)}$.

Proposition 3.1. *The solution to (3.17)–(3.21) is always a distribution with support having at most three points.*

Proof Any basic feasible solution to the linear program (3.17)–(3.21) has at least $|\mathcal{J}| - 3$ zeros, which means that there are at most three non-zero values of x_j . \square

In [Algorithm 2](#) we present a heuristic pricing procedure, which computes the optimal two-point distribution in the lower level of (3.16)–(3.21) instead of solving the linear program (3.17)–(3.21), which would yield a three-point distribution. The motivation to do this is that we have found in our computational experiments that (i) two of the three support points are usually very close in value, (ii) [Algorithm 2](#) is much faster compared to solving many linear programs, and (iii) the performance is not notably different.

In [Algorithm 2](#), $U(p)$ represents the upper-level objective value at toll price p , while $L^*(p)$ is the optimal lower-level objective value at toll price p . Note also that ℓ and u are the two price points with non-zero probability mass (lower and upper, respectively), and

Algorithm 2 Two-Point Pricing

INPUT: $\mu^{(i)}$, $\text{Var}^{(i)}$, maximum allowed price $Q^{(i)}$
Set $\underline{\Omega}^{(i)} := \{\omega \in \Omega : \omega \leq Q^{(i)}\}$; $\underline{\Omega}^u := \{\omega \in \underline{\Omega}^{(i)} : \omega > \mu^{(i)}\}$; $\underline{\Omega}^\ell := \{\omega \in \underline{\Omega}^{(i)} : \omega < \mu^{(i)}\}$
for $p \in \underline{\Omega}^{(i)}$ **do**
 Initialize $L^*(p) := Q^{(i)}$, $U(p) := Q^{(i)}$
 for $\ell \in \underline{\Omega}^\ell$ **do**
 for $u \in \underline{\Omega}^u$ **do**
 Set $x := \frac{u - \mu^{(i)}}{u - \ell}$
 if $x\ell^2 + (1 - x)u^2 \leq \text{Var}^{(i)} + (\mu^{(i)})^2$ **then**
 if $L^*(p) > x\ell + (1 - x)p$ **then**
 Set $L^*(p) := x\ell + (1 - x)p$; $U(p) := (1 - x)p$
 end if
 end if
 end for
 end for
end for
OUTPUT: price for the i -th period: $p^{(i)} := \min \arg \max_{p \in \underline{\Omega}^{(i)}} U(p)$

x is the probability mass at price ℓ .

3.2.3 Approximate numerical two-point solution

When the set of the prices $\underline{\Omega}^{(i)}$ is too large, finding the revenue maximizing price by finding a two point distribution for each price as in [Algorithm 2](#) can be too time consuming. Although a closed form formula for optimal price is not possible to derive, a numerical approximation, much less time consuming to calculate, can be obtained as follows. For the rest of this section we drop the pricing period index for convenience.

Let the best response of nature when toll is set at p be a two point distribution with support (ℓ, u) and probabilities $(x, 1 - x)$ with specified mean μ and variance $\sigma^2 = \text{Var}$. Let us denote this distribution as B2D. Given ℓ, u and μ using [Equation 3.19](#) and [Equation 3.20](#) we can write

$$x = \frac{u - \mu}{u - \ell} \quad (3.23)$$

Similarly, we have

$$\ell = \mu - \sqrt{\frac{1 - x}{x}} \sigma \quad (3.24)$$

$$u = \mu + \sqrt{\frac{x}{1 - x}} \sigma \quad (3.25)$$

Algorithm 3 DJA Algorithm

INPUT: $p^{(0)}, \tau^{\text{DJA}}, \delta, \epsilon, \text{UU}, \text{OU}$
 Announce $p^{(0)}$ and wait until the end of the pricing period to observe $u^{(0)}$
 Set $\text{UUreturn} = \text{TRUE}$
for $i \in \mathcal{I}$ with $i \geq 1$ **do**
 if $\sum_{0 \leq m \leq i-1} N^{(m)} \geq \tau^{\text{DJA}}$ **then**
 $\text{UU} = 0$ and $\text{OU} = 1$
 end if
 For k such that $p^{(i-1)} = \omega_k$, calculate $\bar{u}_k^{(i)}$ using (3.7) and set $\bar{u}^{(i)} := \bar{u}_k^{(i)}$
 while $\bar{u}^{(i)} \leq \text{UU}$ **and** $p^{(i-1)} > \omega_2$ (Forced exploration) **do**
 if $\text{UUreturn} = \text{FALSE}$ **then**
 $\text{UU} = \text{UU} - \delta$
 end if
 Set $p^{(i)} := \min\{\omega \in \Omega \setminus \{q\} : \omega \geq p^{(i-1)} - \frac{\sum_{0 \leq m \leq i-1} N^{(m)}}{\tau^{\text{DJA}}}(p^{(i-1)} - \omega_2)\}$
 Announce $p^{(i)}$ and wait until the end of the pricing period to observe $u^{(i)}$
 Set $i := i + 1$
 For k such that $p^{(i-1)} = \omega_k$, calculate $\bar{u}_k^{(i)}$ using (3.7) and set $\bar{u}^{(i)} := \bar{u}_k^{(i)}$
 end while
 For k such that $p^{(i-1)} = \omega_k$, calculate $\bar{u}_k^{(i)}$ using (3.7) and set $\bar{u}^{(i)} := \bar{u}_k^{(i)}$
 if $\bar{u}^{(i)} \geq \text{OU}$ **and** $p^{(i-1)} < \omega_{J-1}$ **then**
 if $\text{OUreturn} = \text{FALSE}$ **then**
 $\text{OU} = \text{OU} + \delta$
 end if
 Set $Q^{(i)}$ using (2.1)
 Set $p^{(i)} := \min\{\omega \in \Omega \setminus \{Q\} : \omega \geq p^{(i-1)} + \min\{50\%(\omega_{J-1} - p^{(i-1)}); Q^i - p^{(i-1)}\}\}$
 else
 Set $\text{UUreturn} = \text{FALSE}; \text{OUreturn} = \text{FALSE}$
 Set $\Gamma_k^{(i)}$ for all $k \in \mathcal{K}^{(i)}$ using (3.9)
 Run Algorithm 1 (Estimation)
 Set $\mu^{(i)}$ using (3.14), $\text{Var}^{(i)}$ using (3.15), and $Q^{(i)}$ using (2.1)
 Run Algorithm 2 (Pricing)
 if $|\mu^{(i)} - \mu^{(i-1)}| < \epsilon$ **then**
 Set $p^{(i)} := \min\{p^{(i)}, p^{(i-1)}\}$
 end if
 end if
 Announce $p^{(i)}$ and wait until the end of the pricing period to observe $u^{(i)}$
end for

Since nature has chosen B2D when toll is chosen at p , x is chosen to minimize

$$L(x) = x\ell(x) + p(1 - x)$$

Differentiating with respect to x and putting $L'(x) = 0$ and rearranging for p we get

$$p(\mu, \sigma, x) = \mu - \frac{1 - 2x}{2\sqrt{x(1 - x)}}\sigma. \quad (3.26)$$

Recall that the toll-setter's objective is

$$\max_x p(\mu, \sigma, x)(1 - x).$$

Setting the derivative w.r.t to x equal to zero we get

$$\frac{4x^3 - 6x^2 + x + 1}{4[x(1 - x)]^{\frac{3}{2}}} = \frac{\mu}{\sigma}.$$

This results in a quartic equation in x which is not amenable for a closed-form solution. However, a numerical approximation to a desired precision can easily be calculated.

3.2.4 Initialization and Forced Exploration

It is desirable to set an initial price $p^{(0)}$ so that it does not fall in the tails of the price set Ω . The main reason is that if starting at a too high or a too low initial price, one may be unable to sufficiently explore the price set and learn enough about the unknown distribution F . Further, the initial price should be such that we observe at least 5 and at most $N^{(0)} - 5$ days of usage in the initial period, so that the normal approximation used in our algorithm is justified. Finally, one can get trapped at very low prices, especially due to the price cap, if starting price is chosen too low. As we show in [Section 5](#) a judicious choice of starting price will lead to a very good algorithm performance. We suggest to take $p^{(0)}$ as the median of the price set Ω .

Similar situations may, however, also happen in subsequent periods. In order to avoid that and thus to enable sufficient exploration of the price set in order to collect more information, we introduce parameters for over-usage (OU) and under-usage (UU), and override the estimation and pricing steps by forcedly setting the price as described next.

The forced exploration phase lasts as long as we observe the average daily usage lower

than UU, but not longer than τ^{DJA} days, where we suggest to set $\tau^{\text{DJA}} := 20\%T$. During this phase, the price is decreased proportionally to the number of days passed, but relatively to the previous price, so that the prices at the decision epochs are approximately sigmoid-shaped, going from $p^{(0)}$ to the second lowest allowed price ω_2 . (We avoid setting the price to the lowest allowed price q to account for the assumption that the toll road is preferred when the cost of the alternative road is equal to the toll road price, i.e. the observed usage would be $N^{(i)}$ if we set $p^{(i)} = q$.) The idea is to increase the exploration depth if we observe low usage despite price decrease by decreasing the price at a faster pace for a while, slowing down when approaching ω_2 . This exploration strategy depends on the parameter UU which may be selected depending on price range, that is, the larger the $\frac{Q}{q}$, the higher the UU. We suggest to set $UU = 0.25$. Ideally, we would want this parameter to be adjusted in a data-driven fashion with the initial choice depending on price range.

Similarly, the parameter OU is set to increase the price if usage is very high with rate of increase depending on the current price, aiming at the convergence to the second highest allowed price ω_{J-1} , as long as the price cap allows. We avoid setting the price to the largest allowed price Q because we want to avoid the possibility of minimum usage.

Finally, to account for the fact that optimal price may lie in the UU and OU zones (i.e., at optimal price, say ω_k , we have $\bar{u}_k \leq UU(\geq OU)$), for example if algorithm sets price in these zone of prices we still stay in the forced exploration phase but adjust the UU and OU limits by a pre-specified value δ . Therefore, if it happens that the optimal price is within UU or OU zone, due to adjusting the UU/OU limits (decreasing UU and increasing OU) the algorithm will eventually set price closer to this optimal price. Note that even without this adjustment algorithm will achieve this after getting past the forced exploration phase. The reason for this adjustment is to make this happen quicker.

3.3 Convergence

In infinite horizon problems (such as those usually considered in machine learning), an algorithm is considered good if it converges to the theoretical optimal price, and if the frequency of using suboptimal prices reduces quickly. The frequency of trying different prices cannot be reduced to zero, because otherwise it would not be possible to achieve complete learning, i.e., collect enough evidence to be confident about which price is optimal. However, given the uncertainty about the distribution F , designing an algorithm which achieves this is difficult, if not impossible, without making assumptions on F ,

which is not suitable for our non-parametric approach.

However, in finite horizon case it is more desirable that an algorithm settles at a price towards the end of the horizon, and in some situations it may be desirable to settle quickly. For instance, from the road pricing point of view, changing price often means extra costs and is hence not desirable. We now show that the estimated means converge and therefore, the DJA Algorithm settles at a price.

Proposition 3.2. *Let $p^{(i)}$ be the first price set after the forced exploration phase of the DJA algorithm, i.e., once the usage observed is greater than UU. Then, the sequence of means $\mu^{(i)}, \mu^{(i+1)}, \dots, \mu^{(n)}$ is increasing, where n is first period in which $|\mu^{(n)} - \mu^{(n-1)}| < \epsilon$.*

Proof Consider any $i^* \in \{i, i+1, \dots, n-1\}$. If $p^{(i^*+1)} > p^{(i^*)}$, then the probability mass is shifted from $p^{(i^*)}$ to $p^{(i^*+1)}$ in the estimation step because only one of the lower confidence bounds is changed, and therefore, the mean increases by $x_k^{(i^*)}(p^{(i^*+1)} - p^{(i^*)})$. If $p^{(i^*+1)} < p^{(i^*)}$, then, using a similar reasoning, the probability mass is shifted from a price strictly lower than $p^{(i^*+1)}$. If $p^{(i^*+1)} = p^{(i^*)}$, then $\Gamma_{k+1}^{(i^*)} > \Gamma_k^{(i^*)}$, and therefore the mean increases because lower prices get less probability mass. \square

Corollary 3.1. *Let $\{p^{(i)}\}_{i \in \mathcal{I}}$ be the sequence of prices set by the DJA Algorithm. Then, the sequence $\{p^{(i)}\}_{i \in \mathcal{I}}$ converges to the price at which it settles at least at linear rate.*

The proof is straightforward and it is omitted.

Of course, the rate of convergence to the price at which DJA algorithm settles depends on the actual distribution and the parameters of the algorithm. In our experiments we found that price converges within first 15 periods for a range of distributions when the number periods is equal to 100. The convergence can happen during the forced exploration phase or during the main phase. If the optimal price lies in UU or OU zone, then the algorithm can have an oscillating price behavior where it pushes the price to UU or OU zones while the forced exploration phase pushes it outside of these zones. However, price oscillations will be settled by adjusting the UU and OU limits by δ , reaching the convergence faster.

4 Numerical Experiments

4.1 Performance Metric

To evaluate the performance of an algorithm, we use *relative regret*, which is commonly used in the pricing and learning literature, and is calculated as follows:

$$\text{relative regret} = \frac{\max\{\text{optimal revenue} - \text{algorithm revenue}, 0\}}{\text{optimal revenue}} \cdot 100\%$$

where optimal revenue is the revenue obtained using the static optimal price i.e., the price in Ω at which maximum revenue is obtained over all I periods. Note that the best theoretically achievable relative regret is 0%, however there is a hard-to-evaluate practical lower limit due to the imperfect learning within a finite horizon, while the worst algorithm (which always sets the price too high and thus observes zero revenue) achieves 100%. In the numerical experiments reported in this paper we have never observed relative regret lower than 3%.

4.2 Benchmark Algorithm

In this subsection we give an adaptation of Algorithm 1 of [6], which we will refer to as the *Besbes-Zeevi (BZ) Algorithm*. The BZ algorithm is based on the state-of-the-art idea from the field of design of experiments, in which acting is split into two phases: (i) *Learning phase* (i.e., exploration of allowable prices), in which a predetermined number of prices is chosen for experimentation along with predetermined holding times, and (ii) *Earning phase* (i.e., exploitation of the best price), in which the price identified as best (in terms of revenue) out of those chosen in the learning phase is constantly applied. A formal description of the BZ algorithm is given in [Algorithm 4](#).

The input parameter τ is the duration (in number of days) of the learning phase, and the input parameter η is the number of prices to experiment with during the learning phase. Note that the description in [Algorithm 4](#) implicitly assumes that the set of allowable prices is regularly spaced and τ and η are chosen so that $\omega_{\bar{\eta}\Delta} \in \Omega$ for all $\bar{\eta} \in \{1, 2, \dots, \eta\}$, with $\omega_{\eta\Delta} = \omega_J$ and $\omega_{\Delta} = \omega_1$. For the more general sets Ω , the Learning phase of the BZ algorithm would have to be rewritten similarly to the Forced Exploration

Algorithm 4 BZ Algorithm

```
INPUT:  $\tau, \eta$ 
Set  $i := 0, \Delta = \frac{\tau}{\eta}, rev^{best} = 0, p^{best} = Q - 1$ 
for  $\bar{\eta} = \eta : 1$  do
  Set count =  $\Delta; i = 0$ 
   $u_{\bar{\eta}} = 0$ 
  while count > 0 do
    Set  $p^{(i)} := \omega_{\bar{\eta}\Delta}$ 
    Announce  $p^{(i)}$  and wait until the end of the pricing period to observe  $u^{(i)}$ 
    Set  $u_{\bar{\eta}\Delta} = u_{\bar{\eta}\Delta} + u^{(i)}$ 
    Set  $i := i + 1, \text{count} = \text{count} - 1$ 
  end while
  Set  $\bar{u}_{\bar{\eta}\Delta} = \frac{u_{\bar{\eta}\Delta}}{\sum_{j=i-\Delta+1}^i N^{(j)}}$ 
  if  $rev^{best} \leq \omega_{\bar{\eta}\Delta} \bar{u}_{\bar{\eta}}$  then
    Set  $p^{best} = \omega_{\bar{\eta}\Delta}$ 
  end if
end for
Set  $\underline{i} := i + 1$ 
for  $i \in \mathcal{I}$  with  $i \geq \underline{i}$  do
  Set  $Q^{(i)}$  using (2.1)
  Set  $p^{(i)} := \min \{p^{best}, Q^{(i)}\}$ 
end for
```

phase of DJA, with the price setting using the formula

$$p^{(i)} := \min \{ \omega \in \Omega \setminus \{q\} : \omega \geq p^{(0)} - \frac{\sum_{0 \leq m \leq i-1} N^{(m)}}{\tau} (p^{(0)} - \omega_2) \} \quad (4.1)$$

Thus, one can see DJA as an approximate extension of BZ, because if we set $UU := 1$ and set the new price $p^{(i)}$ relatively to $p^{(0)}$ rather than to $p^{(i-1)}$, then the DJA's Forced Exploration phase essentially behaves like BZ's Learning phase.

However, the main phases are fundamentally different, DJA doing a dynamic pricing while BZ doing a static pricing, and it is not possible to easily compare their price choices.

Note that BZ starts with Q as the initial price, and we use $Q - 1$ in DJA in order to obtain a fair comparison with BZ algorithm.

4.3 Simulated Data

In our first set of experiments we have sampled the cost of an alternative product (public road) from six different distributions, which are often used to model travel time in transportation and/or to model demand in pricing and revenue management literature. These distributions are listed in Table 1, where the distribution called Minima is generated by taking the minimum of the five distributions listed in Table 2. This is intended to capture a more realistic case of several alternative products (e.g., multiple public roads). The distributions on these alternative products can be different. We generate the Minima sample by taking the minimum of each day’s cost sampled from the five different distributions.

4.3.1 Data Generation

For each distribution we have created 100 instances with each instance having $T = 4,800$ days (approximately 13 years). Each instance from a specified distribution have different parameters randomly selected from an interval, these intervals for each distribution are given in Table 1. For Minima, samples are generated using parameters randomly chosen from intervals given in Table 2.

We consider three experimental setups with constant pricing-period durations $N^{(i)} = 50$ (approximately monthly), $N^{(i)} = 100$ (approximately quarterly), and $N^{(i)} = 400$ (approximately yearly); which correspond to 96, 48 and 12 pricing periods respectively. We will refer to these three setups as Setup50, Setup100, and Setup400 respectively when we present our results. The three experimental setups reflect different price revision opportunities. The rationale for choosing the setups in this way also comes from the toll-pricing motivation. In Figure 1 and Figure 2, we show the shape of revenue and usage curves versus the price respectively for a sample of 5 instances out of 100 used from each distribution type. We also depict the converged prices of both algorithms on these curves. Figure 3 gives the price paths of the same five instances.

4.3.2 DJA vs BZ

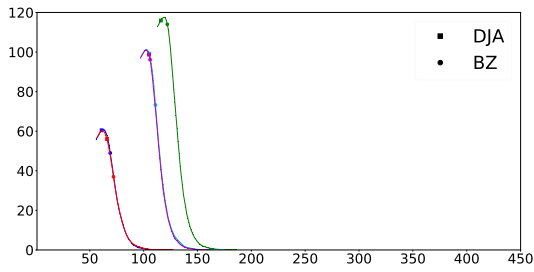
To evaluate BZ we have considered several combinations of τ (learning time) and η (number of prices to explore), shown in Table 3, and report the lowest average regret obtained over all these parameter combinations. We compare this (optimized) BZ to the (non-optimized) DJA with the following default parameters: $UU = 0.25$, $\tau^{\text{DJA}} = 20$, $p^{(0)} = \omega_J - 1$, and $(1 - \alpha) = 0.95$. Table 4 gives the average relative regrets along with

Distribution/Data	Parameters
Gumbel	$[20 : 50, 2]$
Gamma	$[3 : 5, \frac{1}{3} : \frac{1}{5}]$
Uniform	$[30 : 80, 120 : 170]$
Normal	$[90 : 110, 10 : 30]$
Minima	*
Log-Normal	$[\frac{1}{5} : \frac{1}{2}, \frac{1}{5} : \frac{1}{2}]$

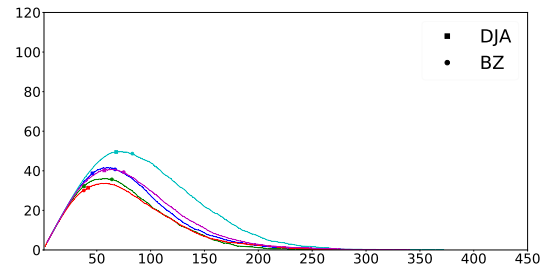
Table 1: Parameter Ranges

Distribution/Data	Parameters
Gamma	$[3 : 5, \frac{1}{3} : \frac{1}{5}]$
Uniform	$[50 : 99, 100 : 150]$
Normal	$[90 : 110, 10 : 30]$
Log-Normal	$[\frac{1}{5} : \frac{1}{2}, \frac{1}{5} : \frac{1}{2}]$
Beta	$[2 : 5, 2 : 5]$

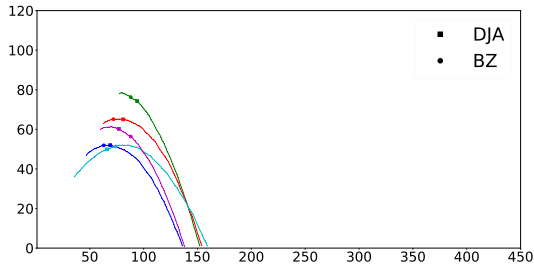
Table 2: Data for minima distribution



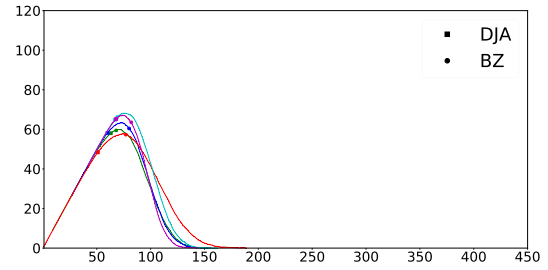
(a) Gumbel



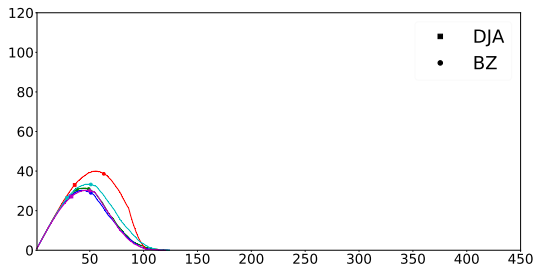
(b) Gamma



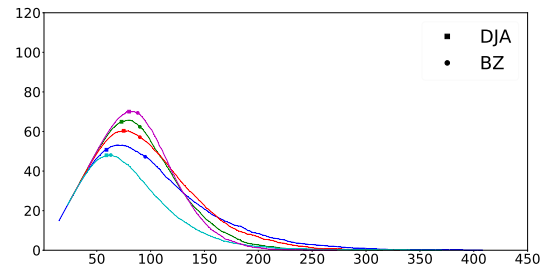
(c) Uniform



(d) Normal



(e) Minima



(f) Lognormal

Figure 1: Revenue vs Price plots for 5 instances of data generated from different distributions

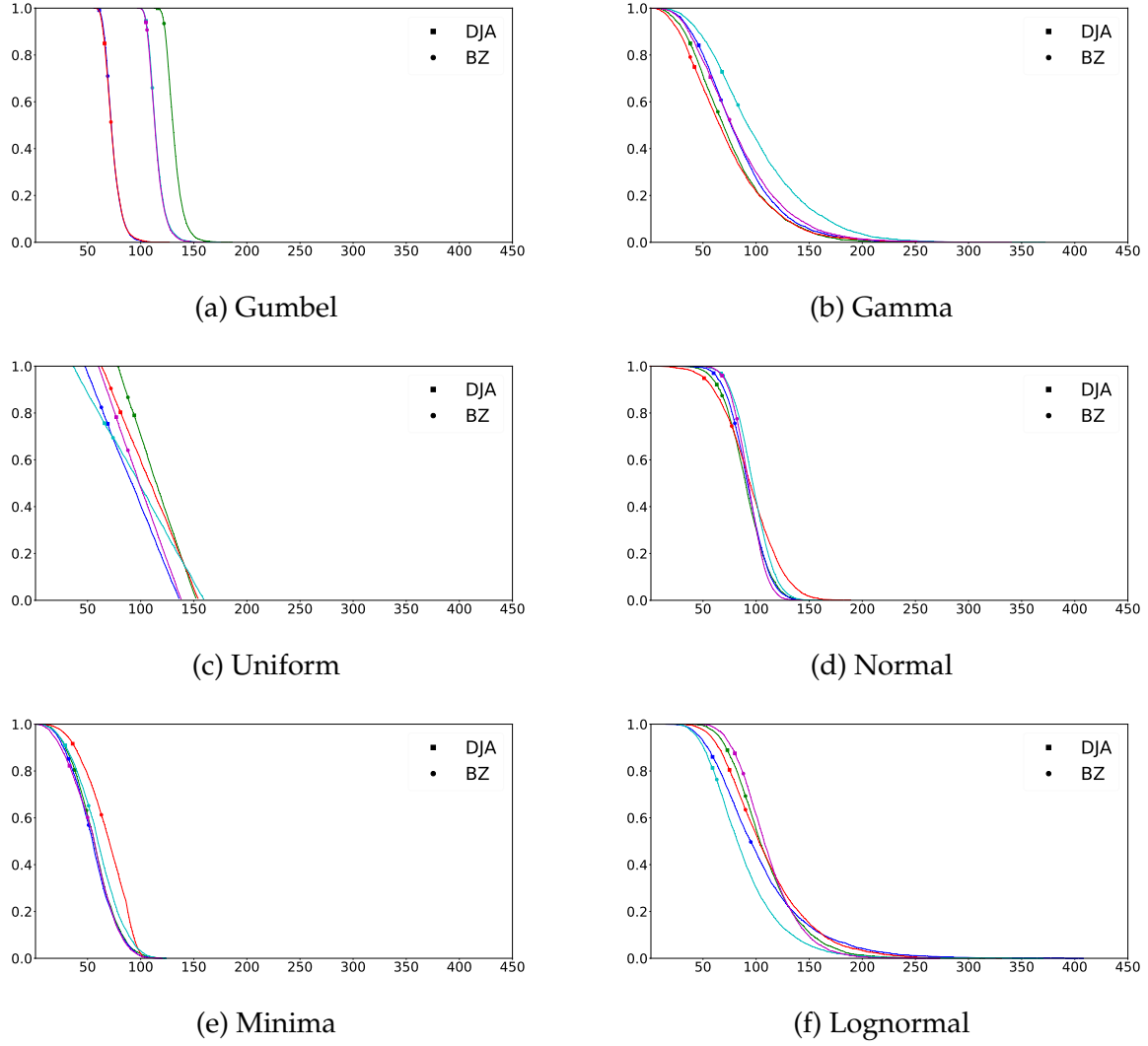
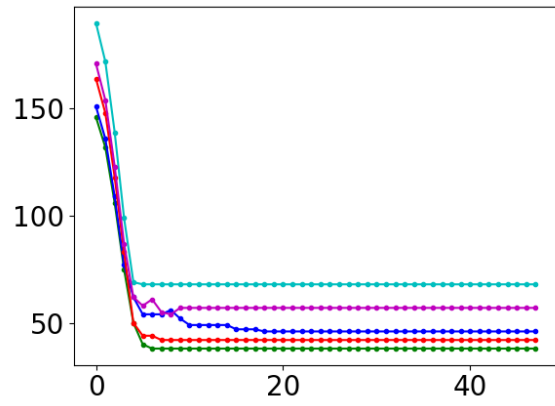


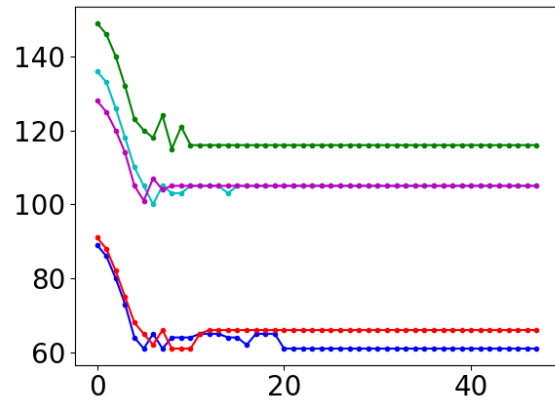
Figure 2: Usage vs Price plots for 5 instances of data generated from different distributions

τ	10	12	14	15	16	18	20
η	2, 5, 10	2, 3, 4, 6, 12	2, 7, 14	3, 5, 15	2, 4, 8, 16	2, 3, 6, 9, 18	2, 4, 5, 10, 20

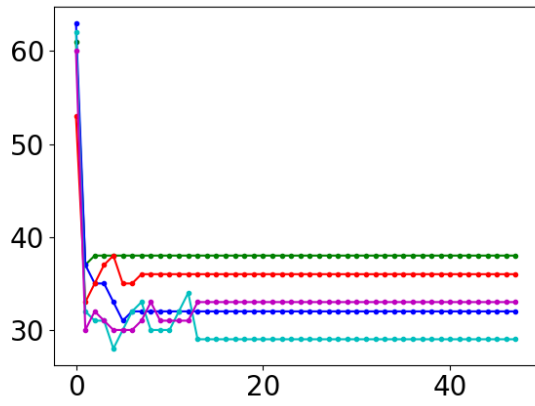
Table 3: Parameters of the BZ algorithm.



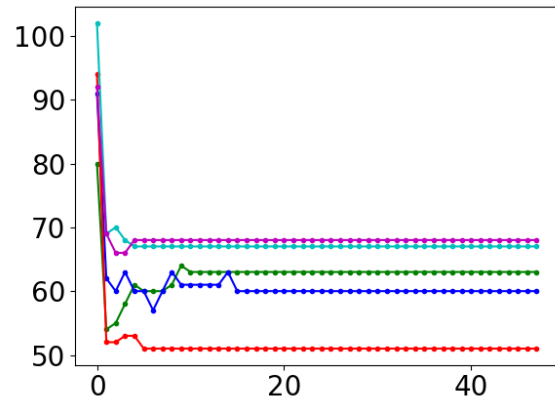
(a) Gamma



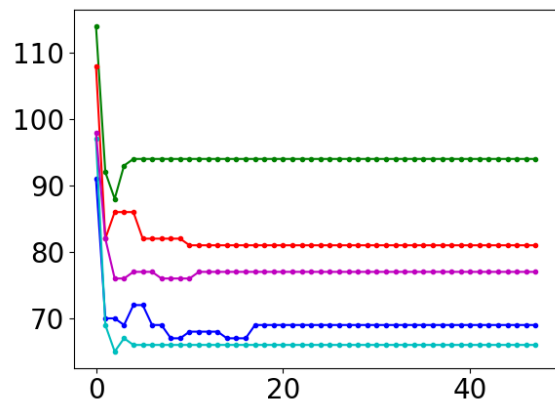
(b) Gumbel



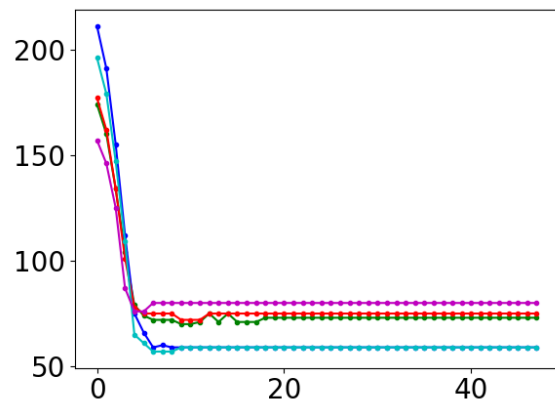
(c) Minima



(d) Normal



(e) Uniform



(f) Lognormal

Figure 3: Price paths for different distributions in Setup100 in presence of price caps

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA	37.76 (4.10)	18.50 (1.55)	26.99 (20.71)	26.33 (20.31)	12.46 (3.27)	27.73 (2.70)
BZ	34.75 (6.21)	63.58 (9.03)	21.65 (4.19)	24.19 (3.85)	15.36 (3.17)	45.23 (13.51)

(a) Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA	13.49 (2.45)	12.28 (1.76)	13.13 (3.56)	12.00 (3.42)	8.55 (3.61)	11.76 (1.83)
BZ	18.43 (3.02)	28.28 (10.49)	12.74 (2.74)	12.82 (3.02)	9.83 (3.28)	17.88 (2.73)

(b) Setup100

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA	12.40 (2.99)	10.54 (2.31)	13.27 (3.84)	10.39 (3.63)	7.52 (3.17)	9.72 (1.98)
BZ	11.31 (3.56)	21.36 (11.28)	8.90 (3.64)	8.35 (2.44)	7.86 (4.08)	11.13 (2.36)

(c) Setup50

Table 4: Comparison DJA vs BZ

the standard deviations for both algorithms. As expected, the average regrets are highest in Setup400 due to very small opportunity to learn. Also, the variability is higher in this setup compared to the other two.

In all the cases the performance of (non-optimized) DJA is competitive to (optimized) BZ. In cases of Gumbel and Uniform DJA always outperforms BZ, while in other cases DJA without even parameter optimization comes close to BZ. The high standard deviations of DJA in Setup400 are due to the fact that the problem does not offer sufficient time to find the near optimal price.

As will be seen in subsequent sections, a bit of parameter tuning and judicious setting improves DJA’s performance considerably. We will defer the discussion on performance improvement to later sections and first consider incorporating the price caps into the pricing algorithms.

4.3.3 Price Caps

We assess both algorithms in the case when there is a limit on the price increase allowed between consecutive pricing periods. Observe that the price caps are taken into account in the pricing step of DJA. More specifically, only prices which can be set are considered when solving the bi-level program. Similarly, price caps are taken into account after the learning phase finishes in BZ.

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA*	15.45 (9.54)	15.82 (2.05)	11.11 (2.98)	7.23 (3.28)	3.41 (2.58)	14.53 (5.85)
DJA	39.30 (3.95)	18.50 (1.57)	27.37 (12.22)	23.50 (10.64)	12.49 (3.26)	28.08 (2.85)
BZ	34.75 (6.21)	63.58 (9.03)	22.67 (4.19)	23.39 (3.3)	15.24 (3.18)	45.23 (13.51)

(a) Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA*	6.04 (1.34)	10.76 (4.07)	11.43 (3.69)	5.59 (3.38)	3.17 (2.18)	6.45 (1.61)
DJA	14.82 (3.04)	11.92 (1.48)	15.25 (3.86)	12.55 (3.66)	8.41 (3.58)	11.86 (1.93)
BZ	18.43 (2.51)	28.28 (10.49)	14.49 (2.90)	14.98 (2.2)	9.81 (3.03)	17.78 (2.43)

(b) Setup100

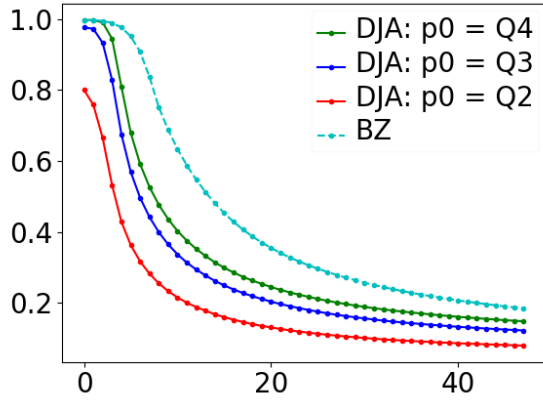
	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
DJA*	5.16 (1.40)	9.68 (3.96)	11.15 (4.01)	4.75 (3.55)	3.55 (2.21)	5.18 (1.54)
DJA	13.80 (3.40)	9.86 (2.19)	15.14 (4.45)	11.91 (4.43)	7.23 (3.21)	9.97 (2.11)
BZ	11.28 (3.29)	21.36 (11.28)	9.46 (3.27)	9.53 (1.54)	7.79 (3.9)	11.12 (2.31)

(c) Setup50

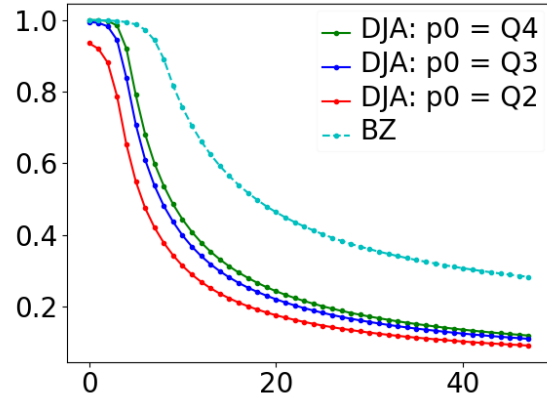
Table 5: Comparison DJA vs BZ with price cap of 5%

We compare the performance of both algorithms on same data in the presence of cap constraint. The cap value is set at 5%. [Table 5](#) gives the summary of results of outputs of algorithms in the presence of cap. Note that the table includes also performance of DJA*, which will be discussed in [subsection 5.5](#).

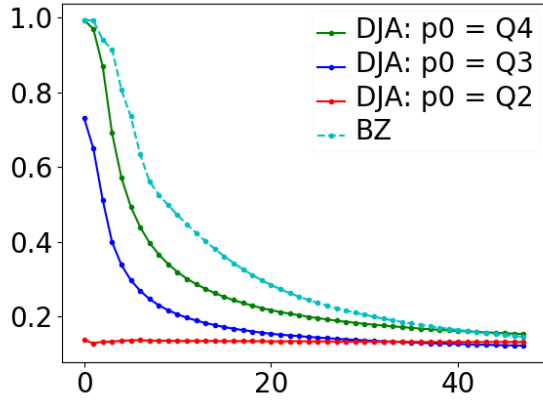
In comparison with the no-cap case, the performance of both algorithms is, approximately, similar when starting at high price, with a decrease in average regret for some distributions and the opposite in others. Note that here again we compare the non-optimized DJA with optimized BZ. As we mentioned in the last section, starting at good price can achieve much better performance for DJA which is even more important in the presence of price caps. To illustrate, we show the average regret growth curves for three different strating prices ($Q_4 > Q_3 > Q_2$ — explained in [Section 5](#)) in DJA versus BZ in [Figure 4](#) in the case of Setup100. [Figure 4](#) illustrates the tremendous improvement gained by DJA by starting at lower prices compared to the highest price (Q_4). We postpone the thorough sensitivity analysis to [Section 5](#).



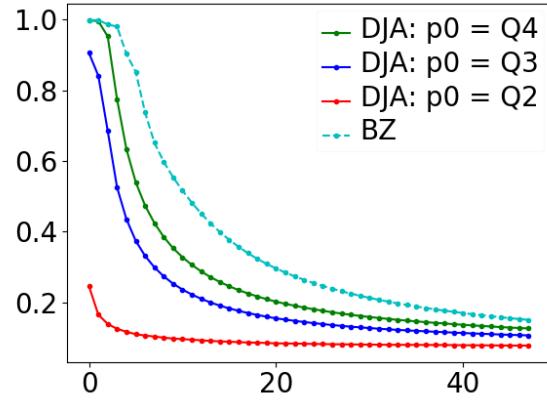
(a) Gamma



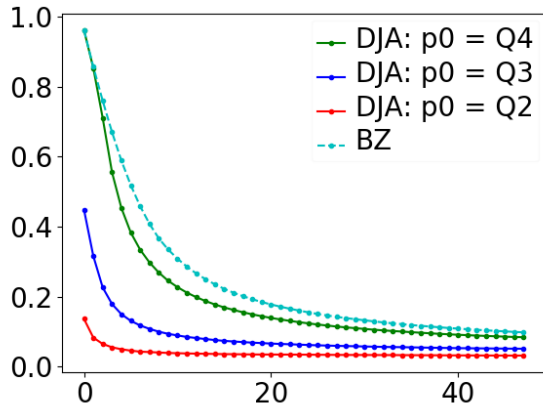
(b) Gumbel



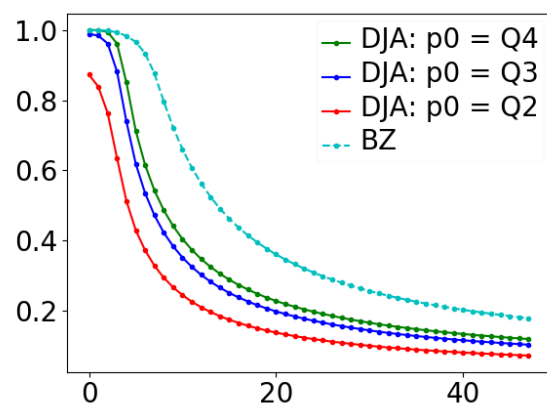
(c) Minima



(d) Normal



(e) Uniform



(f) Lognormal

Figure 4: Regret growth curves for different distributions for Setup100 in the presence of price caps

4.4 Real data

In real world the travel times and hence costs may not have any known distribution which can be analytically expressed and may have seasonality and trends over time. Moreover, a model which performs well on simulated data may not perform well on real data. To test our approach on real data which as we explain below is given as travel times on a real road network, we assume the cost is proportional to the travel times. We first explain the details of data collection and then give then details of experimental setup and results.

4.4.1 Data Collection and Cleaning

We used data provided by the City of Chicago(<https://data.cityofchicago.org>), which provides a live traffic data interface. We recorded traffic updates in a 15-minute interval over a time horizon of 24 hours for several days between March 28th 2017 to May 12th 2017. A total of 4,363 data observations were used.

Every observation contains the traffic speed for a subset of a total of 1,257 segments. There were 1,045 segments where the data was recorded at least once of the 4,363 time points with almost all segments having at least 400 records. We used linear interpolation to fill the missing records keeping in mind that data was collected over time. Segment lengths were given through longitude and latitude coordinates, and approximated using the Euclidean distance.

As segments are purely geographical objects without structure, some processing has been done to create a graph. For the sake of space we omit including the graph plot here, instead we refer the interested reader to [8]. The final graph contains 538 nodes and 1,308 arcs.

4.4.2 Experimental Setup, Results and Discussion

We assume cost of travel on each road in this network is proportional to the time of travel. Since there are no toll roads in this network which we can use for our experiments we have randomly selected 100 pairs of cities and calculated tolls by imagining a toll road between each of these pair of cities, under the assumption that cost other than tolls is zero or such costs have been adjusted in the costs of the other roads. Of 4,463 observations we used 4,400 data points with price revision points for every 100 time points.

Table 6 gives the average and standard deviations of relative regrets over all 100 instances. It is worth noting that the regrets are lower when there is cap constraint com-

Relative regret	Cap on	Cap off
Average	8.09	8.25
Stdev	4.23	3.99

Table 6: Results for Chicago data with $UU = 0.30$ and $p^{(0)} = Q2$

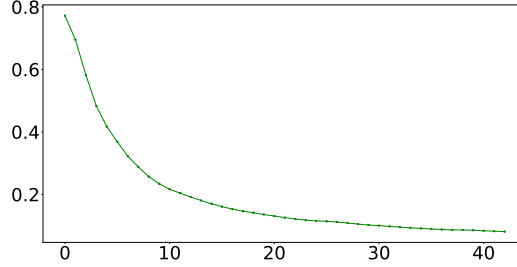


Figure 5: Chicago dataset: relative regret growth.

pared to that of when there is no constraint. Figure 5 gives the change in average relative regret with respect to time.

5 Sensitivity Analysis

In this Section we study the impact of changing the algorithm parameters on the performance of the DJA Algorithm. More specifically by choosing the price and confidence level at right levels the performance of DJA improves to a large extent and outperforms BZ in almost all cases.

5.1 Impact of Starting price

In the previous section to compare DJA-Algorithm with BZ-Algorithm the initial price was chosen high in both algorithms. However, in case of DJA starting at a price closer to optimal will give best performance. Of course one do not have the information on the optimal price, in which case it is better to avoid the prices at the tails of the support. Even if the optimal price happens to be in these tails algorithm automatically moves towards this price. To understand the impact of starting price we repeated the experiments with 4 different starting prices which are indicated in the Table 7 as $Q1$, $Q2$, $Q3$ and $Q4$, which are the lower quartile, median, upper quartile, and the maximum value of the interval

$[\omega_2, \omega_{J-1}]$. As can be noted from the [Table 7](#), starting prices have considerable impact on the performance of the algorithm. Especially, in the case of Minima distribution this impact is more compared to other cases. While in all three setups and for all distributions Q1 price seem to give lowest average regrets, except in case of Minima, this may be a risky strategy especially when there is a price cap.

When there are more periods, in Setup50 ([Table 7 \(c\)](#)), except for Minima, for all distributions algorithm gives a very good performance when starting price is chosen at Q2 and Q3. [Table 7 \(b\)](#) shows that starting at a good price can offset the difficulty of less periods. Particularly interesting are the results in Setup400, where there are less chances of price revision and hence less opportunity for optimal learning, it is noteworthy that at Q2 the regrets are much more competitive than when starting high. In all the distributions on average the optimal price seem to lie between Q2 and Q1 prices, again the exception being Minima.

For the experiments of [Table 7](#) we fixed all the parameters same as in [subsubsection 4.3.3](#) i.e., we set $(1 - \alpha)$ at 0.95, τ^{DJA} at $20\%T$, UU at 0.25.

Since Algorithm DJA uses all the history, unlike algorithms which only use data from experimentation phase, the price path is dynamically selected. As can be seen in [Table 7](#) the performance of DJA Algorithm is much better when the initial price is set at the median price.

The impact of duration of period versus the number of periods is much more visible from the [Table 7](#). More specifically, counter to the intuition in Minima case starting at Q2 in Setup400 gives best performance. This illustrates the trade-off between period duration and number of periods. With good enough period duration the estimate of usage at a price is more robust and along with large period size translates to shorter confidence intervals in estimation phase. This explains the better performance of DJA, being much better in Setup100 and Setup400. It also helps the algorithm avoid getting trapped in lower prices. This is illustrated in [Figure 6](#). The prices set by Algorithm in Setup50 are low due to conservative lower confidence bounds a result of smaller period duration. This shows the robustness of DJA when right starting price is coupled with right period duration. Note that period duration can be regarded as a decision parameter itself. On the other hand increasing the period durations at the cost of the lesser number of periods can be bad idea and can result in bad performance especially if other parameters are not chosen at right values.

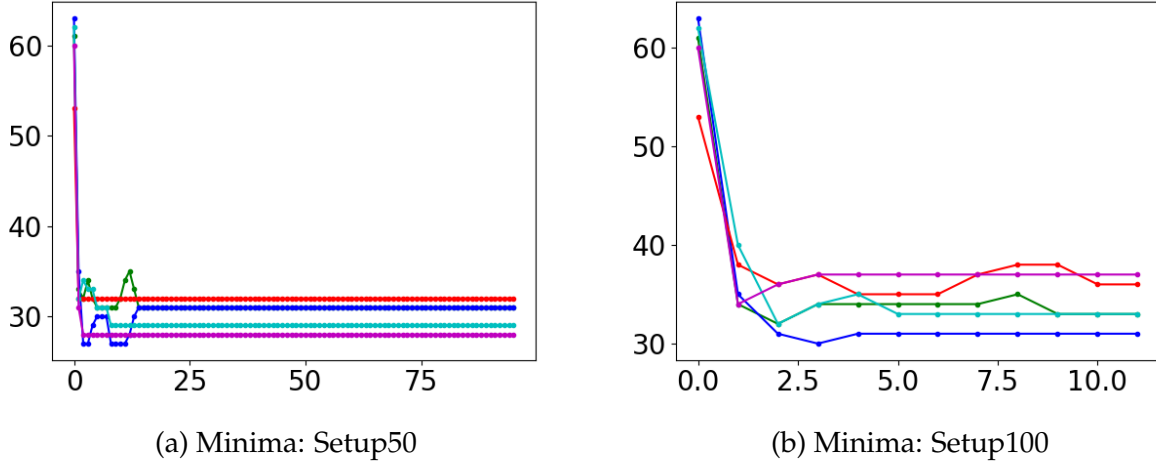


Figure 6: Price paths comparison for Minima: Setup50 vs Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
Q4	39.30 (3.95)	18.50 (1.57)	27.37 (12.22)	23.50 (10.64)	12.49 (3.26)	28.08 (2.85)
Q3	33.20 (3.58)	16.65 (1.16)	18.45 (13.43)	13.70 (3.21)	7.08 (3.91)	25.34 (4.98)
Q2	16.13 (9.32)	15.30 (1.68)	11.93 (3.05)	8.07 (3.26)	3.32 (2.34)	15.31 (6.47)
Q1	7.96 (2.47)	5.32 (1.66)	31.56 (6.06)	15.96 (10.27)	4.17 (3.91)	4.99 (1.53)

(a) Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
Q4	14.82 (3.04)	11.92 (1.48)	15.25 (3.86)	12.55 (3.66)	8.41 (3.58)	11.86 (1.93)
Q3	12.24 (2.24)	11.00 (1.57)	12.16 (4.00)	10.58 (3.91)	5.13 (3.52)	10.24 (1.91)
Q2	7.98 (2.34)	9.12 (1.87)	13.12 (4.38)	7.70 (4.58)	3.17 (2.05)	7.14 (1.96)
Q1	9.57 (3.13)	3.63 (2.33)	31.51 (6.44)	10.81 (8.43)	4.72 (4.10)	4.53 (1.46)

(b) Setup100

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
Q4	13.80 (3.40)	9.86 (2.19)	15.14 (4.45)	11.91 (4.43)	7.23 (3.21)	9.97 (2.11)
Q3	11.91 (3.22)	8.52 (1.86)	13.81 (4.14)	9.99 (4.29)	4.67 (2.93)	8.86 (2.20)
Q2	9.04 (2.64)	7.58 (2.15)	15.47 (4.30)	7.11 (4.82)	3.73 (2.17)	6.74 (2.33)
Q1	11.55 (3.62)	4.70 (2.99)	26.65 (7.26)	7.25 (6.58)	5.30 (4.19)	5.32 (2.08)

(c) Setup50

Table 7: Performance of DJA-Algorithm for different starting prices

5.2 Impact of confidence level

The confidence bound calculated in the estimation phase is a control parameter in our algorithm which controls and captures the risk attitude of the toll-setter/decision maker. A risk averse toll-setter may set the α at a low level to increase the confidence level. This implies the price decisions should be more conservative. On the other hand less risk averse seller may choose to set a more modest level of α . This is also observed in the average regrets give in [Table 8](#), where we report the performances against different levels of confidence from 0.5% to 0.95%. We observe that the regrets are the least (almost in all cases) at the lower level of confidence. The effect is more pronounced for some distributions than others. Two exceptions being Gumbel and Uniform. In Gumbel case, the prices (and the optimal) are in the lower tails of price range. We noticed that the price paths with varying level of confidence are not significantly different and it is most likely that the difference in performance is likely by chance of randomness. Similarly in the Uniform case the difference in the performance is very insignificant. As one would expect the effect of lower confidence would be more visible in the case with more periods. The variance in regrets increases with decreasing confidence, this is representative of the risk or loss of confidence as one may expect.

5.3 Impact of τ^{DJA}

Parameter τ^{DJA} is used to control the maximum duration of the Forced Exploration phase of DJA algorithm. The Forced Exploration phase ends either because of observing a usage greater than UU or because of reaching (or exceeding) its maximum duration, in which case $p^{(i)}$ is set to ω_2 .

In [Section 4](#), τ^{DJA} was set to $20\%T$. [Table 9](#) illustrates the sensitivity of the algorithm to this parameter, by comparing it to three shorter maximum durations, $5\%T$, $10\%T$ and $15\%T$. [Table 9\(c\)](#) indicates that in Setup50, where there are many short pricing periods, the algorithm is generally quite robust, although it is important to note that $10\%T$ and $15\%T$ almost always outperform $20\%T$. The preference between $5\%T$ and $10\%T$ is inconclusive, as it is distribution-dependent.

As the number of pricing periods decreases, see [Table 9\(b\)](#), performance gains from decreasing τ^{DJA} to $10\%T$ or $15\%T$ become inconclusive, and decreasing to $5\%T$ leads to large performance losses for several distributions. Note that in case $5\%T$, the algorithm spends at most 3 pricing periods in the Forced Exploration phase, in which case the prices

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.95	39.30 (3.95)	18.50 (1.57)	27.37 (12.22)	23.50 (10.64)	12.49 (3.26)	28.08 (2.85)
0.80	37.40 (4.59)	18.26 (1.36)	26.76 (12.38)	23.08 (10.58)	12.55 (3.35)	27.47 (2.90)
0.65	36.15 (4.35)	18.15 (1.35)	26.77 (12.35)	22.65 (10.47)	12.49 (3.21)	26.92 (2.84)
0.50	35.05 (4.45)	17.88 (1.15)	26.32 (12.64)	22.27 (10.65)	12.62 (3.51)	26.65 (2.97)

(a) Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.95	14.82 (3.04)	11.92 (1.48)	15.25 (3.86)	12.55 (3.66)	8.41 (3.58)	11.86 (1.93)
0.80	12.97 (2.51)	12.26 (2.21)	13.78 (3.53)	11.58 (3.12)	8.66 (4.04)	10.85 (1.67)
0.65	12.16 (2.40)	12.58 (2.28)	13.17 (3.75)	10.81 (2.82)	8.91 (4.06)	10.49 (1.32)
0.50	11.70 (2.46)	12.96 (2.74)	12.41 (2.92)	10.69 (2.99)	8.92 (4.17)	10.39 (1.35)

(b) Setup100

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.95	13.80 (3.40)	9.86 (2.19)	15.14 (4.45)	11.91 (4.43)	7.23 (3.21)	9.97 (2.11)
0.80	11.19 (2.54)	9.93 (2.27)	12.74 (3.75)	10.49 (3.71)	7.76 (3.72)	8.76 (1.53)
0.65	10.05 (1.96)	10.40 (2.60)	11.84 (3.52)	9.50 (2.98)	7.93 (3.74)	8.34 (1.30)
0.50	9.10 (1.72)	10.39 (2.78)	10.47 (3.15)	9.38 (3.09)	8.24 (4.02)	8.11 (1.28)

(c) Setup50

Table 8: Performance of DJA Algorithm at different values of $(1 - \alpha)\%$

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
5% T	86.51 (4.51)	11.16 (1.52)	88.39 (12.37)	60.45 (23.79)	16.63 (10.64)	54.81 (10.80)
10% T	21.88 (2.74)	12.32 (1.26)	47.69 (4.83)	32.83 (12.50)	12.70 (4.63)	14.34 (1.59)
15% T	18.41 (2.20)	20.95 (3.83)	20.37 (3.13)	15.16 (3.37)	11.08 (2.26)	21.29 (12.83)
20% T	39.30 (3.95)	18.50 (1.57)	27.37 (12.22)	23.50 (10.64)	12.49 (3.26)	28.08 (2.85)

(a) Setup400

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
5% T	33.33 (4.41)	7.29 (1.68)	22.39 (10.56)	17.31 (11.92)	5.52 (2.53)	16.98 (4.37)
10% T	17.80 (4.26)	8.50 (1.60)	16.69 (7.99)	11.04 (3.99)	6.91 (3.26)	11.35 (2.42)
15% T	15.05 (3.18)	10.16 (1.28)	15.55 (4.21)	12.20 (3.80)	7.83 (3.56)	11.51 (2.31)
20% T	14.82 (3.04)	11.92 (1.48)	15.25 (3.86)	12.55 (3.66)	8.41 (3.58)	11.86 (1.93)

(b) Setup100

	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
5% T	17.61 (4.02)	5.63 (1.68)	17.32 (8.34)	9.83 (5.87)	5.37 (2.81)	9.66 (2.75)
10% T	13.76 (3.85)	7.04 (1.90)	15.09 (4.37)	10.31 (4.43)	6.16 (3.08)	9.62 (3.28)
15% T	13.09 (3.07)	8.35 (1.69)	15.95 (4.68)	10.26 (4.09)	6.90 (3.43)	9.55 (2.23)
20% T	13.80 (3.40)	9.86 (2.19)	15.14 (4.45)	11.91 (4.43)	7.23 (3.21)	9.97 (2.11)

(c) Setup50

Table 9: Performance of DJA Algorithm at different values of $\tau^{(\text{DJA})}$

set are $p^{(0)} = \omega_{J-1}$, $p^{(1)} \approx (\omega_{J-1} + \omega_2)/2$, and $p^{(2)} = \omega_2$. The negative effect of decreasing τ^{DJA} is even more pronounced if there are even fewer pricing periods, as shown in Table 9(a). Note that in case 5% T , the algorithm spends at most 2 pricing periods in the Forced Exploration phase, in which case the prices set are $p^{(0)} = \omega_{J-1}$, $p^{(1)} = \omega_2$.

Overall, only Gumbel distribution benefits from very small value of τ^{DJA} , which is because the optimal price is very close to the lowest allowable price q and because all the prices in the neighbourhood of q and of the optimal price yield nearly-optimal revenue.

5.4 Impact of UU

The Conservative nature of the DJA-Algorithm in estimation and pricing means to balance over-conservatism we need a mechanism to make room for reasonable exploration for the algorithm to learn enough about the usage distribution. This is provided by UU limit which also captures a policy perspective, for example in toll road case one goal would be to have certain level of usage. This may also be interpreted as how much toll

setter believes is enough usage to get good revenues. The impact of the level of UU is different in all distributions, with no particular pattern true for all distributions and seems also to vary between setups.

Table 10 suggests that for a fixed τ^{DJA} in case of larger period duration lower UU suffices whereas for shorter durations higher values of UU are required to allow for sufficient exploration. This can be ascertained to the fact that rate of price decrease in forced exploration phase is roughly proportional to period duration. For example, in Setup50 the best results are obtained at 0.35. In almost all cases, UU set at 0.05 does not give enough room for better exploration of the price range, resulting in overly conservative prices with higher variability in performance.

To cover for the sensitivity of the algorithm to UU, it is conceivable that a dynamic UU rule may be incorporated which also updates UU with each observation. Irrespectively of the setup, Minima and Normal distributions seem to be more sensitive to changes in UU levels compared to the other distributions. One explanation for these other distributions, except for Uniform, seems that they are more skewed and that the price ranges for these distributions are also different.

5.5 Discussion

The dynamic and adaptive nature of DJA allows it to avoid unattractive prices if good starting price is chosen, and this feature of the algorithm is extremely important in the presence of price caps. Similarly, the level of conservatism captured by confidence level takes into account the decision maker's risk appetite and is useful in avoiding overly conservative prices even without the knowledge of the distribution. The analysis in subsection 5.1–subsection 5.4 illustrates the gain in the performance that can be achieved by tuning the input parameters to better values. Of course, which parameter values are better depends on the distribution family, the knowledge of which even without knowing the exact distribution parameters can help in making the right choice for values of input parameters. Table 7–Table 10 indicate that it is preferable to choose starting values between Q2 and Q3; α level at 0.5; τ^{DJA} between $15\%T$ and $20\%T$; and UU between 0.25 and 0.35 in order to improve the performance across all distributions, even though in some cases there may be other choices that give better performance for a particular distribution.

Based on these observations from the sensitivity analysis, we illustrate the performance of an improved version of our algorithm, called DJA*, in Table 5, which shows that this version dramatically outperforms BZ in almost all cases. DJA* is based on the

UU	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.05	39.01 (4.87)	18.54 (1.56)	19.98 (6.78)	19.34 (3.84)	12.49 (3.26)	27.61 (3.58)
0.15	39.30 (3.95)	18.50 (1.57)	21.15 (7.18)	19.74 (5.63)	12.49 (3.26)	28.08 (2.85)
0.25	39.30 (3.95)	18.50 (1.57)	27.37 (12.22)	23.50 (10.64)	12.49 (3.26)	28.08 (2.85)
0.35	39.30 (3.95)	18.50 (1.57)	35.02 (12.52)	31.97 (13.86)	12.49 (3.26)	28.08 (2.85)

(a) Setup400

UU	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.05	15.52 (3.14)	11.69 (1.48)	19.83 (6.40)	14.24 (4.45)	7.83 (3.17)	12.06 (1.77)
0.15	14.99 (3.24)	11.69 (1.26)	16.50 (4.80)	12.59 (3.86)	8.29 (3.52)	11.81 (1.85)
0.25	14.82 (3.04)	11.92 (1.48)	15.25 (3.86)	12.55 (3.66)	8.41 (3.58)	11.86 (1.93)
0.35	14.79 (3.04)	11.91 (1.51)	15.00 (3.96)	12.42 (3.54)	8.73 (4.12)	11.97 (1.94)

(b) Setup100

UU	Gamma	Gumbel	Minima	Normal	Uniform	Lognormal
0.05	26.65 (19.39)	9.40 (1.89)	28.62 (16.24)	15.57 (13.80)	7.39 (7.27)	13.81 (9.11)
0.15	14.47 (3.93)	9.45 (2.09)	17.62 (5.07)	12.55 (4.52)	6.88 (2.79)	10.41 (2.13)
0.25	13.80 (3.40)	9.86 (2.19)	15.14 (4.45)	11.91 (4.43)	7.23 (3.21)	9.97 (2.11)
0.35	13.34 (3.14)	9.81 (2.29)	14.29 (3.84)	10.35 (3.75)	7.49 (3.48)	9.82 (2.03)

(c) Setup50

Table 10: Performance of DJA Algorithm at different values of UU.

following parameter values: starting price at Q2, α at 0.5, τ^{DJA} at $20\%T$ and UU at 0.25. Note that a full enumeration of all combinations of potential values of the parameters could lead to an even better performing, optimized, version of our algorithm.

6 Conclusion

We have presented a novel algorithm for dynamic pricing when the demand function is unknown. The algorithm balances the trade-off between price experimentation and revenue maximization, and has been derived by incorporating a novel conservative estimation of the alternative product cost distribution together with a novel non-adversarial robust optimization approach for pricing. The algorithm is flexible to incorporate additional constraints, such as price caps, and performs competitively to an optimized version of a state-of-the-art algorithm that is asymptotically optimal.

Although we have presented the model for a variant, in which there is a single customer that makes repetitive purchasing decisions during every pricing period, the results also apply to another variant, in which there is a known number of customers purchasing in every period. We also believe that an adaptation of the proposed algorithm would perform competitively if the number of customers purchasing in every period is unknown. Minor modifications would be needed to remove the dependence of all the quantities on $N^{(i)}$.

In the estimation step, DJA employs the widely popular normal approximation in the construction of the confidence intervals. This works very well when the period durations are large, but, as the results in the previous section illustrate, the confidence intervals constructed are very sensitive to the period duration. To deal with this sensitivity, one option to look at are other ways of constructing the confidence interval, like Jeffrey's interval, Clopper-Pearson interval, etc. This is especially important when one is constrained to update price at short intervals.

In the presence of price caps and other restrictions, limitation of price experimentation becomes extremely important. Since increasing the price is not free for toll setter, starting at a low price can be very risky resulting in getting stuck at low prices especially in finite horizon problems. Therefore, an algorithm needs to start a higher price. If starting at a reasonably high price a good algorithm able to deal with price caps should avoid falling too low below the optimal price. Therefore, a question toll setter faces is *what is the optimal rate at which to decrease the price if starting at a high price?* In this respect, DJA Algorithm

is semi-myopic as the rate of price decrease depends on the time and UU parameter, and this rate increases with time. Currently, this parameter is fixed, but it is conceivable that it can also be adapted dynamically.

This work mainly dealt with single product pricing, the natural question to consider in future research is an extension to the multi-product setting. From the road toll pricing viewpoint, consider a setting where the network connecting origin and destination consists of routes which are a combination of toll and free roads. While for some topologies the DJA Algorithm can be easily extended, further research is needed to understand the impact of general network structures on pricing. Furthermore, the situation becomes even more complicated when there are more than one origin-destination pairs. At first sight, road pricing in a network may seem disconnected from (any) other product pricing problem, but a natural way of capturing affects such as substitutions, complements etc., between products is by using graphs and networks. So, network pricing naturally captures the features of general pricing problems.

7 Acknowledgements

We would like to thank Dr. Marc Goerigk for helping with Chicago data sets.

References

- [1] D. Albalade, G. Bel, and X. Fageda. Privatization and regulatory reform of toll motorways in europe. *Governance: An International Journal of Policy, Administration, and Institutions*, 22(2):295–318, 2009.
- [2] V. Araman and R. Caldentey. Dynamic pricing for non-perishable products with demand learning. *Operations Research*, 57(5):1169–1188, 2009.
- [3] M. Babaioff, S. Dughmi, R. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. *ACM Transactions on Economics and Computation*, 3(1):4:1–4:26, 2015.
- [4] D. Bergemann and K. Schlag. Pricing without priors. *Journal of European Economic Association.*, 6(2-3):560–569, 2008.
- [5] D. Bergemann and K. Schlag. Robust monopoly pricing. *Journal of Economic Theory.*, 146(6):2527–2543, 2011.

- [6] O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research.*, 57(6):1407–1420, 2009.
- [7] J. Buxbaum and I. Ortiz. *Public Sector Decision Making for Public–Private Partnerships: A Synthesis of Highway Practice*. TRANSPORTATION RESEARCH BOARD, WASHINGTON, D.C., 2009.
- [8] T. Dokka and M. Goerigk. Experimental evaluation of uncertainty sets for robust shortest path problem. *In proceedings of ATMOS 2017*, 56(1):188–203, 2017.
- [9] T. Dokka, A. Zemkoho, S. SenGupta, and F.T. Nobibon. Robust toll pricing. *LUMS Working Paper.*, 2016.
- [10] G. Gallego and G. van Ryzhin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020, 1994.
- [11] J. C. Gittins, K. Glazebrook, and R. Weber. *Multi-Armed Bandit Allocation Indices*. Wiley-Blackwell, 2011.
- [12] J.M. Harrison, N.B. Keskin, and A. Zeevi. Bayesian dynamic pricing policies: learning and earning under a binary prior distributions. *Management Science*, 58(3):570–586, 2012.
- [13] N. Bora Keskin and A. Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research.*, 62(5):1142–1167, 2014.
- [14] A. Lim and J. George Shanthikumar. Relative entropy, exponential utility, and robust dynamic pricing. *Operations Research.*, 55(2):198–214, 2007.
- [15] K.Y Lin. Dynamic pricing with real-time demand learning. *European Journal of Operational Research*, 174(1):522–538, 2006.
- [16] K. Talluri and G. van Ryzhin. *The Theory and Practice of Revenue Management*. Springer, New York, 2004.
- [17] A. van den Boer. Dynamic pricing and learning: Historical origins, current research and new directions. *Surveys in Operations Research and Management Science.*, 20:1–18, 2015.

- [18] A. van den Boer and B. Zwart. Simultaneously learning and optimizing using controlled variance pricing. *Management Science.*, 60(3):770–783, 2014.
- [19] A. van den Boer and B. Zwart. Dynamic pricing and learning with finite inventories. *Operations Research.*, 63(4):965–978, 2015.