Topics in Deep Learning and Data-driven Optimization

Achraf Bahamou

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2023

Abstract

Topics in Deep Learning and Data-driven Optimization

Achraf Bahamou

Data-driven optimization has become an increasingly popular approach for solving complex problems in various domains, such as finance, online retail, and engineering. However, in many real-world applications, the amount of available data can vary significantly, ranging from limited to large data sets. Both of these regimes present unique modeling and optimization challenges.

In this thesis, we explore two distinct problems in two different data availability and model complexity regimes. In the first part (Chapters 2 and 3), we focus on the development of novel optimization algorithms for training deep neural network(DNN) models on large data sets, in particular, we develop practical optimization methods that incorporate curvature information in an economical way to accelerate the optimization process. The performance of the proposed methods is compared to that of several state-of-the-art methods used to train DNNs, to validate their effectiveness both in terms of time efficiency and generalization power.

In the second part of the dissertation (Chapters 4), we focus on data-driven pricing in the limited data regime. More specifically, we study the fundamental problem of a seller pricing a product based on historical information consisting of the observed demand at a single historical price point. We develop a novel framework that allows characterizing optimal performance for deterministic or more general randomized mechanisms and leads to fundamental novel insights on the value of limited demand data for pricing.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisors Donald Goldfarb and Omar Besbes. My PhD achievements would not have been possible without their support and mentorship. The hours we spent working together on complex research problems will always remain etched in my memory. I consider it a great privilege to have learned from both of you, and I am deeply grateful for your dedication to my academic and personal growth. Your faith in my potential, your patience with me in difficult times, and your guidance have been invaluable in shaping my academic journey and beyond. Thank you from the bottom of my heart.

I would also like to thank my thesis committee members: Daniel Bienstock, Cedric Josz and Robert Gower. I am very fortunate and honored to have such amazing researchers on my committee. I would also like to thank all the IEOR administrative staff who were of great help and made IEOR a social and vibrant enjoyable environment, especially Winsor, Lizbeth and Kristen.

I am immensely grateful for the warm and welcoming community at IEOR, where I found a home and made lifelong friends. The journey would not have been as memorable without their presence. I extend my heartfelt thanks to all my fellow Ph.D. students for the incredible moments we shared together, from attending classes and collaborating on research projects to playing soccer, exploring the vibrant city of NYC, traveling, and simply having fun. Thank you: Luc, Jacob, Omar Mouchtaki, Omar El Housni, Amine, Oussama, Yi, Ruizhe, Harsh, Ayoub, Abdellah...

I would like to extend a special thanks to my brother from another mother, Ayoub, who showed remarkable strength and courage in battling cancer, and with whom I shared many special moments

throughout my life and Ph.D. journey. I am also deeply grateful to Othman, with whom I shared numerous unforgettable moments. Your support and friendship are cherished parts of my life.

Last but not least, I want to express my deepest gratitude to my parents and my brother for their boundless love and unwavering support. I am forever indebted to my parents for everything they have done for me, more than words can convey. This thesis is dedicated to my parents, Chafik and Fatima, and my brother Jabir with all my heart.

Dedication

To my family, Chafik, Fatima and Jabir.

Preface

Most of the materials in this thesis are published or submitted works contained in the following papers:

- *A Mini-Block Fisher Method for Deep Neural Networks* [1]. This article is joint with Donald Goldfarb and Yi Ren.
- *A Dynamic Sampling Adaptive-SGD Method for Machine Learning* [2]. This article is joint with Donald Goldfarb.
- *Optimal Pricing with a Single Point* [3]. This article is joint with Omar Besbes and Amine Allouah.

Other contributions that are related but not included in this thesis are:

- *Practical quasi-Newton methods for training deep neural networks* [4]. This article is joint with Donald Goldfarb and Yi Ren.
- *Kronecker-factored Quasi-Newton Methods for Convolutional Neural Networks* [5]. This article is joint with Donald Goldfarb and Yi Ren.
- *Pricing with samples* [6], joint with Omar Besbes and Amine Allouah.

Chapter 1: Introduction

In the era of big data, Data-driven optimization has become a powerful approach that allows practitioners to make informed decisions based on collected historical data and improve the performance of their systems and processes. Data-driven optimization can be applied across diverse industries such as finance, e-commerce, healthcare, and others. For instance, in finance, it can detect fraud and predict market trends while in e-commerce, it can optimize prices and minimize costs.

Generally, the more data available to train a model, the better its predictive performance is likely to be. However, this relationship is not straightforward and can be influenced by other factors, such as the complexity of the model being used, and the effectiveness of the model fitting procedure. For instance, when there is an abundance of data available, more complex models coupled with effective optimization and model-fitting procedures can be used to extract deeper insights and achieve higher predictive performance. However, overly complex models can also lead to overfitting, where the model becomes too closely tailored to the training data and performs poorly on new, unseen data. In situations where data is scarce, simpler models may be more appropriate, as they are less likely to overfit and can still perform adequately.

Overall, the relationship between data availability, model complexity, and model predictive performance is a nuanced and dynamic one that must be carefully considered when developing data-driven modeling solutions. Understanding this relationship is critical to achieving optimal performance in various real-world applications.

This thesis is divided into two main parts tackling two distinct regimes of data availability and model complexity. The first part of the dissertation (Chapters 2 and 3) is dedicated to the development of novel optimization algorithms for training deep neural network models on large data sets, in particular, we devise practical optimization methods that incorporate curvature information in an economical manner to accelerate the optimization process. In the second part of the dissertation

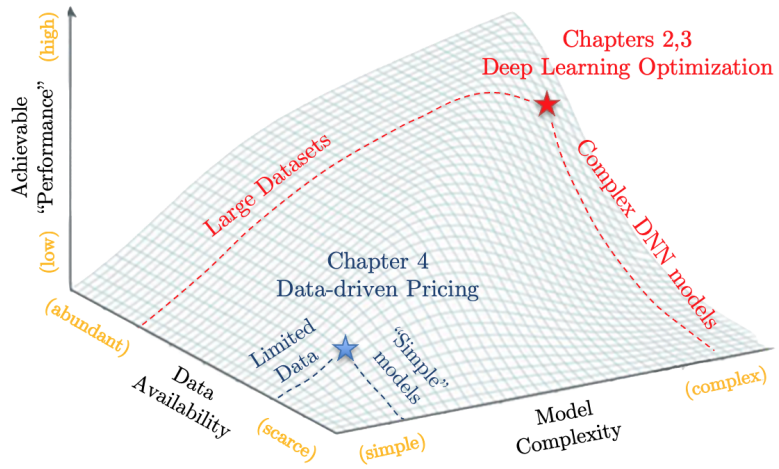


Figure 1.1: A graphical illustration of the relationship between data availability, model complexity, and achievable performance(revised from Figure 1 in [7]).

(Chapter 4), we focus on data-driven pricing in the limited data regime where we study the fundamental problem of a seller pricing a product based on historical information consisting of the observed demand at a single historical incumbent price. Figure 1.1 illustrates how the two main parts are positioned in the landscape of data-driven optimization problems.

Although Parts 1 and 2 fall under the umbrella of data-driven optimization techniques, they are otherwise unrelated and can be read separately.

1.1 Introduction for Part I (Chapters 2 and 3)

Deep learning has revolutionized the field of machine learning in recent years, enabling remarkable breakthroughs in various tasks such as image and speech recognition, natural language processing, and autonomous driving. The effectiveness of deep neural networks in solving complex problems lies in their ability to learn hierarchical representations of data, which allows them to capture intricate patterns and dependencies in the input data.

However, training these networks can be challenging due to the large number of parameters involved and the non-convex nature of the optimization problem. First-order methods based on stochastic gradient descent (SGD) [8], and in particular, the class of adaptive learning rate methods, such as AdaGrad [9], RMSprop [10], and Adam [11], are currently the most widely used methods

to train deep learning models (the recent paper [12] lists 65 methods that have "Adam" or "Ada" as part of their names). While these methods are easy to implement and have low computational complexity, they make use of only a limited amount of curvature information. Standard SGD and its mini-batch variants, use none. SGD with momentum (SGD-m) [13] and stochastic versions of Nesterov's accelerated gradient method [14], implicitly make use of curvature by choosing step directions that combine the negative gradient with a scaled multiple of the previous step direction, very much like the classical conjugate gradient method.

To effectively optimize ill-conditioned functions, one usually needs to use second-order methods, which range from Newton's method to those that use approximations to the Hessian matrix, such as BFGS quasi-Newton (QN) methods [15, 16, 17, 18], including limited memory (LM) variants [19], and Gauss-Newton (GN) methods [20]. To handle large machine learning data sets, stochastic methods such as sub-sampled Newton [21]), QN [22, 23, 24], GN, natural gradient (NG) [25], Hessian-free [26], Krylov subspace, [27], and LM variants of Anderson acceleration [28, 29], that are related to LM multisecant QN methods (see [30]), have been developed. However, in all of these methods, whether they use the Hessian or an approximation to it, the size of the matrix becomes prohibitive when the number of training parameters is huge.

Therefore, deep learning training methods have been proposed that use layer-wise block-diagonal approximations to the second-order preconditioning matrix. These include a Sherman-Morrison-Woodbury based variant [31] and a low-rank variant [32] of the block-diagonal Fisher matrix approximations for NG methods. Also, Kronecker-factored matrix approximations of the diagonal blocks in Fisher matrices have been proposed to reduce the memory and computational requirements of NG methods, starting from KFAC for multilayer perceptrons (MLPs) [33], which was extended to CNNs in [34]; (in addition, see [35, 36, 37]). Kronecker-factored QN methods [38], generalized GN methods [39], an adaptive block learning rate method Shampoo [40], based on AdaGrad, and an approximate NG method TNT [41], based on the assumption that the sampled tensor gradient follows a tensor-normal distribution have also been proposed.

Figure 1.2 summarizes how several existing state-of-the-art methods approximate these diagonal blocks used in the preconditioner matrix.

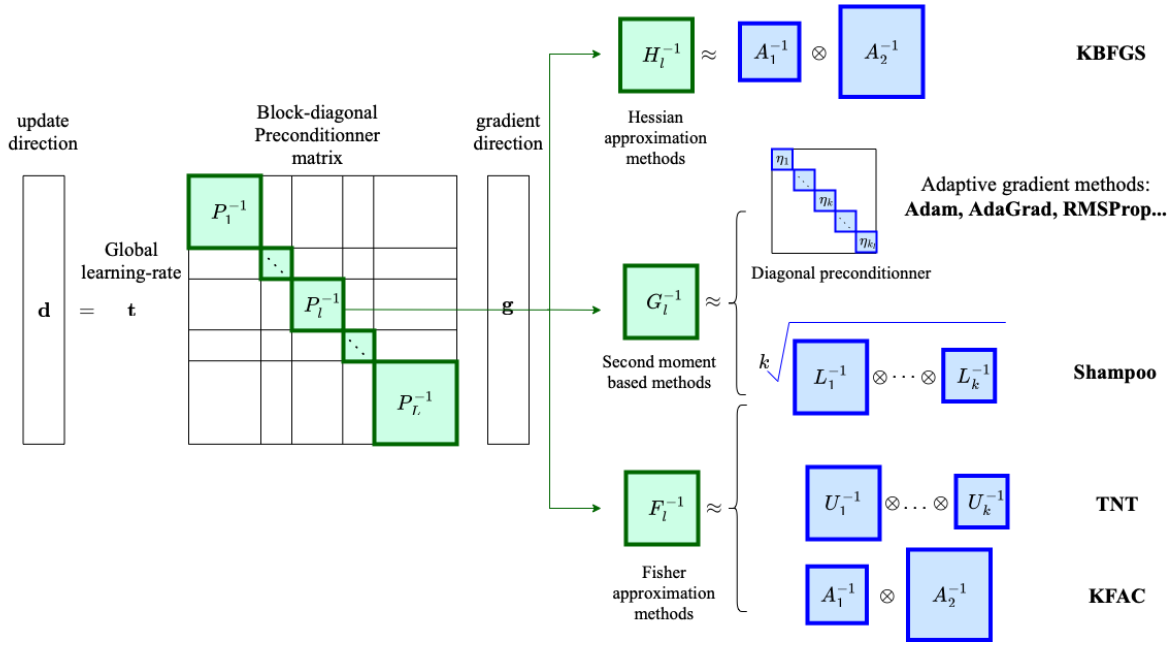


Figure 1.2: Block-diagonal preconditioned gradient methods.

In Chapter 2, we propose a novel approximate natural gradient method named "mini-block Fisher (MBF)", that uses a block-diagonal approximation to the empirical Fisher matrix, where for each layer in the DNN, whether it is convolutional or feed-forward and fully connected, the associated diagonal block is itself block-diagonal and is composed of a large number of mini-blocks of modest size. Our novel approach utilizes the parallelism of GPUs to efficiently perform computations on the large number of matrices in each layer. Consequently, MBF's per-iteration computational cost is only slightly higher than it is for first-order methods. The performance of MBF is compared to that of several baseline methods, on Autoencoder, Convolutional Neural Network (CNN), and Graph Convolutional Network (GCN) problems, to validate its effectiveness both in terms of time efficiency and generalization power. Finally, it is proved that an idealized version of MBF converges linearly.

In Chapter 3, we develop a brand new per-layer adaptive step-size procedure for stochastic first-order optimization methods for minimizing empirical loss functions in deep learning, eliminating

the need for the user to tune the learning rate. The proposed approach exploits local stochastic curvature information and the structure of the block diagonal Hessian in neural networks to compute the per-layer adaptive steps. Crucially, the method has comparable memory requirements to those of first-order methods and avoids tuning the global learning-rate hyper-parameter while its per-iteration time complexity is only roughly equivalent to an additional gradient computation and is much smaller than that of popular second-order methods (e.g. KFAC) for training DNNs. Numerical experiments show that SGD with momentum and AdamW combined with the proposed per-layer step-sizes method are able to choose adequate learning rate schedules and perform favorably to fine-tuned popular first-order and second-order algorithms for training deep neural networks on Convolutional Neural Networks (CNN) and Graph Convolutional Network (GCN) problems. Finally, it is proved that an idealized version of SGD with layer-wise step sizes converges linearly in the full batch setting.

1.2 Introduction for Part II (Chapter 4)

Pricing is a central concept across a large spectrum of industries, ranging from e-commerce to transportation. A key informational dimension faced by decision-makers is the level of knowledge of customers' values. In classical settings in the literature, monopoly pricing problems are studied under the assumption that sellers have an accurate knowledge of consumer preferences through the value distribution (or the prior on values). In those cases, the seller may optimize pricing to maximize the expected revenues.

In practice, however, such information is rarely, if ever, available, and pricing must be conducted not based on the value distribution, but based on historical data. Typical historical data structures in the context of pricing include the prices posted and the responses of consumers observed at those prices: either a customer purchases or not. We illustrate in Figure 1.3 a graphical representation of a typical data-driven pricing cycle in electronic marketplaces.

In this part, we study the fundamental data-driven pricing problem of how a seller should/can design optimal pricing algorithms that maps available data, in the form of a price and past customers'

Figure 1.3: An illustration of the data-driven pricing cycle in electronic marketplaces.

associated observed demand, to the pricing decision. This stream of research has received ample attention from different angles in the literature:

Knowledge about the support of the value distribution. A setting that has been studied is one in which the seller only knows the support of the underlying distribution. Early studies are [42] and [43], in which the authors characterize the optimal pricing policy as well as the worst-case demand distribution with respect to a min-max regret objective in the former and a competitive ratio in the latter. [44] studies a case in which the seller has access to the maximum price at which she would still expect non-zero demand.

Pricing with no price dispersion in the historical data. In [45], the authors studied the related problem of reconstructing demand curves when only a single point has been historically observed and showed how a second point can be extracted from the sales of discounted bundles and used it to estimate linear demand curve parameters. In [46], the authors document a setting at a large OEM where the problem is exactly one with no price dispersion in the historical data and propose a practical approach to determine nominal robust prices.

Pricing with Samples from the value distribution. A data structure that has received attention is one based on samples of the value distribution, as opposed to buy/no buy feedback. Such a structure is typical in market research contexts. [47] studies the sample complexity needed to achieve near-optimal performance. Understanding systematically the performance achievable with a limited

number of observed samples has been a notable open problem across communities. [47], [48] provide achievability results with one sample, [49] and [50] provide achievability results with two samples. In our recent study [51], we provide a novel approach through factor-revealing dynamic programs that provide the first impossibility results with one sample for general randomized pricing policies, the best achievability results for one and two samples, as well as the first achievability results beyond two samples. Beyond the novel methodology developed, these results shed new light on the value of information. Against mhr distributions, just two samples suffice to guarantee more than 70% of oracle performance and 10 samples more than 80% of oracle performance.

Dynamic demand learning. A number of studies look at how to collect and incorporate data on the fly for pricing purposes, in which case an exploration-exploitation trade-off emerges. See [52, 53, 54, 55, 56]. Our research efforts have the potential to significantly improve the performance of such online algorithms by better exploiting the data at hand while dynamically adjusting prices.

As a motivating example, consider an e-commerce firm that has been offering a product at an incumbent price w over the past quarter to a set of heterogeneous consumers, all with values drawn from a value distribution F. The firm observes the fraction of customers who have bought the offered product at the price w; in other words, the firm has an estimate of the probability of sale or conversion rate, the fraction of customers whose values are greater than or equal to w, i.e., an estimate of $\overline{F}(w) = q$ in $[0, 1]$. How should the seller decide on the pricing policy in the following quarter? Can the seller take advantage of the partial demand information extracted (conversion rate at w) to refine her pricing policy? Such historical data structures are commonplace in practice and typically introduce different challenges. The number of past prices that were posted is often very limited and if one only accounts for recent data, can be as low as one, as in the example above. In other words, many historical data structures have very limited price dispersion. This renders elasticity-based price optimization very challenging if not impossible in practice (without further experimentation) when trying to move from data to pricing decisions. A natural question is then if, in the absence of price dispersion, historical data is useful in any way in order to refine pricing decisions.

In Chapter 4, we offer a resounding “yes" to this question and develop a framework to optimize prices given such limited data and quantify the value of such data. More specifically, we consider a setting in which the information available to the seller consists of a percentile of the value distribution and characterizes the value of one measurement for pricing purposes. The results presented in this chapter lay an important foundation for the class of problems we aim to study. In particular, it initiates a systematic study of optimal pricing with percentile data. The mathematical framework enables one to 1) develop optimal pricing strategies and 2) quantify the value of percentile data. It highlights, quite strikingly, that while a single point provides very limited information on the demand curve, it has actually very high informational content. As an example, simply knowing the median and that the value distribution has a monotone increasing hazard rate / mhr (an assumption satisfied by most models used for demand estimation in practice), one can guarantee more than 85% of the performance that an oracle with full knowledge of the distribution could have achieved.

Chapter 2: A Mini-Block Fisher Method for Deep Neural Networks

In this chapter, we propose a new *Mini-Block Fisher* (MBF) gradient method that lies in between adaptive first-order methods and block diagonal second-order methods. Specifically, MBF uses a block-diagonal approximation to the empirical Fisher matrix, where for each DNN layer, whether it is convolutional or feed-forward and fully-connected, the associated diagonal block is also block-diagonal and is composed of a large number of mini-blocks of modest size.

Crucially, MBF has comparable memory requirements to those of first-order methods, while its per-iteration time complexity is smaller, and in many cases, much smaller than that of popular second-order methods (e.g. KFAC) for training DNNs. Further, we prove convergence results for a variant of MBF under relatively mild conditions.

In numerical experiments on well-established Autoencoder, CNN and GCN models, MBF consistently outperformed state-of-the-art (SOTA) first-order methods (SGD-m and Adam) and performed favorably compared to popular second-order methods (KFAC and Shampoo).

2.1 Notation and Definitions

Notation. $\text{Diag}_{i\in[L]}(A_i)$ is the block diagonal matrix with $\{A_1, ..., A_L\}$ on its diagonal; $[L] := \{1, ..., L\}$; $\boldsymbol{X} = [x_1, ..., x_n]^\top \in \mathbb{R}^{n\times d}$ is the input data; $\lambda_{\min}(M), \lambda_{\max}(M)$ are the smallest and largest eigenvalues of the matrix M; $\otimes$ denotes the Kronecker product; $\|.\|_2$ denotes the Euclidean norm of a vector or matrix; and $\text{vec}(A)$ vectorizes A by stacking its columns.

We consider a DNN with L layers, defined by weight matrices W_l, for $l \in [L]$, that transforms the input vector $\boldsymbol{x}$ to an output $f(\boldsymbol{W}, \boldsymbol{x})$. For a data-point (x, y), the loss $\ell\left(f(\boldsymbol{W}, \boldsymbol{x}), y\right)$ between the output $f(\boldsymbol{W}, \boldsymbol{x})$ and y, is a non-convex function of $\text{vec}(\boldsymbol{W})^\top = \left[\text{vec}\left(W_1\right)^\top, ..., \text{vec}\left(W_L\right)^\top\right] \in \mathbb{R}^p$, containing all of the network's parameters, and ℓ measures the accuracy of the prediction (e.g.

squared error loss, cross-entropy loss). The optimal parameters are obtained by minimizing the average loss $\mathcal{L}$ over the training set:

$$\mathcal{L}(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{W}, \mathbf{x}_i), \mathbf{y}_i), \tag{2.1.1}$$

This setting is applicable to most common models in deep learning such as multilayer perceptrons (MLPs), CNNs, recurrent neural networks (RNNs), etc. In these models, the trainable parameter W_l ($l = 1, \ldots, L$) come from the weights of a layer, whether it be a feed-forward, convolutional, recurrent, etc. For the weight matrix $W_l \in \mathbb{R}^{p_l}$ corresponding to layer l and a subset of indices $b \subset \{1, \ldots, p_l\}$, we denote by $W_{l,b}$, the subset of parameters of W_l corresponding to b.

The average gradient over a mini-batch of size m, $\boldsymbol{g}^{(m)} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial \ell(f(\boldsymbol{W}, \mathbf{x}_i), \mathbf{y}_i)}{\partial \boldsymbol{W}}$, is computed using standard back-propagation. In the full-batch case, where $m = n$, $\boldsymbol{g}^{(n)} = \boldsymbol{g} = \frac{\partial \mathcal{L}(\boldsymbol{W})}{\partial \boldsymbol{W}} = \mathcal{D}\boldsymbol{W}$. Here, we are using the notation $\mathcal{D}\boldsymbol{X} := \frac{\partial \mathcal{L}(\boldsymbol{W})}{\partial \boldsymbol{X}}$ for any subset of variables $\boldsymbol{X} \subset \boldsymbol{W}$.

The Jacobian $\boldsymbol{J}(\boldsymbol{W})$ of the loss $\mathcal{L}(\cdot)$ w.r.t the parameters $\boldsymbol{W}$ for a single output network is defined as $\boldsymbol{J} = [\boldsymbol{J}_1^\top, ..., \boldsymbol{J}_n^\top]^\top \in \mathbb{R}^{n \times p}$, where $\boldsymbol{J}_i^\top$ is the gradient of the loss w.r.t the parameters, i.e., $\boldsymbol{J}_i^\top = \text{vec}(\frac{\partial \ell(f(\boldsymbol{W}, \mathbf{x}_i), \mathbf{y}_i)}{\partial \boldsymbol{W}})$. We use the notation $\boldsymbol{J}_i^{X\top} = \text{vec}(\frac{\partial \ell(f(\boldsymbol{W}, \mathbf{x}_i), \mathbf{y}_i)}{\partial X})$ and $\boldsymbol{J}^X = [\boldsymbol{J}_1^{X\top}, ..., \boldsymbol{J}_n^{X\top}]^\top$ for any subset of variables X of $\boldsymbol{W}$.

The Fisher matrix $\boldsymbol{F}(\boldsymbol{W})$ of the model's conditional distribution is defined as

$$\boldsymbol{F}(\boldsymbol{W}) = \mathop{\mathbb{E}}_{\substack{x \sim Q_x \\ y \sim p_{\boldsymbol{W}}(\cdot|x)}} \left[\frac{\partial \log p_{\boldsymbol{W}}(y|x)}{\partial \boldsymbol{W}} \left(\frac{\partial \log p_{\boldsymbol{W}}(y|x)}{\partial \boldsymbol{W}} \right)^\top \right],$$

where Q_x is the data distribution of x and $p_{\boldsymbol{W}}(\cdot|x)$ is the density function of the conditional distribution defined by the model with a given input x. As shown in [57], $\boldsymbol{F}(\boldsymbol{W})$ is equivalent to the Generalized Gauss-Newton (GGN) matrix if the conditional distribution is in the exponential family, e.g., a categorical distribution for classification or a Gaussian distribution for regression.

The empirical Fisher matrix (EFM) $\tilde{\boldsymbol{F}}(\boldsymbol{W})$ defined as:

$$\begin{aligned}\tilde{\boldsymbol{F}}(\boldsymbol{W}) &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial \ell(f(\boldsymbol{W},\mathbf{x}_i),\mathbf{y}_i)}{\partial \boldsymbol{W}}\frac{\partial \ell(f(\boldsymbol{W},\mathbf{x}_i),\mathbf{y}_i)}{\partial \boldsymbol{W}}^{\top}\\ &= \frac{1}{n}\boldsymbol{J}(\boldsymbol{W})^{\top}\boldsymbol{J}(\boldsymbol{W}),\end{aligned}$$

is obtained by replacing the expectation over the model's distribution in $\boldsymbol{F}(\boldsymbol{W})$ by an average over the empirical data. MBF uses the EFM rather than the Fisher matrix, since doing so does not require extra backward passes to compute additional gradients and memory to store them. We note that, as discussed in [58] and [59], the EFM , which is an un-centered second moment of the gradient, captures less curvature information than the Fisher matrix, which coincides with the GGN matrix in many important cases, and hence is closely related to $\nabla^2\mathcal{L}(\boldsymbol{W})$. To simplify notation we will henceforth drop the "tilde"˜and denote the EFM by $\boldsymbol{F}$. We denote by $\boldsymbol{F}^X = \frac{1}{n}(\boldsymbol{J}^X)^{\top}\boldsymbol{J}^X$, the sub-block of $\boldsymbol{F}(\boldsymbol{W})$ associated with any subset of variables $X \subset \boldsymbol{W}$, and write $(\boldsymbol{F}^X)^{-1}$ as F_X^{-1}.

2.2 The Mini-block Fisher (MBF) Method:

At each iteration, MBF preconditions the gradient direction by the inverse of a damped EFM:

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \alpha\left(\boldsymbol{F}(\boldsymbol{W}(k)) + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{g}(k), \tag{2.2.1}$$

where α is the learning rate and λ is the damping parameter.

To avoid the work of computing and storing the inverse of the $p \times p$ damped EFM, $(\boldsymbol{F} + \lambda I)^{-1}$, where p can be in the millions, we assume, as in KFAC and Shampoo, that the EFM has a block diagonal structure, where the l_{th} diagonal block corresponds to the second moment of the gradient of the model w.r.t to the weights in the l_{th} layer. Hence, the block-diagonal EFM is:

$$\boldsymbol{F}(\boldsymbol{W}) \approx \text{Diag}\left(\boldsymbol{F}^{W_1}, ..., \boldsymbol{F}^{W_L}\right).$$

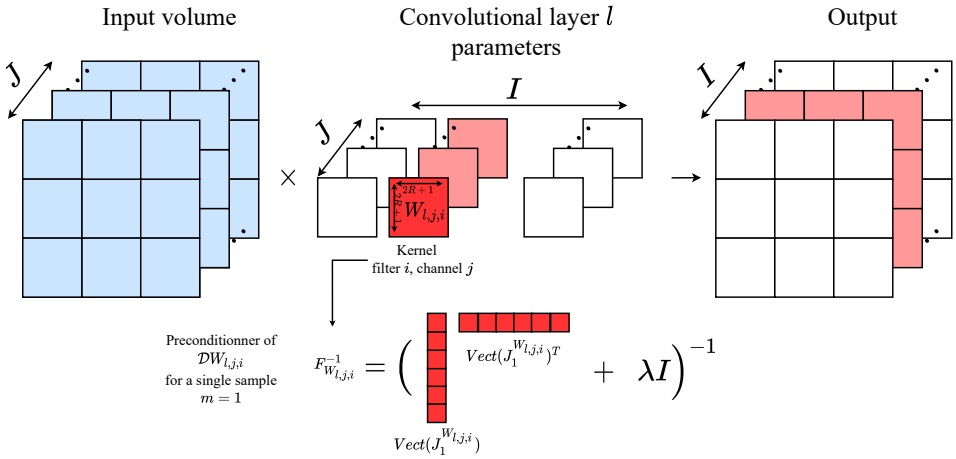


Figure 2.1: Illustration of MBF's approximation for a convolutional layer.

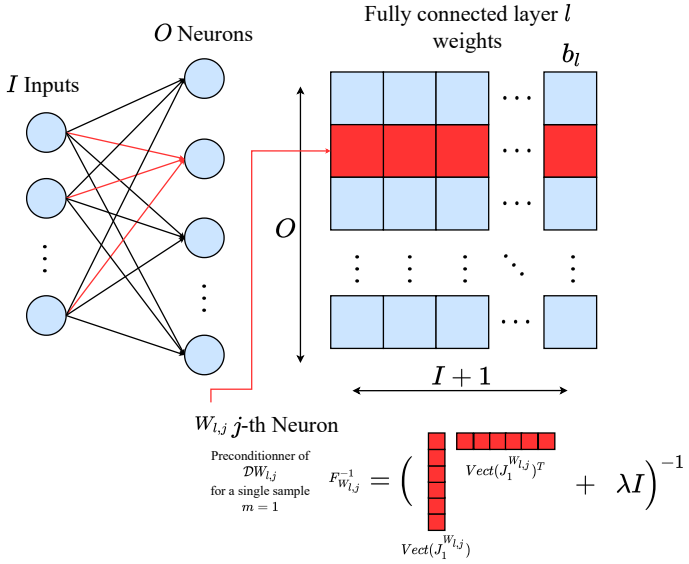


Figure 2.2: Illustration of MBF's preconditionner for a feed-forward fully-connected layer.

MBF further approximates each of the diagonal blocks F_{W_l} by a block-diagonal matrix, composed of a typically large number mini-blocks, depending on the nature of layer l, as follows:

Layer l is convolutional : For simplicity, we assume that the convolutional layer l is 2-dimensional and has J input channels indexed by $j = 1, ..., J$, and I output channels indexed by $i = 1, ..., I$; there are $I \times J$ kernels $W_{l,j,i}$, each of size $(2R+1) \times (2R+1)$, with spatial offsets from the centers of each filter indexed by $\delta \in \Delta := \{-R, ..., R\} \times \{-R, ..., R\}$; the stride is of length 1, and the padding is equal to R, so that the sets of input and output spatial locations ($t \in \mathcal{T} \subset \mathbf{R}^2$) are

the same. For such layers, we use the following $(IJ + 1) \times (IJ + 1)$ block-diagonal approximation to the l_{th} diagonal block $\boldsymbol{F}^{W_l}$ of the Fisher matrix

$$\text{diag}\{\boldsymbol{F}^{W_{l,1,1}}, ..., \boldsymbol{F}^{W_{l,1,I}}, ..., \boldsymbol{F}^{W_{l,J,1}}, ..., \boldsymbol{F}^{W_{l,J,I}}, \boldsymbol{F}^{b_l}\},$$

where each of the IJ diagonal blocks $\boldsymbol{F}^{W_{l,j,i}}$ is a $|\Delta| \times |\Delta|$ symmetric matrix corresponding to the kernel vector $W_{l,j,i}$ and where $\boldsymbol{F}^{b_l}$ is an $I \times I$ diagonal matrix corresponding the bias vector b_l. Therefore, the preconditioning matrix $F^{-1}_{W_{l,j,i}}$ corresponding to the kernel for input-output channel pair (j, i) is given by:

$$F^{-1}_{W_{l,j,i}} := \left(\frac{1}{n}(\boldsymbol{J}^{W_{l,j,i}})^T \boldsymbol{J}^{W_{l,j,i}} + \lambda I\right)^{-1}$$

A common choice in CNNs is to use either a 3×3 or 5×5 kernel for all of the IJ channel pairs in a layer. Therefore, all of these matrices are of the same (small) size, $|\Delta| \times |\Delta|$, and can be inverted efficiently by utilizing the parallelism of GPUs.

We illustrate MBF's approximation for a convolutional layer for the case of one data point in Figure 2.1. From Figure 2.1, it is apparent that the kernel matrices in a convolutional layer that connect the input to the output channels are analogous to the scalar weights that connect the input to output nodes in an ff-cc layer. Hence, the "mini" diagonal blocks $\boldsymbol{F}^{W_{l,j,i}}$ in MBF are analogous to the squares of the components of the gradient in a ff-cc network, and hence MBF can be viewed as a "squared" version of an adaptive first-order method. This observation was in fact the motivation for our development of the MBF approach. In more detail, for simplicity, we drop the subscript l and denote by W the weights corresponding to the elements of all of the filters in this convolution layer. Each element of W is denoted by $W_{i,j,\delta}$, where the first two indices i, j are the output/input channels, and the third index δ specifies the spatial offset within a filter as indicated in item 3 above. The bias b is a vector of length I.

For the weights and biases, we define the vectors

$$\mathbf{w}_i := \left(w_{i,1,\delta_1}, ..., w_{i,J,\delta_{|\Delta|}}, b_i\right)^\top \in \mathbb{R}^{J|\Delta|+1},$$

for $i = 1, ..., I$, and from them the matrix

$$W := (\mathbf{w}_1, ..., \mathbf{w}_I)^\top \in \mathbb{R}^{I \times (J|\Delta|+1)}. \quad (1)$$

We shall also express the vectors $\mathbf{w}_i$ as

$$\mathbf{w}_i := \left(\hat{\mathbf{w}}_{i,1}^\top, ..., \hat{\mathbf{w}}_{i,J}^\top, b_i\right)^\top \in \mathbb{R}^{J\Delta+1}, \quad \text{for all } \ i \in I,$$

where

$$\hat{\mathbf{w}}_{i,j} := (\mathbf{w}_{i,1,j}, \ldots, \mathbf{w}_{i,\Delta,j})^\top \in \mathbb{R}^{\Delta}, \quad \text{for all } \ i \in I, \ j \in J.$$

Let the vector $\mathbf{a} := \{a_{1,t}, \ldots, a_{J,t}\}$, where $a_{j,t}$, denotes the input from channel j of the previous layer to the current layer after padding is added, where t denotes the spatial location of the padded input. Note that the index pairs $t \in \mathcal{T} \subset \mathbf{R}^2$ can be ordered, for example, lexicographically, into a one dimensional set of Δ indices.

It is useful to expand each component $a_{j,t}$ of $\mathbf{a}$ to a Δ-dimensional vector $\hat{\mathbf{a}}_{j,t}$, that includes all components in the input $\mathbf{a}$ covered by the filter centered at t, yielding the following vectors defined for all locations $t \in \mathcal{T}$:

$$\mathbf{a}_t := \left(\hat{\mathbf{a}}_{1,t}^\top, ..., \hat{\mathbf{a}}_{J,t}^\top, 1\right)^\top \in \mathbb{R}^{J\Delta+1},$$

where

$$\hat{\mathbf{a}}_{j,t} := (\mathbf{a}_{j,1,t}, \ldots, \mathbf{a}_{j,\Delta,t})^\top \in \mathbb{R}^{\Delta}, \quad \text{for all } \ j \in J;$$

hence

$$\mathbf{a}_t := \left(a_{1,t+\delta_1}, ..., a_{J,t+\delta_{|\Delta|}}, 1\right)^\top \in \mathbb{R}^{J|\Delta|+1}.$$

Note that a single homogeneous coordinate is concatenated at the end of $\mathbf{a}_t$. Expressing the pre-activation output for the layer at spatial location $t \in \mathcal{T}$ as a vector of length equal to the number of

output channels, i.e.,

$$\mathbf{h}_t := \left(h_{1,t}, ..., h_{I,t}\right)^\top \in \mathbb{R}^I,$$

for all spatial locations $t \in \mathcal{T}$. We note that, given inputs $\mathbf{a}$ and W, the pre-activation outputs $\mathbf{h}$ can be computed, for all locations $t \in \mathcal{T}$, as

$$h_{i,t} = \sum_{j=1}^{J} \sum_{\delta \in \Delta} w_{i,j,\delta} a_{j,t+\delta} + b_i, \quad t \in \mathcal{T},\ i = 1, ..., I. \tag{2.2.2}$$

or equivalently, $\mathbf{h}_t = W\mathbf{a}_t$, whose i-th component $h_{i,t}$ we can write as

$$h_{i,t} = \sum_{j \in J} \hat{\mathbf{w}}_{i,j}^\top \hat{\mathbf{a}}_{j,t} + b_i. \quad (2)$$

Expressing the input-output relationship in a CNN this way, we see that it is analogous to the input-output relationship in a fully connected feed-forward NN, except that the role of input and output node sets J and I are taken on by the input and output channels and the affine mapping of the vector of inputs $\mathbf{a}$ to the vector of outputs $\mathbf{h}$,

$$h_i = \sum_{j \in J} w_{i,j} a_j + b_i, \quad \text{for all } \ i \in I,$$

where the terms $w_{i,j}a_j$ are the products of two scalars become in (2) the inner product of two Δ-dimensional vectors, and this mapping is performed for all locations t. Hence, MBF is analogous to using the squares of the components of the gradient in an ff-cc network and therefore is similar to a "squared" version of an adaptive first-order method.

Layer l is feed-forward and fully connected (ff-fc): For a ff-fc layer with I inputs and O outputs, we use the following $O \times O$ block-diagonal approximation to the Fisher matrix

$$\boldsymbol{F}^{W_l} \approx \text{diag}\{\boldsymbol{F}^{W_{l,1}}, \ldots, \boldsymbol{F}^{W_{l,O}}\},$$

whose j_{th} diagonal block $\boldsymbol{F}^{W_{l,j}}$ is an $(I+1)\times(I+1)$ symmetric matrix corresponding to the vector $W_{l,j}$ of I weights from all of the input neurons and the bias to the j_{th} output neuron. Therefore, the preconditioning matrix $F^{-1}_{W_{l,j}}$ corresponding to the j_{th} output neuron is given by:

$$F^{-1}_{W_{l,j}} := \left(\frac{1}{n}(\boldsymbol{J}^{W_{l,j}})^T\boldsymbol{J}^{W_{l,j}} + \lambda I\right)^{-1}$$

Our choice of such a mini-block subdivision was motivated by the findings presented in [32], first derived in [60], where it was shown that the Hessian of a neural network with one hidden layer with cross-entropy loss converges during optimization to a block-diagonal matrix, where the diagonal blocks correspond to the weights linking all the input units to one hidden unit and all of the hidden units to one output unit.

This suggests that a similar block-diagonal structure applies to the Fisher matrix in the limit of a sequence of iterates produced by an optimization algorithm. The latter suggestion was indeed confirmed by findings presented in [61], where the authors proved that a "unit-wise" block diagonal approximation to the Fisher information matrix is close to the full matrix modulo off-diagonal blocks of small magnitude, which provides a justification for the quasi-diagonal natural gradient method proposed in [62] and our mini-block approximation in the case of fully connected layers. Finally, since the O matrices $\boldsymbol{F}^{W_{l,j}}$, for $j = 1,\ldots,O$, are all of the same size, $(I+1)\times(I+1)$, they can be inverted efficiently by utilizing the parallelism of GPUs. We illustrate MBF's ability to approximate the EFM of a fully connected layer for the case of one data-point in Figure 2.4 for a 7-layer (256-20-20-20-20-20-10) feed-forward DNN using tanh activations, partially trained to classify a 16×16 down-scaled version of MNIST as in [33].

Algorithm 2.1 gives the pseudo-code for a generic version of MBF.

Since updating the Fisher mini-blocks is time consuming in practice as it requires storing and computing the individual gradients, we propose in Section 2.4, a practical approach for approximating these matrices. However, we first present empirical results that justify and motivate both the kernel-based and the all-to-one mini-block subdivisions described above for convolutional

Algorithm 2.1: Generic MBF training algorithm

Input: Given learning rates $\{\alpha_k\}$, damping value λ, batch size m.

for $k = 1, 2, ...$ **do**

  Sample mini-batch M of size m

  Perform a forward-backward pass over M to compute stochastic gradient $\mathcal{D}W_l$ $(l = 1, ..., L)$

  for $l = 1, ..., L$ **do**

    for *mini-block b in layer l,* ***in parallel*** **do**

$$F_{W_{l,b}}^{-1} := \left(\frac{1}{m}(\boldsymbol{J}^{W_{l,b}})^T \boldsymbol{J}^{W_{l,b}} + \lambda I\right)^{-1}$$

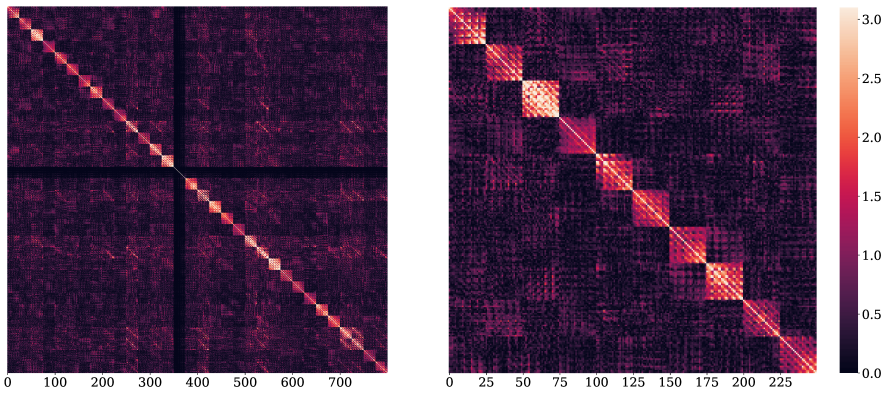
$$W_{l,b} = W_{l,b} - \alpha_k F_{W_{l,b}}^{-1} \mathcal{D}W_{l,b}$$

and ff-fc layers, respectively, followed by a discussion of the linear convergence of an idealized version of the generic MBF algorithm.

After deriving our MBF method, we became aware of the paper [63], which proposes using sub-layer block-diagonal preconditioning matrices for Shampoo, a tensor based DNN training method. Specifically, it considers two cases: partitioning (i) very large individual ff-fc matrices (illustrating this for a matrix of size $[2^9 \times 2^{11}]$ into either a 1×2 or a 2×2 block matrix with blocks all of the same size) and (ii) ResNet-50 layer-wise matrices into sub-layer blocks of size 128. However, [63] does not propose a precise method for using mini-blocks as does MBF.

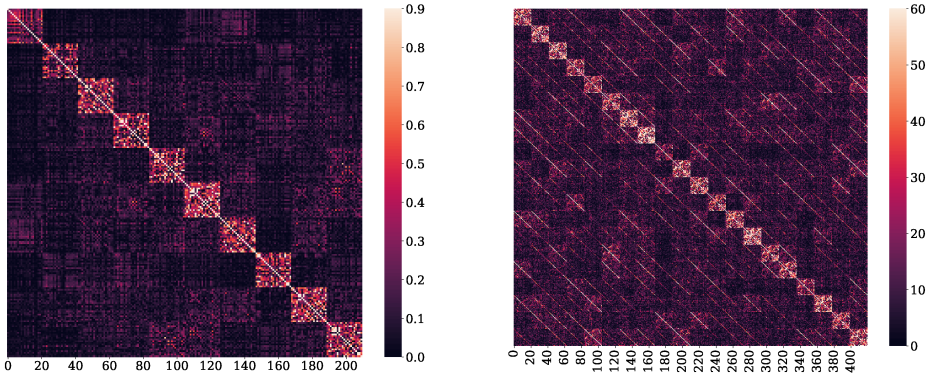
Motivation for MBF: Our choice of mini-blocks for both the convolutional and ff-fc layers was motivated by the observation that most of the weight in the EFM inverse resides in diagonal blocks, and in particular in the mini-blocks described above. More specifically, to illustrate this observation for convolutional layers, we trained a simple convolutional neural network, Simple CNN, on Fashion MNIST [64]. Figure 2.3 shows the heatmap of the absolute value of the EFM inverse corresponding to the first convolutional layer, which uses 32 filters of size 5×5 (thus 32 mini-blocks of size 25×25). One can see that the mini-block (by filter) diagonal approximation is reasonable. Figures for the 2nd convolutional layer are included in section 2.5.4.9. Since the ff-fc layers in the Simple-CNN model result in an EFM for those layers that is too large to work with, we chose to illustrate the mini-block structure of the EFM on a standard DNN, partially trained to

classify a 16×16 down-scaled version of MNIST that was also used in [33]. Figure 2.4 shows the heatmap of the absolute value of the EFM inverse for the last and middle fully connected layers (including bias). One can see that the mini-block (by neuron) diagonal approximation is reasonable. A larger figure for the second fully-connected layer is included in section 2.5.4.9).



(a) First CNN layer (b) Zoom on first 10 blocks

Figure 2.3: Absolute EFM inverse after 10 epochs for the first convolutional layer of the Simple CNN network that uses 32 filters of size 5×5.



(a) Last layer (b) Middle layer

Figure 2.4: Absolute EFM inverse after 50 epochs of the last and middle layers (including bias) of a small FCC-NN.

Comparison: directions of MBF and other methods vs. full block-diagonal EFM: To explore how close MBF's direction is to the one obtained by a block-diagonal full EFM method (BDF), where each block corresponds to one layer's full EFM in the model, we computed the cosine similarity between these two directions. We also included SOTA first-order (SGD-m, Adam) and

second-order (KFAC, Shampoo) methods for reference. The algorithms were run on a 16×16 down-scaled MNIST [65] dataset and a small feed-forward NN with layer widths 256-20-20-20-20-20-10 described in [33]. As in [33], we only show the middle four layers. For all methods, we followed the trajectory obtained using the BDF method. In our implementation of the BDF method, both the gradient and the block- EFM matrices were estimated with a moving-average scheme, with the decay factors set to 0.9. Note that MBF-True refers to the version of MBF in which, similarly to KFAC, the mini-block Fisher is computed by drawing one label from the model distribution for each input image as opposed to MBF, where we use the average over the empirical data. For more details, see section 2.5.4.5.

As shown in Figure 2.5, the cosine similarity between the MBF and MBF-True and the BDF direction falls on most iterations between 0.6 to 0.7 for all four layers and not surprisingly, falls midway between the SOTA first-order and block-diagonal second order methods - always better than SGD-m and Adam, but usually lower than that of KFAC and Shampoo. Moreover, the closeness of the plots for MBF and MBF-True shows that using moving average mini-block versions of the 5 EFM rather than the Fisher matrix does not significantly affect the effectiveness of our approach.

We also report a comparison of the performance of MBF-True and MBF on autoencoders and CNN problems in section 2.5.4.5. Note that, in MBF-True, the only difference between it and MBF is that we are using the mini-batch gradient $\overline{\mathcal{D}_2 W_{l,b}}$ (denoted by $\mathcal{D}_2$) of the model on sampled labels y_t from the model's distribution to update the estimate of mini-block preconditioners, using a moving average (see lines 12, 13 in Algorithm 2.4 in section 2.5.4.4), with a rank one outer-product, which is different from computing the true Fisher for that mini-block.

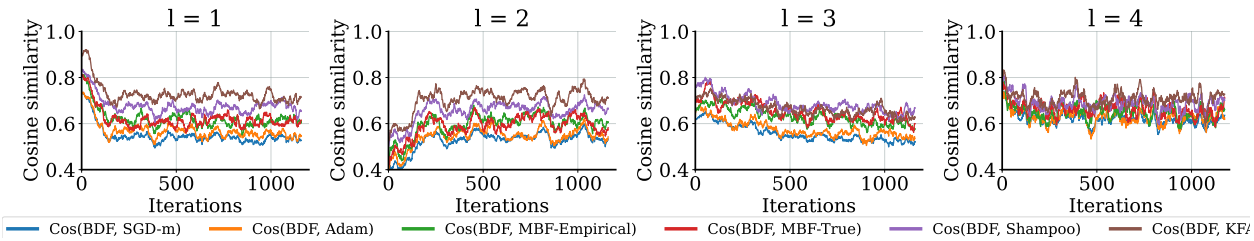


Figure 2.5: Cosine similarity between the directions produced by the methods shown in the legend and that of a block diagonal Fisher method (BDF).

2.3 Linear Convergence

We follow the framework established in [66] to provide convergence guarantees for the idealized MBF with exact gradients (i.e. full batch case with $m = n$) and the mini-block version of the true Fisher matrix, rather than the EFM, as the underlying preconditioning matrix. We focus on the single-output case with squared error loss, but analysis of the multiple-output case is similar.

We denote by $\mathrm{u}(\boldsymbol{W}) = [f(\boldsymbol{W}, x_1), ..., f(\boldsymbol{W}, x_n)]^\top$ the output vector and $y = [y_1, ..., y_n]^\top$ the true labels. We consider the squared error loss $\mathcal{L}$ on a given data-set $\{x_i, y_i\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, i.e. the objective is to minimize

$$\min_{\boldsymbol{W} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{W}) = \frac{1}{2} \| \mathrm{u}(\boldsymbol{W}) - y \|^2.$$

The update rule of MBF with exact gradient becomes

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - \eta \left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{J}(k)^\top (\mathrm{u}(\boldsymbol{W}(k)) - y),$$

where $\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) := \frac{1}{n} \boldsymbol{J}_{MB}(\boldsymbol{W}(k))^\top \boldsymbol{J}_{MB}(\boldsymbol{W}(k))$ is the mini-block-Fisher matrix and the mini-block Jacobian is defined as $\boldsymbol{J}_{MB}(k) = \mathrm{Diag}_{l \in [L]} \mathrm{Diag}_b \left(J^{\boldsymbol{W}_{l,b}}(k) \right)$ and

$$J^{\boldsymbol{W}_{l,b}}(k) := [\frac{\partial f(\boldsymbol{W}(k), \mathbf{x}_1)}{\partial \boldsymbol{W}_{l,b}}, ..., \frac{\partial f(\boldsymbol{W}(k), \mathbf{x}_n)}{\partial \boldsymbol{W}_{l,b}}]^\top$$

We use similar assumptions to those used in [66], where the first assumption, ensures that at initialization, the mini-block Gram matrices are all positive-definite, (i.e., the rows of their respective Jacobians are linearly independent), and the second assumption ensures the stability of the Jacobians by requiring that the network is close to a linearized network at initialization and therefore MBF's update is close to the gradient descent direction in the output space. These assumptions allow us to control the convergence rate.

Assumption 2.1. *The mini-block Gram matrices $J^{\boldsymbol{W}_{l,b}}(0) J^{\boldsymbol{W}_{l,b}}(0)^T$ at initialization are positive definite, i.e.* $\min_{l \in [L]} \min_b \lambda_{min}(J^{\boldsymbol{W}_{l,b}}(0)^T J^{\boldsymbol{W}_{l,b}}(0)) = \lambda_0 > 0$.

Assumption 2.2. *There exists* $0 < C \leq \frac{1}{2}$ *that satisfies* $\|\boldsymbol{J}(\boldsymbol{W}(k)) - \boldsymbol{J}(\boldsymbol{W}(0))\|_2 \leq \frac{C}{3}\sqrt{\lambda_0}$ *if* $\|\boldsymbol{W}(k) - \boldsymbol{W}(0)\|_2 \leq \frac{3}{\sqrt{\lambda_0}}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2$.

Theorem 2.1. *Suppose Assumptions 2.1, 2.2 hold. Consider the Generic MBF Algorithm 2.1, using exact gradients and the mini-block version of the true Fisher as the underlying preconditioning matrix for a network with L layers. Then there exists an interval of suitable damping values* λ *in* $[\underline{\lambda}, \overline{\lambda}]$ *and corresponding small enough learning rates* η_λ*, such that for any learning rate* $0 \leq \eta \leq \eta_\lambda$ *we have* $\|\,\mathrm{u}(\boldsymbol{W}(k)) - \,\mathrm{y}\|_2^2 \leq (1-\eta)^k \|\,\mathrm{u}(\boldsymbol{W}(0)) - \,\mathrm{y}\|_2^2$.

Theorem 2.1 states that an idealized version of MBF converges to the global optimum with a linear rate under Assumptions 2.1 and 2.2. Our analysis is an adaptation of the proof in [66], that uses exact gradients (i.e. full batch case with $m = n$), where we first exploit Assumptions 2.1 and 2.2 to obtain a positive lower bound on the eigenvalues of the mini-block version of the true Fisher matrix $\boldsymbol{F}_{MB}(\boldsymbol{W}(k))$, which then allows us to characterize the rate of convergence of the method.

Proof of Theorem 2.1. If Assumption 6.2 holds, then one can obtain a lower bound on the minimum eigenvalue of the mini-block Fisher matrix $\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) = \frac{1}{n}\boldsymbol{J}_{MB}(k)^\top \boldsymbol{J}_{MB}(k)$ at each iteration.

In fact, if $\|\boldsymbol{W}(k) - \boldsymbol{W}(0)\|_2 \leq \frac{3}{\sqrt{\lambda_0}}\|\boldsymbol{y} - \boldsymbol{u}(0)\|_2$, then, by Assumption 6.2, there exists $0 < C \leq \frac{1}{2}$ that satisfies $\|\boldsymbol{J}(\boldsymbol{W}(k)) - \boldsymbol{J}(\boldsymbol{W}(0))\|_2 \leq \frac{C}{3}\sqrt{\lambda_0}$, and therefore, we have that

$$\|\boldsymbol{J}_{MB}(k) - \boldsymbol{J}_{MB}(0)\|_2 \leq \frac{C\sqrt{\lambda_0}}{3} \leq \frac{\sqrt{\lambda_0}}{3}.$$

On the other hand, based on the inequality $\sigma_{\min}(\mathbf{A} + \mathbf{B}) \geq \sigma_{\min}(\mathbf{A}) - \sigma_{\max}(\mathbf{B})$, where σ denotes singular value, we have

$$\begin{aligned}\sigma_{\min}(\boldsymbol{J}_{MB}(k)) &\geq \sigma_{\min}(\boldsymbol{J}_{MB}(0)) - \sigma_{\min}(\boldsymbol{J}_{MB}(k) - (\boldsymbol{J}_{MB}(k))) \\ &\geq \sigma_{\min}(\boldsymbol{J}_{MB}(0)) - \|\boldsymbol{J}_{MB}(k) - \boldsymbol{J}_{MB}(0)\|_2 \geq \sqrt{\lambda_0} - \frac{\sqrt{\lambda_0}}{3} = \frac{2\sqrt{\lambda_0}}{3}.\end{aligned}$$

Therefore

$$\lambda_{min}(\boldsymbol{G}_{MB}(\boldsymbol{W}(k))) \geq \frac{4\sqrt{\lambda_0}}{9},$$

where $\boldsymbol{G}_{MB}(\boldsymbol{W}(k)) := \boldsymbol{J}_{MB}(\boldsymbol{W}(k))\boldsymbol{J}_{MB}(\boldsymbol{W}(k))^\top$ is the mini-block Gram matrix. We prove Theorem 1 by induction. Assume $||\ \mathrm{u}(\boldsymbol{W}(k)) - \boldsymbol{y}||_2^2 \leq (1-\eta)^k||\ \mathrm{u}(\boldsymbol{W}(0)) -\ \mathrm{y}||_2^2$. One can see that the relationship between the Jacobian $\boldsymbol{J}(\boldsymbol{W}(k))$ and the mini-Block Jacobian $\boldsymbol{J}_{MB}(\boldsymbol{W}(k))$ is:

$$\boldsymbol{J}^\top(\boldsymbol{W}(k)) = \boldsymbol{J}_{MB}(\boldsymbol{W}(k))^\top \boldsymbol{K},$$

where the matrix $\boldsymbol{K} = \underbrace{[I_n, \ldots, I_n]^\top}_{K} \in \mathbb{R}^{Kn\times n}$, I_n is the identity matrix of dimension n, the number of samples, and K is the total number of mini-blocks. We define

$$\begin{aligned}
\boldsymbol{W}_k(s) &= s\boldsymbol{W}(k+1) + (1-s)\boldsymbol{W}(k) \\
&= \boldsymbol{W}(k) - s\frac{\eta}{n}\left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda I\right)^{-1}\boldsymbol{J}(\boldsymbol{W}(k))^\top(\ \mathrm{u}(\boldsymbol{W}(k)) - \boldsymbol{y})) -\ \mathrm{u}(\boldsymbol{W}(k)),
\end{aligned}$$

we have:

$$\begin{aligned}
&\mathrm{u}(\boldsymbol{W}(k+1)) -\ \mathrm{u}(\boldsymbol{W}(k)) \\
&= \boldsymbol{u}(\boldsymbol{W}(k) - \frac{\eta}{n}\left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda I\right)^{-1}\boldsymbol{J}(\boldsymbol{W}(k))^\top(\ \mathrm{u}(\boldsymbol{W}(k)) - \boldsymbol{y})) -\ \mathrm{u}(\boldsymbol{W}(k)) \\
&= -\int_{s=0}^{1}\left\langle\frac{\partial \boldsymbol{u}(\boldsymbol{W}_k(s))}{\partial \boldsymbol{W}^\top}, \frac{\eta}{n}\left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda I\right)^{-1}\boldsymbol{J}(\boldsymbol{W}(k))^\top(\boldsymbol{u}(\boldsymbol{W}(k)) - \boldsymbol{y}))\right\rangle ds \\
&= \underbrace{-\int_{s=0}^{1}\left\langle\frac{\partial \boldsymbol{u}(\boldsymbol{W}(k))}{\partial \boldsymbol{W}^\top}, \frac{\eta}{n}\left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda I\right)^{-1}\boldsymbol{J}(\boldsymbol{W}(k))^\top(\boldsymbol{u}(\boldsymbol{W}(k)) - \boldsymbol{y}))\right\rangle ds}_{\text{(A)}} \\
&+ \underbrace{\int_{s=0}^{1}\left\langle\frac{\partial \boldsymbol{u}(\boldsymbol{W}(k))}{\partial \boldsymbol{W}^\top} - \frac{\partial \boldsymbol{u}(\boldsymbol{W}_k(s))}{\partial \boldsymbol{W}^\top}, \frac{\eta}{n}\left(\boldsymbol{F}_{MB}(\boldsymbol{W}(k)) + \lambda I\right)^{-1}\boldsymbol{J}(\boldsymbol{W}(k))^\top(\boldsymbol{u}(\boldsymbol{W}(k)) - \boldsymbol{y}))\right\rangle ds}_{\text{(B)}}.
\end{aligned}$$

In what follows, to simplify the notation, we drop $\boldsymbol{W}(k)$ whenever the context is clear. Thus, we have

$$\text{(A)} = \frac{\eta}{n}\boldsymbol{J}\left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1}\boldsymbol{J}^{\top}(\boldsymbol{y} - \boldsymbol{u}(k)). \tag{2.3.1}$$

Now, we bound the norm of (B):

$$\begin{aligned}
||\text{(B)}||_2 &\leq \frac{\eta}{n}\left\|\int_{s=0}^{1}\boldsymbol{J}(\boldsymbol{W}_k(s)) - \boldsymbol{J}(\boldsymbol{W}(k))ds\right\|_2 \left\|\left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1}\boldsymbol{J}^{\top}(\boldsymbol{u}(k) - \boldsymbol{y})\right\|_2 \\
&\overset{(1)}{\leq} \frac{\eta 2C}{3n}\lambda_0^{\frac{1}{2}}\left\|\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{F}_{MB} + \lambda I\right)^{-1}\boldsymbol{F}_{MB}^{\top}\boldsymbol{K}(\boldsymbol{u}(k) - \boldsymbol{y}))\right\|_2 \\
&\leq \frac{\eta 2C}{3n}\lambda_0^{\frac{1}{2}}\left\|\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{J}_{MB} + \lambda I\right)^{-1}\boldsymbol{J}_{MB}^{\top}\right\|_2 \|\boldsymbol{K}(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2 \\
&\overset{(2)}{\leq} \frac{\eta C}{3\sqrt{\lambda n}}\sqrt{\lambda_0}\,\|\boldsymbol{K}(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2 \overset{(3)}{=} \frac{\eta C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}}\,\|(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2\,,
\end{aligned} \tag{2.3.2}$$

where in (1) we used Assumption 6.2, which implies

$$\begin{aligned}
\left\|\int_{s=0}^{1}\boldsymbol{J}(\boldsymbol{W}_k(s)) - \boldsymbol{J}(\boldsymbol{W}(k))ds\right\|_2 &\leq \|\boldsymbol{J}(\boldsymbol{W}(k)) - \boldsymbol{J}(\boldsymbol{W}(0))\|_2 + \|\boldsymbol{J}(\boldsymbol{W}(k+1)) - \boldsymbol{J}(\boldsymbol{W}(0))\|_2 \\
&\leq \frac{2C}{3}\sqrt{\lambda_0}.
\end{aligned}$$

The inequality (2) follows from the fact that

$$\begin{aligned}
\left\|\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{J}_{MB} + \lambda I\right)^{-1}\boldsymbol{J}_{MB}^{\top}\right\|_2 &= \sigma_{max}\left(\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{J}_{MB} + \lambda I\right)^{-1}\boldsymbol{J}_{MB}^{\top}\right) \\
&= \sqrt{\lambda_{max}\left(\boldsymbol{J}_{MB}\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{J}_{MB} + \lambda I\right)^{-2}\boldsymbol{J}_{MB}^{\top}\right)},
\end{aligned}$$

and that

$$\lambda_{max}\left(\boldsymbol{J}_{MB}\left(\frac{1}{n}\boldsymbol{J}_{MB}^{\top}\boldsymbol{J}_{MB} + \lambda I\right)^{-2}\boldsymbol{J}_{MB}^{\top}\right) = \max_{\mu\text{ eigenvalue of }\boldsymbol{G}_{MB}}\frac{\mu}{(\frac{\mu}{n} + \lambda)^2} \leq \frac{n\lambda}{(\frac{n\lambda}{n} + \lambda)^2} = \frac{n}{4\lambda}.$$

and in the equality (3), we have used the fact that $\|\boldsymbol{K}(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2 = \sqrt{K}\,\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2$. Finally, we have:

$$
\begin{aligned}
||\boldsymbol{u}(k+1)-\boldsymbol{y}||_2^2 &= ||\boldsymbol{u}(k)-\boldsymbol{y}+\boldsymbol{u}(k+1)-\boldsymbol{u}(k)||_2^2 \\
&= ||\boldsymbol{u}(k)-\boldsymbol{y}||_2^2 - 2\,(\boldsymbol{y}-\boldsymbol{u}(k))^\top\,(\boldsymbol{u}(k+1)-\boldsymbol{u}(k)) + ||\boldsymbol{u}(k+1)-\boldsymbol{u}(k)||_2^2 \\
&\leq ||\boldsymbol{u}(k)-\boldsymbol{y}||_2^2 - \frac{2\eta}{n}\underbrace{(\boldsymbol{y}-\boldsymbol{u}(k))^\top\,\boldsymbol{J}(k)\,(\boldsymbol{F}_{MB}+\lambda I)^{-1}\,\boldsymbol{J}(k)^\top\,(\boldsymbol{y}-\boldsymbol{u}(k))}_{①} \\
&\qquad + \frac{2\eta C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}}\,\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2^2 + \underbrace{||\boldsymbol{u}(k+1)-\boldsymbol{u}(k)||_2^2}_{②} \\
&\leq ||\boldsymbol{u}(k)-\boldsymbol{y}||_2^2 - \frac{2\eta K\lambda_0}{\lambda_0+\frac{9}{4}n\lambda}||\boldsymbol{u}(k)-\boldsymbol{y}||_2^2 \\
&+ \frac{2\eta C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}}\,\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2^2 + \eta^2\left(K+\frac{C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}}\right)^2\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2^2 \\
&\leq (1-\eta)\,\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2^2 \\
&+ \eta\,\|(\boldsymbol{u}(k)-\boldsymbol{y}))\|_2^2\left(\eta\left(K+\frac{C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}}\right)^2 - \left(\frac{2K\lambda_0}{\lambda_0+\frac{9}{4}n\lambda} - \frac{2C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} - 1\right)\right).
\end{aligned}
$$

Part ① is lower bounded as follows:

$$
\begin{aligned}
① &\geq \lambda_{\min}\left(\boldsymbol{J}_{MB}\left(\frac{1}{n}\boldsymbol{J}_{MB}^\top\boldsymbol{J}_{MB}+\lambda I\right)^{-1}\boldsymbol{J}_{MB}^\top\right)\|\boldsymbol{K}(\boldsymbol{u}(k)-\boldsymbol{y})\|_2^2 \\
&= K\lambda_{\min}\left(\boldsymbol{J}_{MB}\left(\frac{1}{n}\boldsymbol{J}_{MB}^\top\boldsymbol{J}_{MB}+\lambda I\right)^{-1}\boldsymbol{J}_{MB}^\top\right)\|\boldsymbol{u}(k)-\boldsymbol{y}\|_2^2 \\
&= nK\|\boldsymbol{u}(k)-\boldsymbol{y}\|_2^2\frac{\lambda_{min}(\boldsymbol{G}_{MB}(k))}{\lambda_{min}(\boldsymbol{G}_{MB}(k))+n\lambda} \\
&\geq \frac{nK\lambda_0}{\lambda_0+\frac{9}{4}n\lambda}\|\boldsymbol{u}(k)-\boldsymbol{y}\|_2^2.
\end{aligned}
$$

Part (2) is upper bounded, on the other hand, using equality (2.3.1) and inequality (2.3.2). More specifically, we have:

$$
\begin{aligned}
||\boldsymbol{u}(k+1) - \boldsymbol{u}(k)||_2 &\leq \frac{\eta}{n} \left\| \boldsymbol{J}(k) \left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1} \boldsymbol{J}(k)^\top (\boldsymbol{y} - \boldsymbol{u}(k))) \right\| + ||\text{(B)}||_2 \\
&\leq \frac{\eta K}{n} \left\| \boldsymbol{J}_{MB}(k) \left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1} \boldsymbol{J}_{MB}(k)^\top \right\| \|(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2 + \frac{\eta C \sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} \|(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2 \\
&\leq \eta \left(K + \frac{C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} \right) \|(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2 \,.
\end{aligned}
$$

The last inequality follows from the fact that if (μ, v) is an (eigenvalue, eigenvector) pair for $\boldsymbol{G}_{MB} = \boldsymbol{J}_{MB}\boldsymbol{J}_{MB}^\top$, then $(\mu, \boldsymbol{J}_{MB}^\top v)$ and $(\frac{1}{\frac{\mu}{n}+\lambda}, \boldsymbol{J}_{MB}^\top v)$ are such pairs for $\boldsymbol{F}_{MB}$ and $(\frac{1}{n}\boldsymbol{F}_{MB} + \lambda I)^{-1}$, respectively, and it follows that

$$
\begin{aligned}
\left\| \boldsymbol{J}_{MB}(k) \left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1} \boldsymbol{J}_{MB}(k)^\top \right\|_2 &= \lambda_{max} \left(\boldsymbol{J}_{MB}(k) \left(\boldsymbol{F}_{MB} + \lambda I\right)^{-1} \boldsymbol{J}_{MB}(k)^\top \right) \\
&= \max_{\mu \text{ eigenvalue of } \boldsymbol{G}_{MB}(k)} \frac{n\mu}{\mu + n\lambda} \leq n.
\end{aligned}
$$

Let us consider the function $\lambda \xrightarrow{f} f(\lambda) := \left(\frac{2K\lambda_0}{\lambda_0 + \frac{9}{4}n\lambda} - \frac{2C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} - 1 \right)$. We have that

$$
f(\frac{4\lambda_0}{9n}) = K - C\sqrt{K} - 1 \geq K - \frac{1}{2}\sqrt{K} - 1 > 0 \quad \text{for } K \geq 3.
$$

Therefore by continuity of the function $f(.)$, there exists an interval $[\underline{\lambda}, \overline{\lambda}]$, such as $\frac{4\lambda_0}{9n} \in [\underline{\lambda}, \overline{\lambda}]$, and for all damping values λ in $[\underline{\lambda}, \overline{\lambda}]$, the function $f(.)$ is positive. For such a choice of damping value λ (for example $\lambda = \frac{4\lambda_0}{9n}$), and for a small enough learning rate, i.e:

$$
\eta \leq \frac{\frac{2K\lambda_0}{\lambda_0 + \frac{9}{4}n\lambda} - \frac{2C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} - 1}{\left(K + \frac{C\sqrt{\lambda_0 K}}{3\sqrt{\lambda n}} \right)^2} := \eta_\lambda.
$$

We Hence, we get that

$$||\boldsymbol{u}(k+1) - \boldsymbol{y}||_2^2 \leq (1-\eta)\, \|(\boldsymbol{u}(k) - \boldsymbol{y}))\|_2^2 ,$$

which concludes the proof. □

2.4 Implementation details of MBF and comparison on complexity

Mini-batch averages, Exponentially decaying averages and Momentum: Because the size of training data sets is usually large, we use mini-batches to estimate the quantities needed for MBF. We use $\overline{X}$ to denote the average value of X over a mini-batch for any quantity X. Moreover, for the EFM mini-blocks, we use moving averages to both reduce the stochasticity and incorporate more information from the past, more specifically, we use a moving average scheme to get a better estimate of the EFM mini-blocks, i.e.$\widehat{G_{W_{l,b}}} = \beta\widehat{G_{W_{l,b}}} + (1-\beta)\overline{G_{W_{l,b}}}$, where $\overline{G_{W_{l,b}}}$ is the current approximation to the mini-block EFM defined below. In order to bring MBF closer to a drop-in replacement for adaptive gradient methods such as Adam, we add momentum to the mini-batch gradient, let: $\widehat{\mathcal{D}W_l} = \mu\widehat{\mathcal{D}W_l} + \overline{\mathcal{D}W_l}$ and then apply the preconditioner to $\widehat{\mathcal{D}W_l}$ to compute the step.

Approximating the mini-block Fisher matrices: As mentioned previously, computing the matrices $\overline{G_{W_{l,b}}} := \frac{1}{m}(\boldsymbol{J}^{W_{l,b}})^T\boldsymbol{J}^{W_{l,b}}$ to update the EFM mini-blocks is inefficient in practice as this requires storing and computing the individual gradients. Hence, we approximate these mini-block matrices by the outer product of the part of the mini-batch gradient corresponding to the subset of weights $W_{l,b}$, i.e., $\overline{G_{W_{l,b}}} \approx (\overline{\mathcal{D}W_{l,b}})(\overline{\mathcal{D}W_{l,b}})^\top$.

Spacial average for large fully-connected layers: In some CNN and autoencoder models, using the EFM mini-blocks can still be computationally prohibitive for fc layers, where both the input and output dimensions are large. Therefore, for such layers we used a Spatial Averaging technique, similar to one used in [67], where we maintained a single preconditioning matrix for all the mini-blocks by averaging the approximate mini-block EFM matrices whenever we updated the preconditioning matrix. This technique also leads to more stable curvature updates as a side benefit,

as observed for the method proposed in [67], where the Hessian diagonal was "smoothed" across each layer. We also explored using spacial averaging for convolutional layers. However since the kernel-wise mini-blocks are small in size, spacial averaging does not compare favorably to the full MBF method (see section 2.5.4.6).

Amortized updates of the preconditioning matrices: The extra work for the above computations, as well as for updating the inverses $F_{W_{l,j}}^{-1}$ compared with first-order methods is amortized by only performing the Fisher matrix updates every T_1 iterations and computing their inverses every T_2 iterations. This approach which is also used in KFAC and Shampoo, does not seem to degrade MBF's overall performance, in terms of computational speed(see section 2.5.4.7 for empirical evidence).

Comparison of Memory and Per-iteration Time Complexity: In Table 2.1, we compare the space and computational requirements of the proposed MBF method with KFAC, Shampoo and Adam, which are among the predominant 2nd and 1st-order methods used to train DNNs. For one convolutional layer, with J input channels, I output channels, kernel size $|\Delta| = (2R+1)^2$, and $|\mathcal{T}|$ spacial locations. Let m denote the size of the mini batches, and T_1 and T_2 denote, respectively, the frequency for updating the preconditioners and inverting them for KFAC, Shampoo and MBF. As indicated in Table 2.1, the amount of memory required by MBF is the same order of magnitude as that required by Adam, (specifically, more by a factor of $|\Delta|$, which is usually small in most CNN architectures; e.g, in VGG16 [68] $|\Delta| = 9$) and less than KFAC, Shampoo and other SOTA Kronecker-factored preconditioners, (specifically, e.g., by a factor of $O\left(J + \frac{I}{|\Delta|}\right)$ for KFAC.

Table 2.1: Computation and Storage Requirements per iteration for convolutional layer.

Algorithm	Additional pass	Curvature	Step ΔW_l	Storage P_l
MBF	—	$O(IJ(\frac{\|\Delta\|^2}{T_1} + \frac{\|\Delta\|^3}{T_2}))$	$O(IJ\|\Delta\|^2)$	$O(IJ\|\Delta\|^2)$
Shampoo	—	$O(\frac{(J^2+\|\Delta\|^2+I^2)}{T_1} + \frac{J^3+I^3+\|\Delta\|^3}{T_2})$	$O((I+J+\|\Delta\|)IJ\|\Delta\|)$	$O(I^2+J^2+\|\Delta\|^2)$
KFAC	$O(\frac{mIJ\|\Delta\|\|\mathcal{T}\|}{T_1})$	$O(\frac{m(J^2\|\Delta\|^2+I^2)\|\mathcal{T}\|}{T_1} + \frac{J^3\|\Delta\|^3+I^3}{T_2})$	$O(IJ^2\|\Delta\|^2 + I^2J\|\Delta\|)$	$O(J^2\|\Delta\|^2 + I^2)$
Adam	—	$O(IJ\|\Delta\|)$	$O(IJ\|\Delta\|)$	$O(IJ\|\Delta\|)$

We indicate in Table 2.1, in gray, the portion of the computational complexity for both the curvature and step computations that can benefit from GPU broadcasting and parallelism. Since MBF maintains mini-block curvature matrices of the same size, its **effective** computational complexity is $O\left(\frac{|\Delta|^2}{T_1} + \frac{|\Delta|^3}{T_2} + |\Delta|^2\right)$, which is of modest magnitude as it is a function of only the **kernel-size** Δ, which is small in most CNN architectures. Note that in our experiments, $T_1 \approx |\Delta|$ and $T_2 \approx |\Delta|^2$. The computational and storage requirements for a fully connected layer, with d_i inputs and d_o outputs, are given in Tables 2.2 and 2.3.

Table 2.2: Storage Requirements for fully connected layer

Algorithm	$\mathcal{D}W$	P_l
MBF	$O(d_i d_o)$	$O(d_i^2)$
KFAC	$O(d_i d_o)$	$O(d_i^2 + d_o^2 + d_i d_o)$
Shampoo	$O(d_i d_o)$	$O(d_i^2 + d_o^2)$
Adam	$O(d_i d_o)$	$O(d_i d_o)$

Table 2.3: Computation per iteration beyond that required for the minibatch stochastic gradient for fully connected layer

Algorithm	Additional pass	Curvature	Step ΔW_l
MBF	—	$O(\frac{d_o d_i^2}{T_1} + \frac{d_o d_i^3}{T_2})$	$O(d_o d_i^2)$
KFAC	$O(\frac{m d_i d_o}{T_1})$	$O(\frac{m d_i^2 + m d_o^2}{T_1} + \frac{d_i^3 + d_o^3}{T_2})$	$O(d_i^2 d_o + d_o^2 d_i)$
Shampoo	—	$O(\frac{d_i^2 + d_o^2}{T_1} + \frac{d_i^3 + d_o^3}{T_2})$	$O((d_i + d_o) d_i d_o)$
Adam	—	$O(d_i d_o)$	$O(d_i d_o)$

A pseudo-code that fully describes our MBF algorithm is given in Algorithm 2.2. For the parameters in the BN layers, we used the direction used in Adam, which is equivalent to using mini-blocks of size 1, dividing each gradient component by that block's square root. We did a warm start to estimate the pre-conditioning mini-block matrices in an initialization step that iterated through the whole data set and adopted a moving average scheme to update them with $\beta = 0.9$.

Algorithm 2.2: Mini-Block Fisher method (MBF)

Input: Given batch size m, learning rate $\{\eta_k\}_{k\geq 1}$, weight decay factor γ, damping value λ, statistics update frequency T_1, inverse update frequency T_2

$\mu = 0.9$, $\beta = 0.9$

Initialize $\widehat{G_{l,b}} = \mathbb{E}[G_{l,b}]$ ($l = 1, .., k$, mini-blocks b) by iterating through the whole dataset, $\widehat{\mathcal{D}W_{l,b}} = 0$ ($l = 1, .., k$, mini-blocks b)

for $k = 1, 2, \ldots$ **do**

  Sample mini-batch M_t of size m

  Perform a forward-backward pass over M_t to compute the mini-batch gradient $\overline{\mathcal{D}W_{l,b}}$

  for $l = 1, ...L$ **do**

    for *mini-block b in layer l,* ***in parallel*** **do**

      $\widehat{\mathcal{D}W_{l,b}} = \mu\widehat{\mathcal{D}W_{l,b}} + \overline{\mathcal{D}W_{l,b}}$

      if $k \equiv 0 \pmod{T_1}$ **then**

        If Layer l is convolutional: $\widehat{G_{l,j,i}} = \beta\widehat{G_{l,j,i}} + (1-\beta)\overline{\mathcal{D}W_{l,j,i}}\left(\overline{\mathcal{D}W_{l,j,i}}\right)^\top$

        If Layer l is fully-connected: $\widehat{G_l} = \beta\widehat{G_l} + \frac{1-\beta}{O}\sum_{j=1}^{O}\overline{\mathcal{D}W_{l,j}}\left(\overline{\mathcal{D}W_{l,j}}\right)^\top$

      if $k \equiv 0 \pmod{T_2}$ **then**

        Recompute and store $(\widehat{G_{l,b}} + \lambda I)^{-1}$

      $p_{l,b} = (\widehat{G_{l,b}} + \lambda I)^{-1}\widehat{\mathcal{D}W_{l,b}} + \gamma W_{l,b}$

      $W_{l,b} = W_{l,b} - \eta_k p_{l,b}$

2.5 Numerical Experiments

In this section, we compare MBF with some SOTA first-order (SGD-m, Adam) and second-order (KFAC, Shampoo) methods. Since MBF uses information about the second-moment of the gradient to construct a preconditioning matrix, Adam, KFAC and Shampoo were obvious choices for comparison with MBF. We used the most popular version of Adam, AdamW [69] as a representative of adaptive first-order methods. An extensive study in [12] of more than 100 optimization methods, 65 of which have “Adam” or “Ada” as part of their names, concluded that no method was "clearly dominating across all tested tasks and that ADAM remains a strong contender, with newer methods failing to significantly and consistently outperform it". We also include in section 2.5.4.8 additional results that include Adabelief and Adagrad.

Our experiments were run on a machine with one V100 GPU and eight Xeon Gold 6248 CPUs using PyTorch [70]. Each algorithm was run using the best hyper-parameters, determined by a grid

search (specified in sections 2.5.4.2 and 2.5.4.1), and 5 different random seeds. The performance of MBF and the comparison algorithms are plotted in Figures 2.6 and 2.7: the solid curves depict the results averaged over the 5 different runs, and the shaded areas depict the ±standard deviation range for these runs.

2.5.1 Description of Competing Algorithms

SGD-m: In SGD with momentum, we updated the momentum m_t of the gradient using the recurrence

$$m_t = \mu \cdot m_{t-1} + g_t$$

at every iteration, where g_t denotes the mini-batch gradient at current iteration and $\mu = 0.9$. The gradient momentum is also used in the second-order methods, in our implementations. For the CNN problems, we used weight decay with SGD-m, as it is used in SGDW in [69].

Adam: For Adam, we followed exactly the algorithm in [11] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, updating the momentum of the gradient at every iteration by the recurrence

$$m_t = \beta_1 \cdot mt - 1 + (1 - \beta_1) \cdot g_t.$$

The role of β_1 and β_2 is similar to that of μ and β in Algorithms 2.2 and 2.3, as we will describe below. For the CNN problems, we used weight decay with Adam, as it is used in AdamW in [69].

Shampoo: We implemented Shampoo as described below in Algorithm 2.3 following the description given in [40], and include major improvements, following the suggestions in [63]. These improvements are (i) using a moving average to update the estimates $\widehat{G_l^{(i)}}$ and (ii) using a coupled Newton method to compute inverse roots of the preconditioning matrices,

KFAC: In our implementation of KFAC, the preconditioning matrices that we used for linear layers and convolutional layers are precisely those described in [33] and [34], respectively. For the parameters in the BN layers, we used the gradient direction, exactly as in `https://github.com/alecwangcq/KFAC-Pytorch`. We did a warm start to estimate the pre-

Algorithm 2.3: Shampoo

Input: Given batch size m, learning rate $\{\eta_k\}_{k\geq 1}$, weight decay factor γ, damping value ϵ, statistics update frequency T_1, inverse update frequency T_2

$\mu = 0.9$, $\beta = 0.9$

Initialize $\widehat{G_l^{(i)}} = \mathbb{E}[G_l^{(i)}]$ $(l = 1, .., k, i = 1, ..., k_l)$ by iterating through the whole dataset,

$\widehat{\nabla_{W_l}\mathcal{L}} = 0$ $(l = 1, ..., L)$ **for** $k = 1, 2, \ldots$ **do**

  Sample mini-batch M_k of size m

  Perform a forward-backward pass over the current mini-batch M_k to compute the minibatch gradient $\overline{\nabla \mathcal{L}}$ **for** $l = 1, ...L$ **do**

    $\widehat{\nabla_{W_l}\mathcal{L}} = \mu\widehat{\nabla_{W_l}\mathcal{L}} + \overline{\nabla_{W_l}\mathcal{L}}$ **if** $k \equiv 0 \pmod{T_1}$ **then**

      Update $\widehat{G_l^{(i)}} = \beta\widehat{G_l^{(i)}} + (1-\beta)\overline{G_l}^{(i)}$ for $i = 1, ..., k_l$ where $\overline{G_l} = \overline{\nabla_{W_l}\mathcal{L}}$

    if $k \equiv 0 \pmod{T_2}$ **then**

      Recompute $\left(\widehat{G_l^{(1)}} + \epsilon I\right)^{-1/2k_l}, ..., \left(\widehat{G_l^{(k_l)}} + \epsilon I\right)^{-1/2k_l}$ with the coupled Newton method

    $p_l = \widehat{\nabla_{W_l}\mathcal{L}} \times_1 \left(\widehat{G_l^{(1)}} + \epsilon I\right)^{-1/2k_l} \times_2 \cdots \times_k \left(\widehat{G_l^{(k_l)}} + \epsilon I\right)^{-1/2k_l}$ $p_l = p_l + \gamma W_l$

    $W_l = W_l - \eta_k \cdot p_l$

conditioning KFAC matrices in an initialization step that iterated through the whole data set, and adopted a moving average scheme to update them with $\beta = 0.9$ afterwards. As in the implementation described in [71], for autoencoder experiments, we inverted the damped KFAC matrices and used them to compute the updating direction, where the damping factors for both A and G were set to be $\sqrt{\lambda}$, where λ is the overall damping value; and for the CNN experiments, we employed the SVD (i.e. eigenvalue decomposition) implementation suggested in `https://github.com/alecwangcq/KFAC-Pytorch`, which, as we verified, performs better than splitting the damping value and inverting the damped KFAC matrices (as suggested in [33, 34]). Further, for the CNN problems, we implemented weight decay exactly as in MBF (Algorithm 2.2) and Shampoo (Algorithm 2.3).

2.5.2 Generalization performance, CNN problems

We first compared the generalization performance of MBF to SGD-m, Adam, KFAC and Shampoo on three CNN models, namely, ResNet32 [72], VGG16 [68] and VGG11 [68], respectively,

on the datasets CIFAR-10, CIFAR-100 and SVHN [73]. The first two have 50,000 training data and 10,000 testing data (used as the validation set in our experiments), while SVHN has 73,257 training data and 26,032 testing data. For all algorithms, we used a batch size of 128. In training, we applied data augmentation as described in [74], including random horizontal flip and random crop, since these setting choices have been used and endorsed in many previous research papers, e.g. [75, 76, 41]. (see section 2.5.4.2 for more details about the experimental set-up)

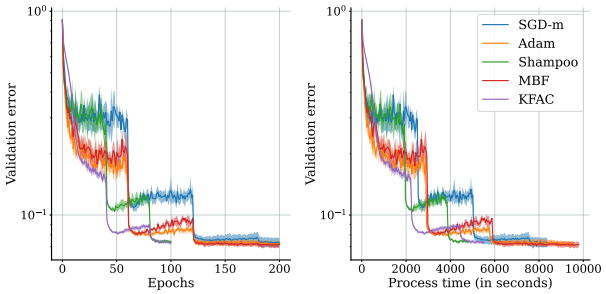
On all three model/dataset problems, the first-order methods were run for 200 epochs, and KFAC and Shampoo for 100 epochs, while MBF was run for 150 epochs on VGG16/CIFAR-100 and VGG11/SVHN, and 200 epochs on ResNet32/CIFAR-10. The reason that we ran MBF for 200 epochs (i.e., the same number as run for Adam) on ResNet32 was because all of ResNet32's convolutional layers use small (3×3) kernels, and it contains just one fully connected layer of modest size $(I, O) = (64, 10)$. Hence as we expected, MBF and Adam took almost the same time to complete 200 epochs. As can be seen in Figure 2.6, MBF could have been terminated after 150 epochs, without a significant change in validation error. On the other hand, since VGG16 and VGG11 have two large fully connected-layers (e.g [4096, 4096, 10/100]), MBF's per-iteration computational cost is substantially larger than Adam's due to these layers. Consequently, for both methods to finish roughly in the same amount of time, we ran MBF for only 150 epochs.

All methods employed a learning rate (LR) schedule that decayed LR by a factor of 0.1 every K epochs, where K was set to 60, 50 and 40 , for the first-order methods, MBF, and KFAC/Shampoo, respectively, on the VGG16 and VGG11 problems, and set to 80, 60 and 40, respectively, on the ResNet32 problem

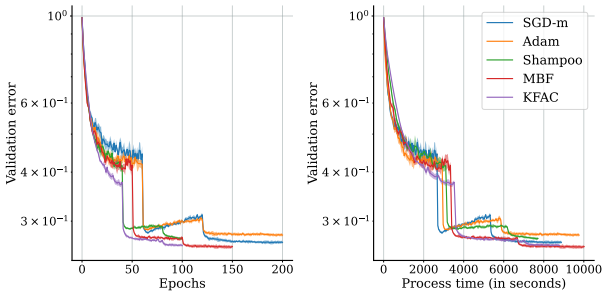
Moreover, weight decay, which has been shown to improve generalization across different optimizers [69, 75], was employed by all of the algorithms, and a grid search on the weight decay factor and the initial learning rate based on the criteria of maximal validation classification accuracy was performed. Finally, the damping parameter was set to 1e-8 for Adam (following common practice), and 0.03 for KFAC (`https://github.com/alecwangcq/KFAC-Pytorch`). For

Shampoo, we set $\epsilon = 0.01$. For MBF, we set $\lambda = 0.003$. We set $T_1 = 10$ and $T_2 = 100$ for KFAC, Shampoo and MBF.

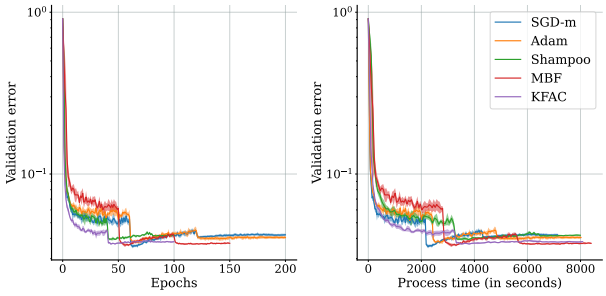
From Figure 2.6, we see that MBF has a similar (and sometimes better) generalization performance than the other methods. Moreover, in terms of process time, MBF is roughly as fast as SGD-m and Adam on ResNet32/CIFAR-10 in Figure 2.6, and is competitive with all of the SOTA first and second-order methods in our experiments.



(a) CIFAR-10, ResNet-32



(b) CIFAR-100, VGG16



(c) SVHN, VGG11

Figure 2.6: Generalization ability of MBF, KFAC, Shampoo, Adam, and SGD-m on three CNN problems.

2.5.3 Optimization Performance, Autoencoder Problems

We also compared the optimization performance of the algorithms on three autoencoder problems [77] with datasets MNIST [65], FACES, and CURVES, which were also used for benchmarking algorithms in [26, 33, 39, 38]. The details of the layer shapes of the autoencoders are specified in section 2.5.4.1. For all algorithms, we used a batch size of 1,000, and settings that largely mimic the settings in the latter papers. Each algorithm was run for 500 seconds for MNIST and CURVES, and 2000 seconds for FACES.

For each algorithm, we conducted a grid search on the LR and damping value based on minimizing the training loss. We set the Fisher matrix update frequency $T_1 = 1$ and inverse update frequency $T_2 = 20$ for second-order methods, as in [41]. From Figure 2.7, it is clear that MBF outperformed SGD-m and Adam, both in terms of per-epoch progress and process time. Moreover, MBF performed (at least) as well as KFAC and Shampoo. We postulate that the performance of MBF is due to its ability to capture important curvature information from the mini-block Fisher matrix, while keeping the computational cost per iteration low and close to that of Adam.

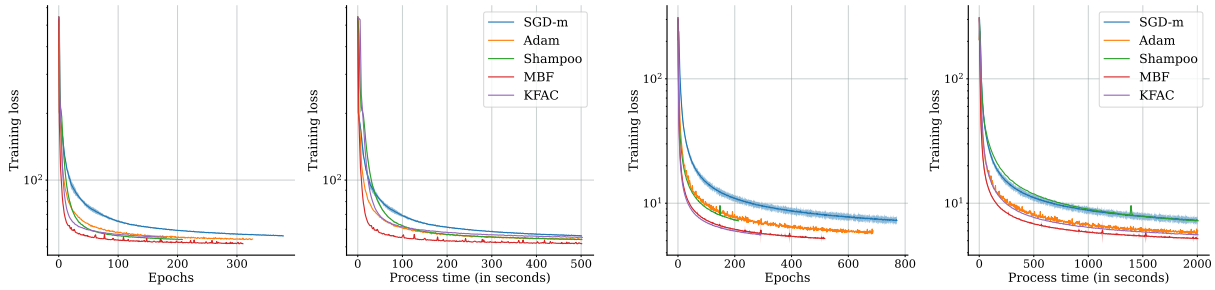
2.5.4 Additional Numerical Experiments and Details

2.5.4 Experiment Settings for the Autoencoder Problems

Table 2.4 describes the model architectures of the autoencoder problems. The activation functions of the hidden layers are always ReLU, except that there is no activation for the very middle layer.

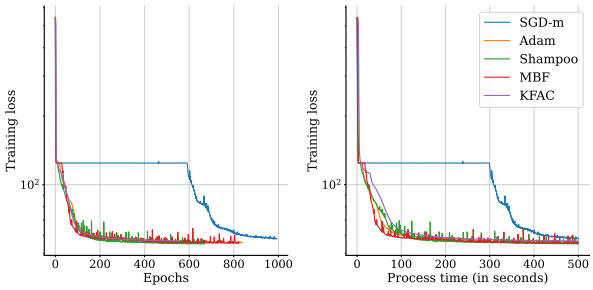
Table 2.4: DNN architectures for the MLP autoencoder problems

	Layer width
MNIST	[784, 1000, 500, 250, 30, 250, 500, 1000, 784]
FACES	[625, 2000, 1000, 500, 30, 500, 1000, 2000, 625]
CURVES	[784, 400, 200, 100, 50, 25, 6, 25, 50, 100, 200, 400, 784]



(a) MNIST autoencoder

(b) FACES autoencoder



(c) CURVES autoencoder

Figure 2.7: Optimization performance of MBF, KFAC, Shampoo, Adam, and SGD-m on three autoencoder problems.

MNIST[1], FACES[2], and CURVES[3] contain 60,000, 103,500, and 20,000 training samples, respectively, which we used in our experiment to train the models and compute the training losses.

We used binary entropy loss (with sigmoid) for MNIST and CURVES, and squared error loss for FACES. The above settings largely mimic the settings in [26, 33, 39, 41]. Since we primarily focused on optimization rather than generalization in these tasks, we did not include L_2 regularization or weight decay.

In order to obtain Figure 2.7, we first conducted a grid search on the learning rate (lr) and damping value based on the criteria of minimizing the training loss. The ranges of the grid searches used for the algorithms in our tests are specified in Table 2.5.

The best hyper-parameter values determined by our grid searches are listed in Table 2.4.

[1] `http://yann.lecun.com/exdb/mnist/`
[2] `http://www.cs.toronto.edu/~jmartens/newfaces_rot_single.mat`
[3] `http://www.cs.toronto.edu/~jmartens/digs3pts_1.mat`

Table 2.5: Grid of hyper-parameters for autoencoder problems

Algorithm	learning rate	damping λ
SGD-m	1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2	damping: not applicable
Adam	1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2	1e-8, 1e-4, 1e-2
Shampoo	1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3	1e-4, 3e-4, 1e-3, 3e-3, 1e-2
MBF	1e-7, 3e-7, 1e-6, 3e-6, 1e-5, 3e-5, 1e-4	1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2
KFAC	1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2, 1e-2, 3e-2	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1

Table 2.6: Hyper-parameters (learning rate, damping) used to produce Figure 2.7

Name	MNIST	FACES	CURVES
MBF	(1e-5, 3e-4) $\rightarrow$ 51.49	(1e-6, 3e-3) $\rightarrow$ 5.17	(1e-5, 3e-4) $\rightarrow$ 55.14
KFAC	(3e-3, 3e-1) $\rightarrow$ 53.56	(1e-1, 1e1) $\rightarrow$ 5.55	(1e-2, 1e0) $\rightarrow$ 56.47
Shampoo	(3e-4, 3e-4) $\rightarrow$ 53.80	(3e-4, 3e-4) $\rightarrow$ 7.21	(1e-3, 3e-3) $\rightarrow$ 54.86
Adam	(3e-4, 1e-4) $\rightarrow$ 53.67	(1e-4, 1e-4) $\rightarrow$ 5.55	(3e-4, 1e-4) $\rightarrow$ 55.23
SGD-m	(3e-3, -) $\rightarrow$ 55.63	(1e-3, -) $\rightarrow$ 7.08	(1e-2, -) $\rightarrow$ 55.49

2.5.4 Experiment Settings for the CNN Problems

The ResNet32 model refers to the one in Table 6 of [72], whereas the VGG16 model refers to model D of [68], with the modification that batch normalization layers were added after all of the convolutional layers in the model. For all algorithms, we used a batch size of 128 at every iteration.

We used weight decay for all the algorithms that we tested, which is related to, but not the same as L_2 regularization added to the loss function, and has been shown to help improve generalization performance across different optimizers [69, 75]. The use of weight decay for MBF and Shampoo is implemented in lines 16 and 17 in Algorithm 2.2 and in lines 15 and 16 in Algorithm 2.3, respectively, and is similarly applied to SGD-m, Adam, and KFAC.

For MBF, we set $\lambda = 0.003$. We also tried values around 0.003 and the results were not sensitive to the value of λ. Hence, λ can be set to 0.003 as a default value. For KFAC, we set the overall damping value to be 0.03, as suggested in the implementation in `https://github.com/alecwangcq/KFAC-Pytorch`. We also tried values around 0.03 for KFAC and confirmed that 0.03 is a good default value.

In order to obtain Figure 2.6, we first conducted a grid search on the initial learning rate (lr) and weight decay (wd) factor based on the criteria of maximizing the classification accuracy on the validation set. The range of the grid searches for the algorithms in our tests are specified in Table 2.7.

Table 2.7: Grid of hyper-parameters for CNN problems

Algorithm	learning rate	weight decay γ
SGD-m	3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2, 1e-1, 3e-1, 1e0	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1
Adam	1e-6, 3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1
Shampoo	3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2, 1e-1	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1
MBF	1e-6, 3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1
KFAC	3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2, 3e-2	1e-2, 3e-2, 1e-1, 3e-1, 1e0, 3e0, 1e1

The best hyper-parameter values, and the validation classification accuracy obtained using them, are listed in Table 2.8.

Table 2.8: Hyper-parameters (initial learning rate, weight decay factor) used to produce Figure 2.6 and the average validation accuracy across 5 runs with different random seeds shown in Figure 2.6

Name	CIFAR-10 + ResNet32	CIFAR-100 + VGG16	SVHN + VGG11
MBF	(1e-4, 3e0) $\rightarrow$ 93.42%	(3e-5, 1e1) $\rightarrow$ 74.80%	(1e-3, 3e-1) $\rightarrow$ 96.59%
KFAC	(3e-3, 1e-1) $\rightarrow$ 93.02%	(1e-3, 3e-1) $\rightarrow$ 74.38%	(3e-3, 1e-1) $\rightarrow$ 96.37%
Shampoo	(1e-2, 1e-1) $\rightarrow$ 92.97%	(1e-3, 3e-1) $\rightarrow$ 73.37%	(3e-3, 1e-1) $\rightarrow$ 96.15%
Adam	(3e-3, 1e-1) $\rightarrow$ 93.34%	(3e-5, 1e1) $\rightarrow$ 72.95%	(3e-4, 1e0) $\rightarrow$ 96.34%
SGD-m	(1e-1, 1e-2) $\rightarrow$ 93.23%	(3e-2, 1e-2) $\rightarrow$ 73.99%	(3e-2, 1e-2) $\rightarrow$ 96.63%

2.5.4 Graph Convolutional Networks (GCN) Problems

In this section, we compare the performance of the optimizations algorithms on a 3-layer GCN for the task of node classification in graphs applied to three citation datasets, Cora, CiteSeer, and PubMed(see [78]). In Table 3.1, nodes and edges correspond to documents and citation links, respectively, for these datasets. A sparse feature vector of document keywords, and a class label are associated with each node. For our experiments, as in [79], for each dataset we used all of the nodes for training, except for 1000 nodes that were reserved for testing.

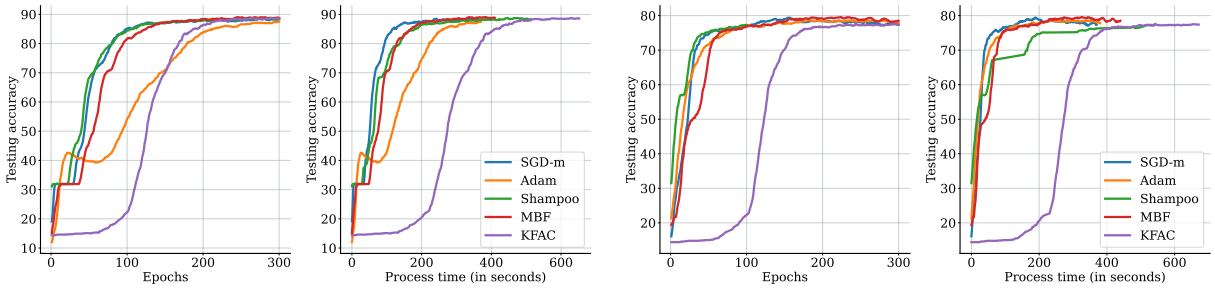
Table 2.9: Citation network datasets statistics

Dataset	Nodes	Edges	Classes	Features
Citeseer	3,327	4732	6	3,703
Cora	2,708	5,429	7	1,433
Pubmed	19,717	44,338	3	500

In our experiments, we used a 3-layer GCN with the following node-sizes $[I, 128, 64, O]$, where I and O are the numbers of input features and classes, respectively. In the first and second layers of this GCN, the activation function ReLU was followed by a dropout function with a rate of 0.5. The loss function was evaluated as the negative log-likelihood of Softmax of the last layer. The weights of parameters were initialized as in [80] and input vectors were row-normalized as in [81]. The models were trained for 300 epochs on the Cora and Citeseer datasets and 500 epochs on the Pubmed dataset. The hyperparameter search space was the same as that used for the CNN problems with no LR schedule. For MBF, spatial averaging was only used in the first layer to mitigate the memory and computational burden in that layer. We set the Fisher matrix update frequency $T_1 = 1$ and the inverse update frequency $T_2 = 25$ for all second-order methods. The optimization performance was measured by the test accuracy. From Figure 2.8, we see that MBF had better final generalization performance than the other methods and, in terms of process time, MBF was roughly as fast as SGD-m and Adam on Cora and Citeseer, and was competitive with all of the SOTA first and second-order methods.

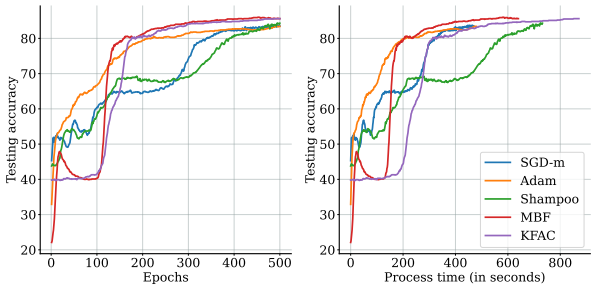
2.5.4 Details on the Cosine similarity experiment:

We provide in Algorithm 2.4 the full implementation of MBF-True for completeness. Note that, in MBF-True, the only difference between it and MBF is that we are using the mini-batch gradient $\overline{\mathcal{D}_2 W_{l,b}}$ (denoted by $\mathcal{D}_2$) of the model on sampled labels y_t from the model's distribution (see lines 10-13 in Algorithm 2.4) to update the estimate of mini-block preconditioners, using a moving average (lines 12, 13), with a rank one outer-product, which is different from computing the true Fisher for that mini-block.



(a) Cora GCN

(b) Citeseer GCN



(c) Pubmed GCN

Figure 2.8: Generalization performance of MBF, KFAC, Shampoo, Adam, and SGD-m on three GCN problems.

We explored how close MBF's direction is to the one obtained by a block-diagonal full EFM method (that we call BDF). We provide here a detailed implementation of the procedure that we used for completeness. More specifically, for any algorithm X, we reported the cosine similarity between the direction given by X and that obtained by BDF in the procedure described in Algorithm 2.5.

The algorithms were run on a 16×16 down-scaled MNIST [65] dataset and a small feed-forward NN with layer widths 256-20-20-20-20-20-10 described in [33]. For all methods, we followed the trajectory obtained using the BDF method as described in Algorithm 2.5.

2.5.4 Comparison between MBF and MBF-True on Autoencoder and CNN problems

The cosine similarity results reported in Figure 2.5 on the down-scaled MNIST suggest that the direction obtained by MBF and MBF-True behave similarly with respect the direction obtained by BDF. In this section, we compare the performance of MBF-True to MBF on the same Autoencoder

Algorithm 2.4: MBF-True

Input: Given batch size m, learning rate $\{\eta_k\}_{k\geq 1}$, weight decay factor γ, damping value λ, statistics update frequency T_1, inverse update frequency T_2

$\mu = 0.9$, $\beta = 0.9$

Initialize $\widehat{G_{l,b}} = \mathbb{E}[G_{l,b}]$ ($l = 1, .., k$, mini-blocks b) by iterating through the whole dataset, $\widehat{\mathcal{D}W_{l,b}} = 0$ ($l = 1, .., k$, mini-blocks b) **for** $k = 1, 2, \ldots$ **do**

  Sample mini-batch M_t of size m

  Perform a forward-backward pass over M_t to compute the mini-batch gradient $\overline{\mathcal{D}W_{l,b}}$

  for $l = 1, ...L$ **do**

    for *mini-block b in layer l,* ***in parallel*** **do**

      $\widehat{\mathcal{D}W_{l,b}} = \mu\widehat{\mathcal{D}W_{l,b}} + \overline{\mathcal{D}W_{l,b}}$ **if** $k \equiv 0 \pmod{T_1}$ **then**

        Sample the labels y_t from the model's distribution

        Perform a backward pass over y_t to compute the mini-batch gradients $\overline{\mathcal{D}_2W_{l,b}}$

        If Layer l is convolutional: $\widehat{G_{l,j,i}} = \beta\widehat{G_{l,j,i}} + (1-\beta)\overline{\mathcal{D}_2W_{l,j,i}}\left(\overline{\mathcal{D}_2W_{l,j,i}}\right)^\top$

        If Layer l is fully-connected: $\widehat{G_l} = \beta\widehat{G_l} + \frac{1-\beta}{O}\sum_{j=1}^{O}\overline{\mathcal{D}_2W_{l,j}}\left(\overline{\mathcal{D}_2W_{l,j}}\right)^\top$

      if $k \equiv 0 \pmod{T_2}$ **then**

        Recompute and store $(\widehat{G_{l,b}} + \lambda I)^{-1}$

      $p_{l,b} = (\widehat{G_{l,b}} + \lambda I)^{-1}\widehat{\mathcal{D}W_{l,b}} + \gamma W_{l,b}$

      $W_{l,b} = W_{l,b} - \eta_k p_{l,b}$

Algorithm 2.5: Cosine(BDF, Algorithm X)

Input: All required parameters for Algorithm X $m = 1000, \eta = 0.01, \mu = 0.9$, $\beta = 0.9, \lambda = 0.01$

Initialize the block EFM matrices $\widehat{F_l} = \mathbb{E}[F_l]$ ($l = 1, .., L$) by iterating through the whole dataset $\widehat{\mathcal{D}W_l} = 0$ ($l = 1, .., L$) **for** $k = 1, 2, \ldots$ **do**

  Sample mini-batch M_t of size m

  Perform a forward-backward pass over M_t to compute the mini-batch gradient $\overline{\mathcal{D}W_l}$ **for** $l = 1, ...L$ **do**

    $\widehat{\mathcal{D}W_l} = \mu\widehat{\mathcal{D}W_l} + \overline{\mathcal{D}W_l}$

    $\widehat{F_l} = \beta\widehat{F_l} + (1-\beta)\mathbb{E}[F_l]$

    $p_l = (\widehat{F_l} + \lambda I)^{-1}\widehat{\mathcal{D}W_{l,b}}$

    Compute the direction d_l given by algorithm X at the current iterate W_l

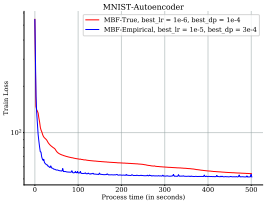
    Compute and store the cosine $\frac{|p_l^T d_l|}{\|p_l\|\|d_l\|}$

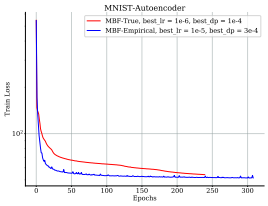
    $W_l = W_l - \eta p_l$

problems (MNIST, FACES, CURVES) described in 2.5.4.1 and the same CNN problems (CIFAR-10 + ResNet32, CIFAR-100 + VGG16, and SVHN + VGG11) described in 2.5.4.2. We used the same

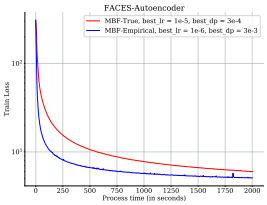
grid of parameters to tune MBF-True as the one described in 2.5.4.1 and 2.5.4.2. We report in Figures 2.9 and 2.10 the training and validation errors obtained on these problems, as well as the best hyper-parameters for both methods in the legends. It seems that using the symmetric outer product of the empirical mini-batch gradient to update the mini-block preconditioner yields better results than using the mini-batch gradient from sampled data from the model's distribution to compute this inner product.

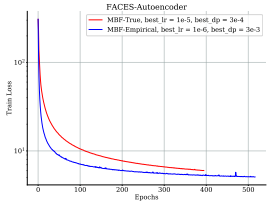
We think this might be the case because MBF is closer to being an adaptive gradient methods, which also use the empirical gradient such as ADAGRAD and ADAM, rather than a second-order natural gradient method such as KFAC, where in the latter case using a sampled gradient yields better results than using the empirical data. Note that, when the mini-block sizes are 1, MBF becomes a diagonal preconditioning method like ADAM minus the square root operation.



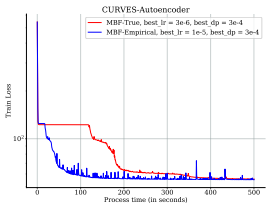


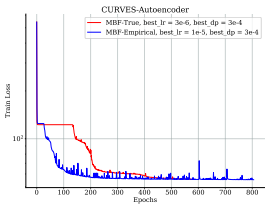
(a) MNIST autoencoder





(b) FACES autoencoder



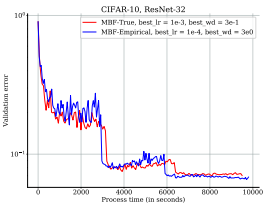


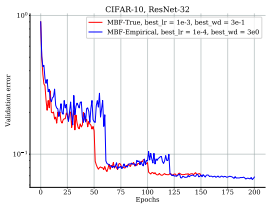
(c) CURVES autoencoder

Figure 2.9: Training performance of MBF-True and MBF on three autoencoder problems.

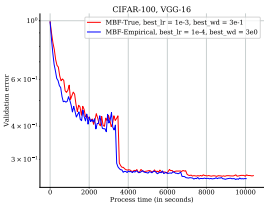
2.5.4 Spacial averaging on convolutional layers.

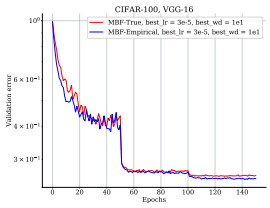
In this section, we compare the performance of MBF with spacial averaging applied to convolutional layers to MBF on the same three CNN problems (CIFAR-10 + ResNet-32, CIFAR-100 + VGG16, and SVHN + VGG11) described in 2.5.4.2. We used the same grid of parameters to



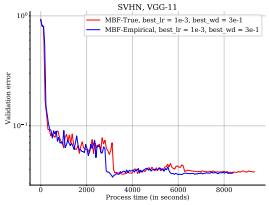


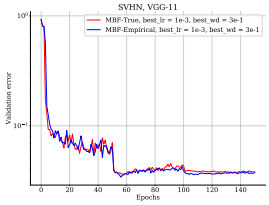
(a) CIFAR-10, ResNet-32





(b) CIFAR-100, VGG-16





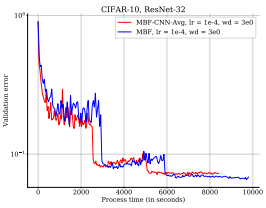
(c) SVHN, VGG-11

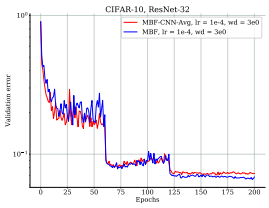
Figure 2.10: Testing performance of MBF-True and MBF on three CNN problems.

tune MBF-CNN-Avg as the one described in 2.5.4.2. We report in Figure 2.11 the validation errors obtained on these problems, as well as the best hyper-parameters for both methods in the legends. It seems that using the average of the kernel-wise mini-blocks to update the preconditioner yields slightly worse results than using the individual mini-blocks as preconditioner. We think this might be the case because the averaging over all mini-blocks results into a loss of curvarture information as the kernel-wise mini-blocks are small in size. Note that, when using the average mini-blocks, MBF requires **less** memory than adaptive first-order methods such as ADAM.

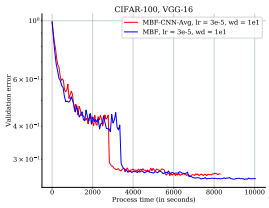
2.5.4 On the effect of the update frequencies T_1, T_2:

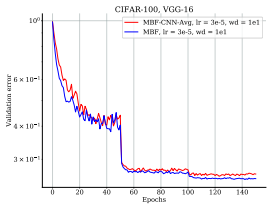
We also explored the effect of the update frequencies T_1, T_2 for the mini-block preconditionners as used in Algorithm 2.2. To be more specific, we tuned the learning rate for various combinations of T_1, T_2 depicted in Figure 2.12. Comparing the performance of Algorithm 2.2 for these different configurations, we can see that the effect of the frequencies T_1, T_2 on the final performance of MBF is minimal and the configurations $T_1, T_2 = (1, 20)$, $T_1, T_2 = (2, 25)$ seem to yield the best performance in terms of process time for autoencoder problems.



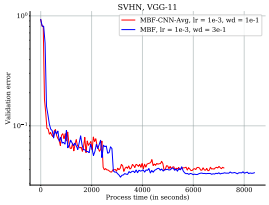


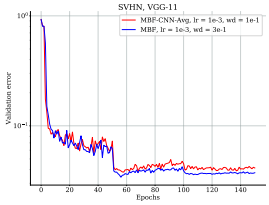
(a) CIFAR-10, ResNet-32





(b) CIFAR-100, VGG-16





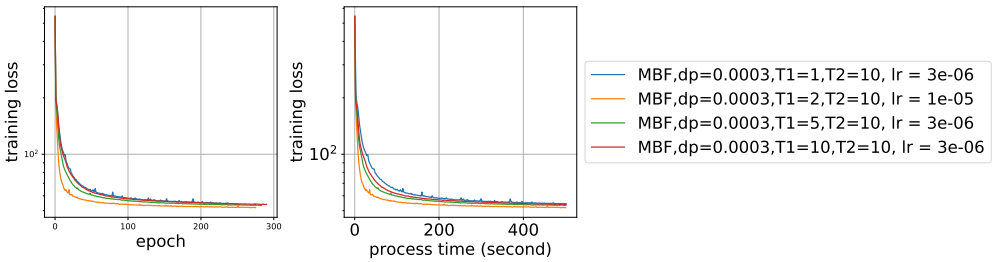
(c) SVHN, VGG-11

Figure 2.11: Testing performance of MBF-CNN-Avg(MBF with spacial averaging applied to CNN layers) and MBF on three CNN problems.

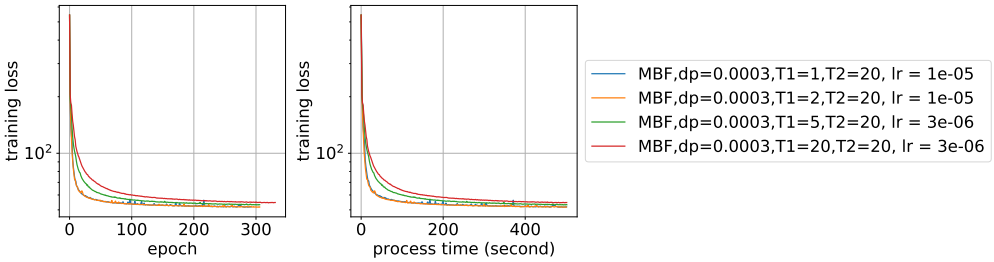
2.5.4 Additional adaptive first order algorithms results

In this section, we compare the performance of two additional adaptive first-order methods AdaBelief and AdaGrad with the performance of SGD-m, Adam(W), Shampoo, MBF and KFAC. The hyperparameterss for these additional methods were tuned using the same grid used to tune Adam(W) on the MNIST Autoencoder problem and CIFAR-100 with VGG-16, and are depicted in Figure 2.13.

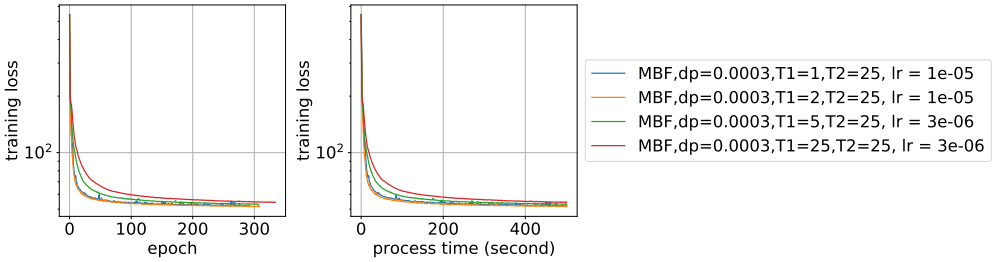
As Figure 2.13 shows, AdaBelief outperformed both AdamW and Adagrad on MNIST and CIFAR-100 (but only slightly so in the comparison to AdamW on MNIST). However, crucially, Adabelief was still outperformed by MBF on these two problems. In the experiments reported in Figures 2.6 and 2.7, we chose to compare MBF (with weight decay, which was included in all of the methods in our tests) against AdamW rather than AdaBelief, since to be fair, if we used the latter variant, we would need to test "belief" versions of MBF, Shampoo and KFAC by incorporating a "belief" term in updating the EMA (Exponential Moving Average) of the preconditioning matrices. This is an interesting research direction for future work.



(a) a) MNIST autoencoder, $T_2 = 10$

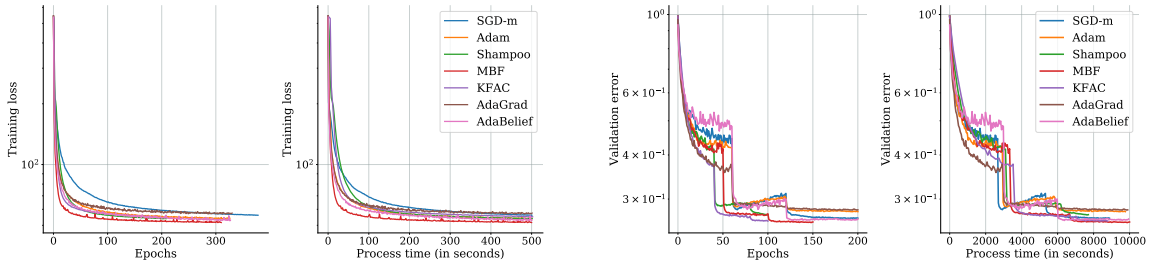


(b) b) MNIST autoencoder, $T_2 = 20$



(c) c) MNIST autoencoder, $T_2 = 25$

Figure 2.12: Training performance of MBF on MNIST autoencoder problems for some combinations of T_1, T_2.



(a) MNIST autoencoder

(b) CIFAR-100, VGG16

Figure 2.13: Additional adaptive first order methods results.

2.5.4 *Additional inverse EFM heatmap illustrations*

We include here additional examples that illustrate that most of the weight in the inverse of the empirical Fisher matrix resides in the mini-blocks used in MBF. For convolutional layers, we trained a simple convolutional neural network, Simple CNN, on Fashion MNIST [64]. The model is identical to the base model described in [82]. It consists of 2 convolutional layers with max pooling with 32 and 64 filters each and 5×5 filters with stride 1, "same" padding, and ReLU activation function followed by 1 fully connected layer. Max pooling uses a 2×2 window with stride 2. The fully connected layer has 1024 units. It does not use batch normalization.

Figure 2.15 shows the heatmap of the absolute value of the inverse empirical Fisher corresponding to the second convolutional layer for channels 1, 16 and 32, which all use 64 filters of size 5×5 (thus 64 mini-blocks of size 25×25 per channel). One can see that the mini-block (by filter) diagonal approximation is reasonable.

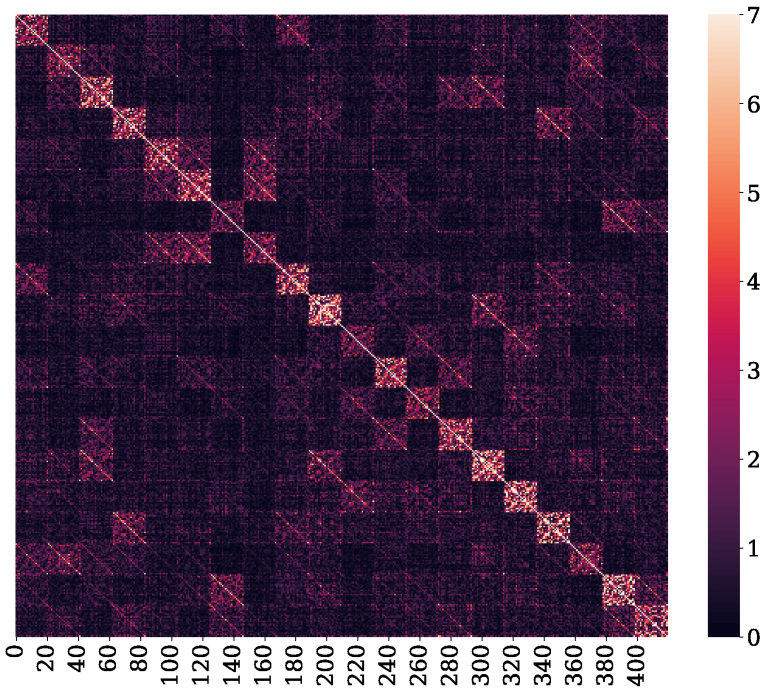
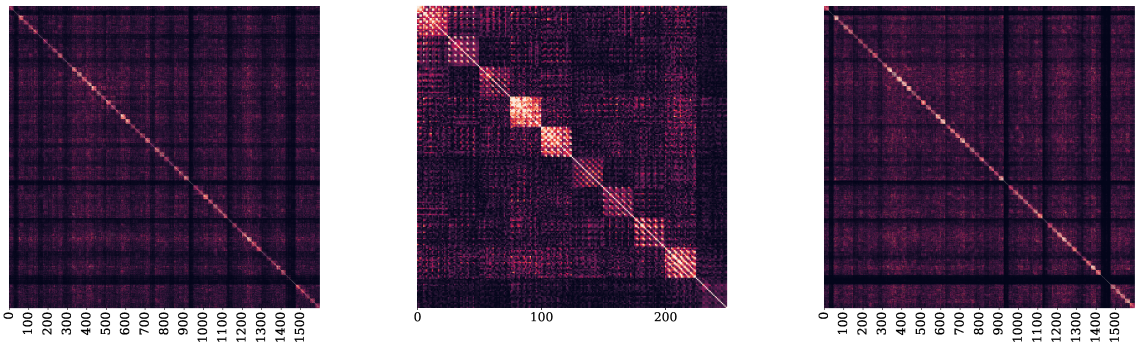


Figure 2.14: Absolute inverse EFM, second fully connected layer 20-20

We illustrate the mini-block structure of the empirical Fisher matrix on a 7-layer (256-20-20-20-20-20-10) feed-forward DNN using tanh activations, partially trained (after 50 epochs using SGD-m) to classify a 16×16 down-scaled version of MNIST that was also used in [33]. Figure 2.14 shows the heatmap of the absolute value of the inverse empirical FIM for the second fully connected layers (including bias). One can see that the mini-block (by neuron) diagonal approximation is reasonable.



(a) Absolute inverse EFM for channel 1

(b) Zoom on the 20th to 30th blocks

(c) Absolute inverse EFM for channel 16

(d) Zoom on the 20th to 30th blocks

(e) Absolute inverse EFM for channel 32

(f) Zoom on the 20th to 30th blocks

Figure 2.15: Absolute inverse of the empirical EFM after 10 epochs for the second convolutional layer of the Simple-CNN.

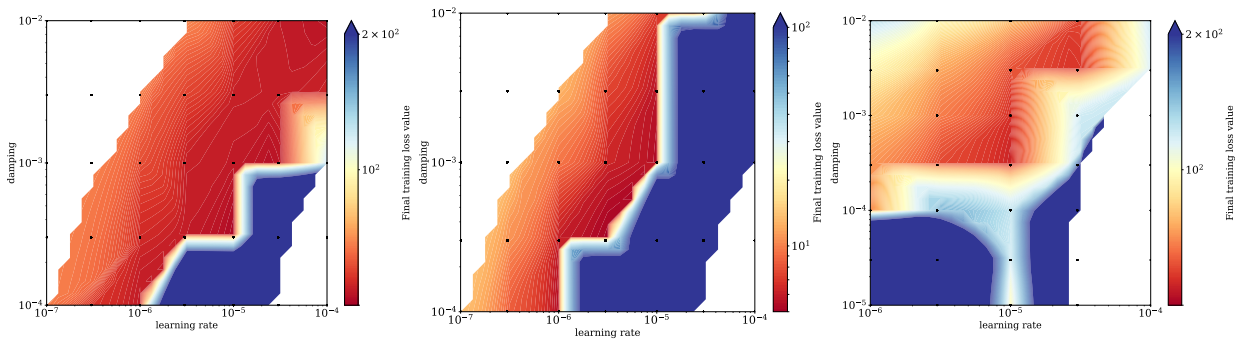


Figure 2.16: The landscape of the final training loss value w.r.t hyper-parameters (i.e. learning rate and damping) for MBF. The left, middle, and right columns depict results for MNIST, FACES, CURVES, which are terminated after 500, 2000, and 500 seconds (CPU time), respectively.

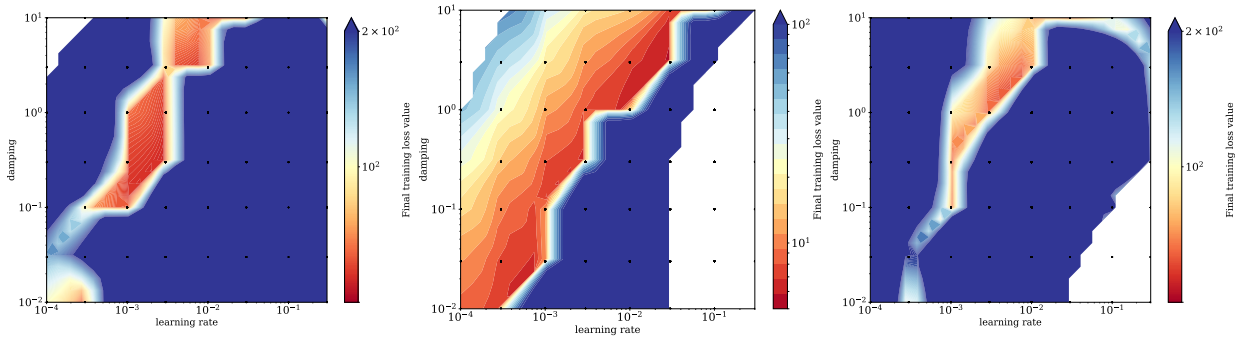
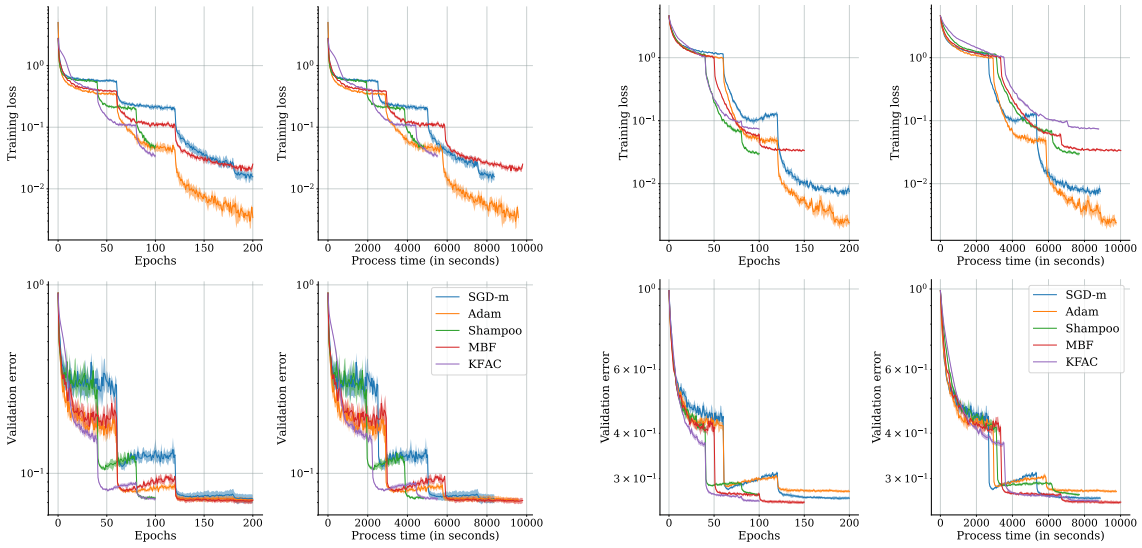


Figure 2.17: The landscape of the final training loss value w.r.t hyper-parameters (i.e. learning rate and damping) for KFAC. The left, middle, and right columns depict results for MNIST, FACES, CURVES, which are terminated after 500, 2000, 500 seconds (CPU time), respectively.

2.5.4 Sensitivity to Hyper-parameters

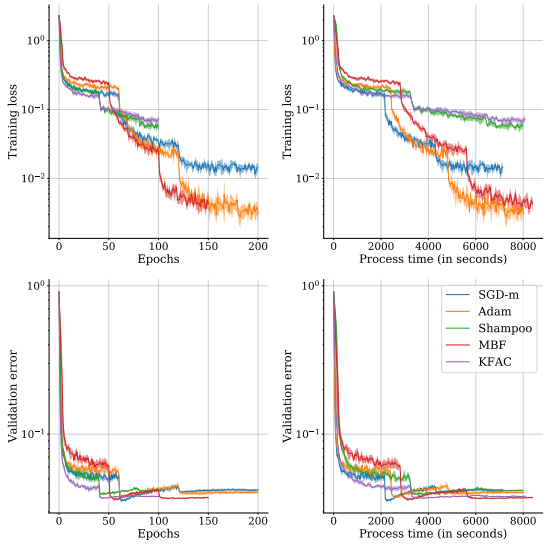
2.5.4 Training and testing plots

For completeness, we report in Figures 2.18 and 2.19 both training and testing performance of the results plotted in Figures 2.6 and 2.7.



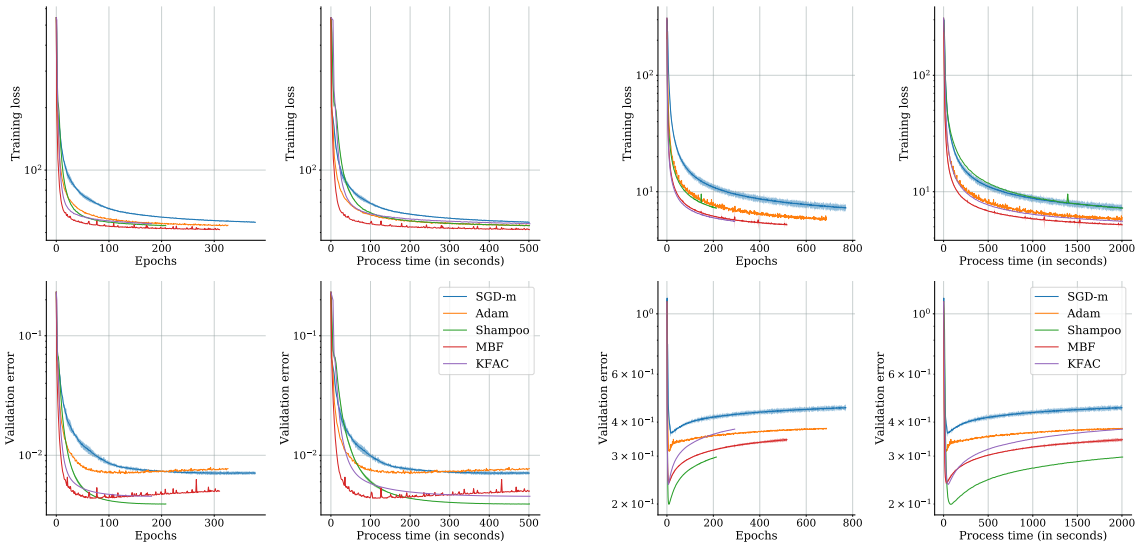
(a) a) CIFAR-10, ResNet-32

(b) b) CIFAR-100, VGG16



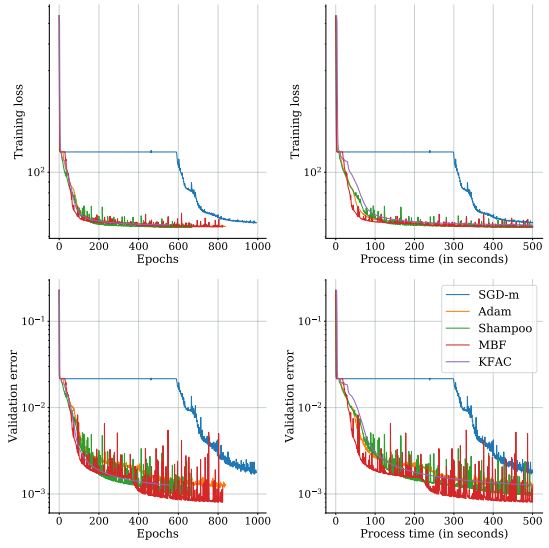
(c) c) SVHN, VGG11

Figure 2.18: Training and testing performance of MBF, KFAC, Shampoo, Adam, and SGD-m on three CNN problems.



(a) MNIST autoencoder

(b) FACES autoencoder



(c) CURVES autoencoder

Figure 2.19: Training and testing performance of MBF, KFAC, Shampoo, Adam, and SGD-m on three autoencoder problems.

2.6 Conclusion and Future Research

We proposed a new EFM-based method, MBF, for training DNNs, by approximating the EFM by a mini-block diagonal matrix that arises naturally from the structure of convolutional and ff-fc layers. MBF requires very mild memory and computational overheads, compared with first-order methods, and is easy to implement. Our experiments on various DNNs and datasets, demonstrate conclusively that MBF provides comparable and sometimes better results than SOTA methods, both from an optimization and generalization perspective. Future research will investigate extending MBF to other deep learning architectures such as Recurrent neural networks.

Chapter 3: Layer-wise Adaptive Step-Sizes for First-Order Optimization Methods

Stochastic gradient descent SGD [83], using either single or mini-batch samples, is widely used because it is parsimonious in terms of both iteration cost and memory usage, and generalizes well [84]. However, to be efficient in practice, the learning rate needs to be chosen with great care. This is also the case for modified variants of SGD that incorporate momentum terms [85]; [86], which have been shown to speed up the convergence of SGD on smooth convex functions. In Deep Neural Networks, other popular variants of SGD scale the individual components of the stochastic gradient with adaptive learning rates using past gradient observations in order to deal with variations in the magnitude of the stochastic gradient components (especially between layers). Among these are ADAGRAD [87], RMSProp [88], ADADELTA [89], and ADAM [90], as well as the structured matrix scaling version SHAMPOO [40] of ADAGRAD. These methods are scale-invariant but do not avoid the need for prior tuning of the base learning rate.

Line search-based methods are classical techniques for determining step sizes in the deterministic setting. The basic idea behind line search-based methods in the stochastic setting is to adjust the step size at each iteration based on the progress made in the previous iteration. This is done by searching along the negative gradient direction for the optimal step size that satisfies a certain criterion, such as the Armijo-Goldstein condition. In the stochastic setting, line search-based methods have been extended in the literature in [91, 92, 93] to work with SGD. Recently, a new group of methods, referred to as SPS (SGD with Polyak Stepsizes), has been proposed in the literature [94, 95, 96, 97, 98, 99]. These techniques utilize both the loss values and gradient norms obtained from sampled points to automatically adjust the step size. This approach enables the use of a non-monotone

adaptive step size that changes from one iteration to the next, based on the current loss value, and therefore adapts to the scaling of the loss function being optimized.

In this chapter, we propose to study a brand new layer-wise adaptive learning rates method that lies between adaptive first-order methods and block diagonal approximate second-order methods. Specifically, we propose a method that uses a block-diagonal pre-conditioner matrix, where the associated block to each layer in the neural network is a scaled identity matrix with a judiciously chosen learning-rate, computed based on the local curvature information of the loss function. Figure 3.1 summarizes how the proposed method approximates the preconditioner matrix.

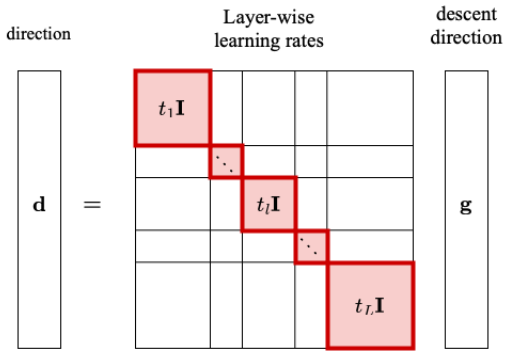


Figure 3.1: Illustration of the proposed per-layer step-sizes method.

Crucially, the method has comparable memory requirements to those of first-order methods and avoids tuning the global learning-rate hyperparameter while its per-iteration time complexity is only roughly equivalent to an additional gradient computation and is much smaller than that of popular second-order methods (e.g. KFAC) for training DNNs.

3.1 Problem Formulation and Notation

The problem of interest is minimizing for $\boldsymbol{W} \in \mathbb{R}^d$ functions of the form

$$F(\boldsymbol{W}) = \int f(\boldsymbol{W}; x, y) dP(x, y) = \mathbb{E}[f(\boldsymbol{W}; \xi)], \tag{3.1.1}$$

common to problems in statistics and machine learning. For instance, in the empirical risk minimization framework, a model is learned from a set $\{\xi_1 = (x_1, y_1), \ldots, \xi_m = (x_n, y_n)\}$, of training

data by minimizing an empirical loss function of the form

$$F(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} f\left(\boldsymbol{W}, (x_i, y_i)\right) = \frac{1}{n}\sum_{i=1}^{n} F_i(\boldsymbol{W}), \tag{3.1.2}$$

where we define $F_i(\boldsymbol{W}) = f(\boldsymbol{W}; x_i, y_i)$, and f is the composition of a prediction function (parametrized by $\boldsymbol{W}$) and a loss function, and (x_i, y_i) are random input-output pairs with the uniform discrete probability distribution $P(x_i, y_i) = \frac{1}{n}$. An objective function of the form (3.1.1) is often impractical, as the distribution of ξ is generally unavailable, making it infeasible to analytically compute $\mathbb{E}[f(\boldsymbol{W}; \xi)]$. This can be resolved by replacing the expectation by the estimate (3.1.2). The strong law of large numbers implies that the sample mean in (3.1.2) converges almost surely to (3.1.1) as the number of samples n increases. However, in practice, even problem (3.1.2) is not tractable for classical optimization algorithms, as the amount of data is usually extremely large. A better strategy when optimizing (3.1.2) is to consider sub-samples of the data to reduce the computational cost. This leads to stochastic algorithms where the objective function changes at each iteration by randomly selecting a mini-batch of sub-samples.

Notation. $\text{Diag}_{i\in[L]}(D_i)$ is the block diagonal matrix with $\{D_1, ..., D_L\}$ on its diagonal; $[L] := \{1, ..., L\}$; $\boldsymbol{x} = [x_1, ..., x_n]^\top \in \mathbb{R}^{n\times d}$ is the input vec; $\lambda_{\min}(H), \lambda_{\max}(H)$ are the smallest and largest eigenvalues of the matrix H; $\otimes$ denotes the Kronecker product; $\|.\|_2$ denotes the Euclidean norm of a vector or matrix; and $\text{vec}(A)$ vectorizes A by stacking its columns.

For this work, we consider F to be the loss function of a Deep Neural Network(DNN) with L layers, defined by weight matrices W_l(including the bias), for $\lambda \in [L]$, that transforms the input data $\boldsymbol{x}$ to an output $f(\boldsymbol{W}, \boldsymbol{x})$.

For a data-point $(\boldsymbol{x}, y)$, the loss $\ell\left(f(\boldsymbol{W}, \boldsymbol{x}), y\right)$ between the output $f(\boldsymbol{W}, \boldsymbol{x})$ and the label y, is a non-convex function of $\text{vec}(\boldsymbol{W})^\top = \left[\text{vec}\left(W^{(1)}\right)^\top, ..., \text{vec}\left(W^{(\lambda)}\right)^\top\right] \in \mathbb{R}^p$, containing all of the network's parameters, concatenated together and ℓ measures the accuracy of the prediction (e.g. squared error loss, cross-entropy loss). The optimal parameters are obtained by minimizing the

average loss F over the training set:

$$F(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{W}, \boldsymbol{x}_i), y_i), \tag{3.1.3}$$

This setting is applicable to most common models in deep learning such as multilayer perceptrons (MLPs), Convolutional Neural Networks(CNNs), Graph Convolutional Networks(GCNs), etc. In these models, the trainable parameters $W^{(\lambda)}$ ($\lambda = 1, \ldots, L$) come from the weights(including the bias) of a layer, whether it be a feed-forward, convolutional, recurrent, etc.

We consider iterative methods that take the following form: at the k-th iteration, we draw m i.i.d samples $S_k = \{\xi_{i_1}, \ldots, \xi_{i_m}\} = \{(\boldsymbol{x}_{i_1}, y_{i_1}), \ldots, (\boldsymbol{x}_{i_m}, y_{i_m})\}$ and define the sub-sampled objective function and its sub-sampled gradient and Hessian at the current weight vector $\boldsymbol{W}_k = \left[W_k^{(1)}, ..., W_k^{(\lambda)}\right]$as

$$F_{S_k}(\boldsymbol{W}_k) = \frac{1}{m}\sum_{i\in S_k} \ell(f(\boldsymbol{W}_k, \boldsymbol{x}_i), y_i) = \frac{1}{m}\sum_{i\in S_k} F_i(\boldsymbol{W}_k),$$

$$g_k = \nabla F_{S_k}(\boldsymbol{W}_k) = \frac{1}{m}\sum_{i\in S_k} \nabla F_i(\boldsymbol{W}_k),$$
$$H_k = \nabla^2 F_{S_k}(\boldsymbol{W}_k) = \frac{1}{m}\sum_{i\in S_k} \nabla^2 F_i(\boldsymbol{W}_k).$$

A first-order method based on these approximations is then given by

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - t_k d_k, \tag{3.1.4}$$

where t_k is a global step size, d_k is the descent direction of the considered first-order algorithm and the sets of samples S_k that are used to estimate the gradient and Hessian. For a layer l, we denote by $F_{S_k}^{(\lambda)}$ the restricted loss function that maps the weights $W_k^{(\lambda)}$ to $F_{S_k}(\boldsymbol{W}_k)$ with the other weight matrices $W_k^{(j)}$, $j \neq \lambda$ fixed. Therefore, similarly, we define the sub-sampled layer-wise gradient and

Hessian as

$$g_k^{(\lambda)} = \nabla F_{S_k}^{(\lambda)}\left(W_k^{(\lambda)}\right) = \frac{1}{m}\sum_{i\in S_k} \nabla_{W_k^{(\lambda)}} F_i\left(\boldsymbol{W}_k\right),$$
$$H_k^{(\lambda)} = \nabla^2 F_{S_k}^{(\lambda)}\left(W_k^{(\lambda)}\right) = \frac{1}{m}\sum_{i\in S_k} \nabla^2_{W_k^{(\lambda)}} F_i\left(\boldsymbol{W}_k\right).$$

Therefore, a layer-wise version of (3.1.4) can be defined as:

$$W_{k+1}^{(\lambda)} = W_k^{(\lambda)} - t_k^{(\lambda)} d_k^{(\lambda)}, \tag{3.1.5}$$

where $t_k^{(\lambda)}$ are layer-wise step-sizes.

3.2 Motivation for Layer-wise Adaptive step-sizes

In this section, we motivate the proposed method through two main results in the literature. The first one is related to closed-form step-sizes that guarantee an improvement in iterative methods described in (3.1.4) for self-concordant loss functions. The second result is the theoretical observation mentioned in [100], where the authors proved that, for feed-forward neural networks with certain activation functions such as ReLU and regularization, the Hessian blocks associated with each layer are Positive definite and therefore, one can show, using results in [101], that the associated restricted loss function $F_{S_k}^{(\lambda)}$ is standard self-concordant if we assume that the loss function general self-concordant. By combining the latter results, we are able to develop a layer-wise closed-form step-sizes procedure for training deep neural networks.

Self-concordant Functions and Nesterov Step-Sizes: Self-concordant functions were introduced by Nesterov and Nemirovski in the context of interior-point methods [102]. We recall the definition of self-concordance property as follow:

Definition 3.1. *A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if there exists a constant c such that for every $x \in \mathbb{R}^n$ and every $h \in \mathbb{R}^n$, we have :*

$$\left|\nabla^3 f(x)[h,h,h]\right| \leq c\left(\nabla^2 f(x)[h,h]\right)^{3/2},$$

f is standard self-concordant if the above is satisfied for $c = 2$.

Many problems in machine learning have self-concordant formulations: In [103] and [104], it is shown that regularized regression, with either logistic loss or hinge loss, is self-concordant. For the iterative method for minimizing self-concordant functions described in (3.1.4) using the following choice of the global step-size

$$t_k^* = \frac{\rho_k}{(\rho_k + \delta_k)\,\delta_k};\;\; \delta_k = \|d_k\|_{W_k} = \sqrt{d_k^T H_k d_k},$$

where $\rho_k = g_k^T d_k$. Methods of this type have been analyzed in [105] and [106] in the deterministic setting (i.e $m = n$). In the latter paper, the above choice of t_k is shown to guarantee a decrease in the function value.

Lemma 3.1. *(Lemma 4.1, [106]) For F standard self-concordant, for all $0 \leq t < \frac{1}{\delta_k}$:*

$$F(x_k - td_k) \leq F(x_k) - \Delta(\delta_k, t). \tag{3.2.1}$$

With $\Delta(\delta_k, t) = (\delta_k + \rho_k)t + log(1 - \delta_k t)$

If $t = t_k^*$ then $F\left(x_k - t_k d_k\right) \leq F\left(x_k\right) - \omega\left(\eta_k\right)$, where $\eta_k = \frac{\rho_k}{\delta_k}$ and $\omega(z) = z - \log(1+z)$. The latter results allow one to prove the linear convergence of such iterative methods by controlling the bound on the improvement of the loss function in every iteration.

Hessian structure in Feed-forward Neural Networks: Consider a feed-forward neural network takes an single sample input $a^{(0)} = x$ and produces an output vector $h^{(L)}$ on the final L layer of the

network, following the forward pass described as

$$h^{(\lambda)} = W^{(\lambda)} a^{(\lambda-1)}; \quad a^{(\lambda)} = f_\lambda \left(h^{(\lambda)} \right) \quad 1 \leq \lambda < L,$$

where $h^{(\lambda)}$ is the pre-activation in layer λ and $a^{(\lambda)}$ are the activation values; and f_λ the element-wise activation functions. The pre-activation Hessian for layer λ is defined as:

$$\left[\mathcal{H}^{(\lambda)} \right]_{i,j} = \frac{\partial^2 F}{\partial h_i^{(\lambda)} \partial h_{\lambda,j}^{(\lambda)}}$$

One can show that the sub-sampled Hessian of $W^{(\lambda)}$ can be expressed as:

$$\begin{aligned} H^{(\lambda)} &= \frac{\partial^2 F}{\partial \operatorname{vec}\left(W^{(\lambda)}\right) \partial \operatorname{vec}\left(W^{(\lambda)}\right)} \\ &= \left(a^{(\lambda-1)} a^{(\lambda-1)\top} \right) \otimes \mathcal{H}^{(\lambda)} \end{aligned}$$

where $\otimes$ denotes the Kronecker product. In [100], the authors show that the pre-activation Hessian can be computed recursively as:

$$\mathcal{H}^{(\lambda)} = B^{(\lambda)} W^{(\lambda+1)\top} \mathcal{H}^{(\lambda+1)} W^{(\lambda+1)} B^{(\lambda)} + D^{(\lambda)}$$

where the diagonal matrices $B^{(\lambda)}$, $D^{(\lambda)}$ are defined as:

$$\begin{aligned} B^{(\lambda)} &= \operatorname{diag}\left(f_\lambda' \left(h^{(\lambda)} \right) \right) \\ D^{(\lambda)} &= \operatorname{diag}\left(f_\lambda'' \left(h^{(\lambda)} \right) \frac{\partial F}{\partial a^{(\lambda)}} \right) \end{aligned}$$

and f_λ' and f_λ'' are the first and second derivatives of f_λ respectively. The recursion is initialized with $\mathcal{H}^{(\lambda)}$, which depends on the objective function and is easily calculated analytically for the usual objectives, for example for squared loss, the Hessian is simply the identity matrix. For more

than a single sample, the recursion is applied per datapoint and the parameter Hessian is given by the average of the individual sample Hessians.

In recent years piece-wise linear activation functions, such as the ReLU function $f(x) = \max(x, 0)$, have become popular. It has been argued that in contrast to the standard sigmoidal functions they don't saturate which prevents the exploding/vanishing gradient problem. Since the second derivative f'' of a piecewise linear function is zero everywhere, the matrices $D^{(\lambda)}$ in the recursion will be zero (away from non-differentiable points). It follows that if $\mathcal{H}^{(\lambda)}$ is Positive Semi-Definite (PSD), which is the case for the most commonly used loss functions, **the block diagonal Hessian matrices are PSD for every layer**, additionally if we add an L2 regularization term to the loss function, the restricted loss $F_{S_k}^{(\lambda)}$ function for each layer λ is **strictly convex**. Using results in [101], if one assumes that the loss function is general self-concordant then, then the restricted loss $F_{S_k}^{(\lambda)}$ becomes standard self-concordant. Consequently, we can exploit Lemma 3.1 to develop closed-form layer-wise step-sizes and combine them judiciously to obtain a loss decrease guarantee.

We numerically investigate the Hessian blocks as well as the full hessian spectrum of a trained simple 3-layer GCN with the following node-sizes $[1433, 128, 64, 7]$, on Cora data set using vanilla Adam for 20 epochs. We exploit the scalable framework proposed in [107] that enables fast computation of the full and block Hessian eigenvalue/spectral density, using the Stochastic Lanczos Quadrature method. We report the results in Figure 3.2. We observe that, indeed, the eigenvalues support of the block Hessians is positive contrary to the full Hessian, it has a non-zero. We also notice that the union of block-Hessian eigenvalues supports overlap with the positive support of the full Hessian indicating that the block diagonal approximation to the Hessian is not unreasonable.

A layer-wise step-sizes procedure: Based on the above arguments, we propose Algorithm 3.1 that gives the pseudo-code for a generic version of the layer-wise procedure for any optimization algorithm that produces a direction d_k at iteration k.

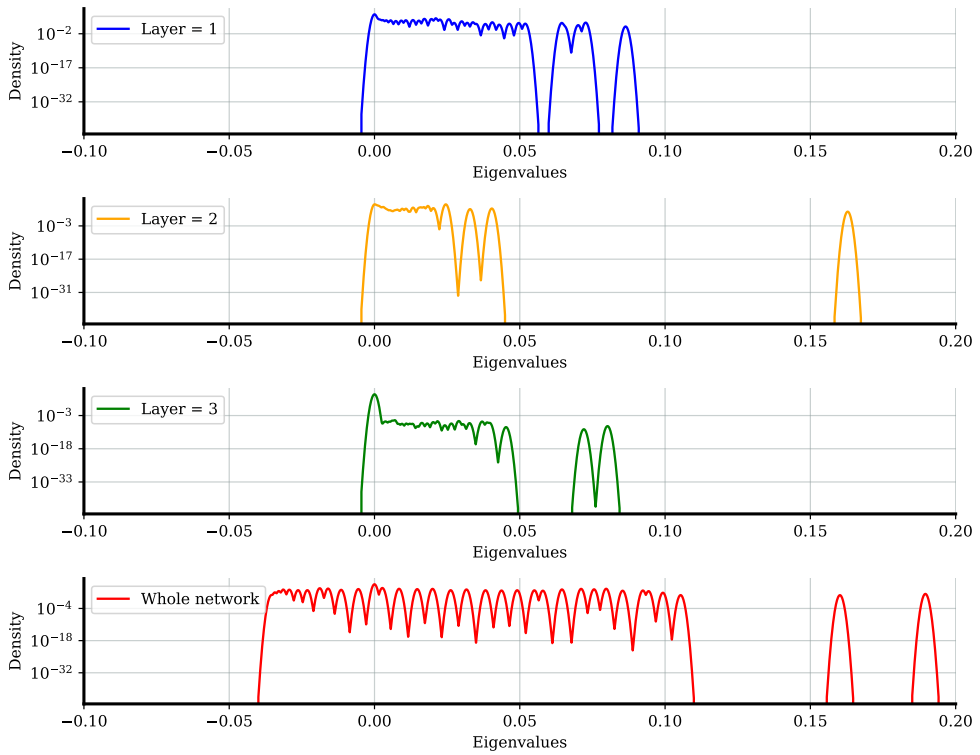


Figure 3.2: Hessian and block-Hessian eigenvalues density estimations using Stochastic Lanczos Quadrature method.

We later propose in Section 6 below, the practical version of the algorithm for both SGD with momentum and AdamW. However, we first present the theoretical results of the linear convergence of the version of the procedure applied to vanilla SGD in the full batch setting.

3.3 Linear Convergence

In this section, we provide a linear convergence guarantee for vanilla gradient descent with Layer-Wise Step-Sizes Procedure(SGD-LW) with exact gradients (i.e. the deterministic case with full batch $m = n$). Our theoretical analysis is based on the following technical assumptions on F:

A1. **Hessian regularity:** There exist $M \geq m > 0$ s.t $\forall \boldsymbol{W} \in \mathbb{R}^n \; mI \preceq \nabla^2 F(\boldsymbol{W}) \preceq MI$

A2. **B–Bounded iterates and Gradient regularity:** There exists $B > B_0 > 0$ with $\boldsymbol{W}^* \in \mathbb{B}(0, B)$, such that if $\boldsymbol{W}_0$ is chosen in $\mathbb{B}(0, B_0)$, then the sequence of iterates $\{\boldsymbol{W}_k\}_{k=0}^{\infty}$ produced by the

Algorithm 3.1: X-LW: Algorithm X with Layer-Wise Step-Sizes Procedure

Input: Initial iterate $\boldsymbol{W}_0$, batch size m, max number of iterations N and Algorithm X hyper-parameters.

$k \leftarrow 0$

while $k < N$ **do**

  Sample m samples S_k.

  Compute $g_k = \nabla F_{S_k}(\boldsymbol{W}_k)$.

  Compute d_k Algorithm X direction.

  for $\lambda = 1 \dots L$ **do**

    Compute $\rho^{(\lambda)} = d_k^{(\lambda)^\top} g_k^{(\lambda)}$.

    Compute $\delta^{(\lambda)} = \sqrt{d_k^{(\lambda)^\top} H_k^{(\lambda)} d_k^{(\lambda)}}$.

    Compute $t_k^{(\lambda)} = \frac{1}{L} \frac{\rho^{(\lambda)}}{(\rho^{(\lambda)} + \delta^{(\lambda)})\delta^{(\lambda)}}$

    $W_{k+1}^{(\lambda)} \leftarrow W_k^{(\lambda)} - t_k^{(\lambda)} d_k^{(\lambda)}$

  Set $k \leftarrow k + 1$

algorithm is contained within $\mathbb{B}(0, B)$. Furthermore, we assume that F has bounded gradients within $\mathbb{B}(0, B)$: $\exists\ \gamma > 0,\ \forall \boldsymbol{W} \in \mathbb{B}(0, B) : \|\nabla F(\boldsymbol{W})\| \leq \gamma$.

A3. **Layer-wise self-concordance:** F is layer-wise standard self-concordant, i.e for each layer $\lambda = 1 \cdots L$ the restricted function $F^{(\lambda)}$ is standard self-concordant.

The bounded gradient assumption A2 could be seen as a technical assumption to simplify the proofs, as gradients of many common self-concordant functions can be unbounded. This assumption is equivalent to assuming that the algorithm is not divergent and the steps stay within a bounded region in which we can assume that the gradient is bounded **locally** because there are no functions F that are strongly convex and for which the gradients are **globally** bounded.

Recent work [108] [109] [**garrigos2023handbook**] has shown that SGD can be proved to converge without requiring A2. Extending these ideas to our setting is a subject for further research. By using Lemma 3.1 in a judicious manner, we can control the loss improvement in every iteration to obtain linear convergence of the Algorithm SGD-LW:

Theorem 3.1. *Suppose that F satisfies Assumptions A1-A3. Let $\boldsymbol{W}_k$ be the iterates generated by taking the layer-wise steps $t_k^{(\lambda)}$ at iteration k for the Algorithm SGD-LW with full batch(i. e $m = n$),*

starting from any $\boldsymbol{W}_0$. Then, for any k, we have :

$$(F(\boldsymbol{W}_k) - F(\boldsymbol{W}^*)) \leq \rho^k (F(\boldsymbol{W}_0) - F(\boldsymbol{W}^*)),$$

where $\rho = 1 - \frac{m}{ML(1+\frac{\gamma}{\sqrt{m}})}$.

Proof of Theorem 3.1. The proof is a consequence of Lemma 3.1 (Lemma 4.1 in [106]), judiciously applied to the layer-wise restricted functions $F^{(\lambda)}(\cdot)$. More specifically, since we are considering the direction given by SGD-LW, we have $d_k^{(\lambda)} = g_k^{(\lambda)}$ and the overall direction of the SGD-LW is given by concatenating the vectors $t_k^{(\lambda)} d_k^{(\lambda)}$ for $\lambda = 1 \dots L$ into an overall step $\hat{p}$ which can be expressed as a convex combination of the following vectors:

$$\hat{p}^{(\lambda)} = \left[\mathbf{0}^{(1)}, ..., \frac{\rho^{(\lambda)}}{\left(\rho^{(\lambda)} + \delta^{(\lambda)}\right) \delta^{(\lambda)}} d_k^{(\lambda)}, ..., \mathbf{0}^{(L)} \right].$$

Therefore $\hat{p}$ can be expressed as $\hat{p} = \sum_{\lambda=0}^{L} \frac{1}{L} \hat{p}^{(\lambda)}$.

$$\begin{aligned} F(\boldsymbol{W}_k - \hat{p}) &= F\left(\frac{1}{L} \sum_{\lambda=0}^{L} \left(\boldsymbol{W}_k - \hat{p}^{(\lambda)} \right) \right) \\ &\overset{(a)}{\leq} \frac{1}{L} \sum_{\lambda=0}^{L} F\left(\boldsymbol{W}_k - \hat{p}^{(\lambda)} \right) \overset{(b)}{=} \frac{1}{L} \sum_{\lambda=0}^{L} F^{(\lambda)} \left(\boldsymbol{W}_k^{(\lambda)} - \frac{\rho^{(\lambda)}}{\left(\rho^{(\lambda)} + \delta^{(\lambda)}\right) \delta^{(\lambda)}} d_k^{(\lambda)} \right) \\ &\overset{(c)}{\leq} \frac{1}{L} \sum_{\lambda=0}^{L} F^{(\lambda)} \left(\boldsymbol{W}_k^{(\lambda)} \right) - \Delta(\delta^{(\lambda)}, \frac{\rho^{(\lambda)}}{\left(\rho^{(\lambda)} + \delta^{(\lambda)}\right) \delta^{(\lambda)}}). \end{aligned}$$

Where, in (a), we used the convexity of F by Assumption A1, in (b), we used the definition of the vectors $\hat{p}^{(\lambda)}$ and the restricted loss functions $F^{(\lambda)}(\cdot)$, and in (c), we applied Lemma 3.1 to the latter functions. We have:

$$\Delta(\delta^{(\lambda)}, \frac{\rho^{(\lambda)}}{\left(\rho^{(\lambda)} + \delta^{(\lambda)}\right) \delta^{(\lambda)}}) = \omega\left(\eta_k^{(\lambda)}\right) \quad \text{where } \omega(z) = z - \log(1+z) \text{ and } \eta_k^{(\lambda)} = \frac{\rho_k^{(\lambda)}}{\delta_k^{(\lambda)}}.$$

Hence, we have:

$$F(\boldsymbol{W}_k - \hat{p}) \overset{(d)}{\leq} F(\boldsymbol{W}_k) - \frac{1}{L}\sum_{\lambda=0}^{L} \omega\left(\eta_k^{(\lambda)}\right).$$

Where in (d), we use the fact that $F^{(\lambda)}\left(\boldsymbol{W}_k^{(\lambda)}\right) = F(\boldsymbol{W}_k)$. We, furthermore, have:

$$\eta_k^{(\lambda)} = \frac{\rho_k^{(\lambda)}}{\delta_k^{(\lambda)}} = \frac{\|g_k^{(\lambda)}\|^2}{\sqrt{g_k^{(\lambda)^\top} H_k^{(\lambda)} g_k^{(\lambda)}}} \leq \frac{\|g_k^{(\lambda)}\|}{m} \leq \frac{\gamma}{m}$$

where we used Assumptions A1 and A2 in the last two inequalities respectively. By observing that $\omega(z) = z - log(1+z)$ satisfies $\omega(z) \geq \frac{1}{2}(1+\Gamma)^{-1}z^2$ for all $z \in [0, \Gamma]$, we have:

$$F(\boldsymbol{W}_k - \hat{p}) \leq F(\boldsymbol{W}_k) - \frac{1}{2L(1+\frac{\gamma}{\sqrt{m}})}\sum_{\lambda=0}^{L}(\eta_k^{(\lambda)})^2.$$

Using Assumption A1, we obtain the following lower bound on the term $\sum_{\lambda=0}^{L}(\eta_k^{(\lambda)})^2$:

$$\begin{aligned}\sum_{\lambda=0}^{L}(\eta_k^{(\lambda)})^2 &= \sum_{\lambda=0}^{L}\frac{\|g_k^{(\lambda)}\|^4}{g_k^{(\lambda)^\top} H_k^{(\lambda)} g_k^{(\lambda)}} \geq \sum_{\lambda=0}^{L}\frac{\|g_k^{(\lambda)}\|^4}{M\|g_k^{(\lambda)}\|^2} \\ &\geq \frac{1}{M}\sum_{\lambda=0}^{L}\|g_k^{(\lambda)}\|^2 := \frac{1}{M}\|\nabla F(\boldsymbol{W}_k)\|^2\end{aligned}$$

Therefore, we obtain

$$F(\boldsymbol{W}_k - \hat{p}) = F(\boldsymbol{W}_{k+1}) \leq F(\boldsymbol{W}_k) - \frac{1}{2ML(1+\frac{\gamma}{\sqrt{m}})}\|\nabla F(\boldsymbol{W}_k)\|^2. \tag{3.3.1}$$

It is well known [110] that for strongly convex functions:

$$\|\nabla F(\boldsymbol{W}_k)\|^2 \geq 2m[F(\boldsymbol{W}_k) - F(\boldsymbol{W}^*)].$$

Substituting this in (3.3.1) and subtracting $F(\boldsymbol{W}^*)$ from both sides, we obtain:

$$F(\boldsymbol{W}_{k+1}) - F(\boldsymbol{W}^*) \leq \rho(F(\boldsymbol{W}_k) - F(\boldsymbol{W}^*)),$$

with :

$$\rho = 1 - \frac{m}{ML(1 + \frac{\gamma}{\sqrt{m}})},$$

from which the theorem follows by induction on k.

□

3.4 Implementation Details and Practical Considerations

Hessian vector product and computational complexity: In our framework, we need to compute the curvature along a mini-batch stochastic gradient direction given by : $\delta^{(\lambda)} = \sqrt{d_k^{(\lambda)^\top} H_k^{(\lambda)} d_k^{(\lambda)}}$. Hence we need to efficiently compute the Hessian-vector product $H_k^{(\lambda)} d_k^{(\lambda)}$. Fortunately, for functions that can be computed using a computational graph (Logistic regression, DNNs, etc) there are automatic methods available for computing Hessian-vector products exactly [111], which take about as much computation as gradient evaluations. **Hence, $H_k^{(\lambda)} d_k^{(\lambda)}$ can be computed with essentially the same effort as that needed to compute $g_k^{(\lambda)}$.** The method described in [111] is based on the differential operator:

$$\mathcal{R}\{F(\boldsymbol{W})\} = (\partial/\partial r) F(\boldsymbol{W} + r\boldsymbol{d})|_{r=0}\,.$$

Since $\mathcal{R}\{\nabla_{\boldsymbol{W}} F\} = Hd$ and $\mathcal{R}\{\boldsymbol{W}\} = d$, to compute Hd, [111] applies $\mathcal{R}$ to the back-propagation equations used to compute ∇F.

Exponentially Moving Averages, Amortized updates and weight-decay: We use moving averages to both reduce the stochasticity and incorporate more information from the past, more specifically, we use a moving average scheme to get a better estimate of the layer-wise learning rates, i.e.$t_{k+1}^{(\lambda)} = \beta_t t_k^{(\lambda)} + (1 - \beta_t) \frac{\rho^{(\lambda)}}{L(\rho^{(\lambda)} + \delta^{(\lambda)})\delta^{(\lambda)}}$ with $\beta_t = 0.99$. The extra work for the $\frac{\rho^{(\lambda)}}{(\rho^{(\lambda)} + \delta^{(\lambda)})\delta^{(\lambda)}}$

computation compared with first-order methods is amortized by only performing the updates every T iterations. This approach is also used in second-order algorithms such as KFAC and Shampoo. We also do not use the computed learning rates in the first 10 epochs to warm-up the moving average estimates. We incorporate weight-decay with the tunnable hyper-parameter γ by adding the term $\gamma \boldsymbol{W}_k^{(\lambda)}$ to the direction $d_k^{(\lambda)}$ when computing the layer-wise adaptive steps $t_k^{(\lambda)}$.

Full algorithms: The pseudocode that fully describes our layer-wise step-size procedure for both SGD with momentum(SGD-m-LW) and AdamW(AdamW-LW) are given in Algorithm 3.2 and Algorithm 3.3.

Algorithm 3.2: SGD-m-LW: Per-Layer Adaptive Step-Size for SGD-m

Input: Initial iterate $\boldsymbol{W}_0$, batch size m, and max number of iterations N.
Betas $\beta, \beta_t = 0.9, 0.99$, Weight-decay γ, Update-frequency T.
$k \leftarrow 0$
while $k < N$ **do**
  Sample m samples S_k.
  Compute $g_k = \nabla F_{S_k}(\boldsymbol{W}_k)$.
  Compute $m_k = \beta d_k + g_k$
  for $l = 1 \ldots L$ **do**
    Compute $d_k^{(\lambda)} = m_k^{(\lambda)} + \gamma \boldsymbol{W}_k^{(\lambda)}$
    if $k \mod T = 0$ **then**
      Compute $\rho^{(\lambda)} = {d_k^{(\lambda)}}^\top g_k^{(\lambda)}$.
      Compute $\delta^{(\lambda)} = \sqrt{{d_k^{(\lambda)}}^\top H_k^{(\lambda)} d_k^{(\lambda)}}$.
      $t_k^{(\lambda)} \leftarrow \beta_t t_{k-1}^{(\lambda)} + (1-\beta_t)\frac{1}{L}\frac{\rho^{(\lambda)}}{(\rho^{(\lambda)}+\delta^{(\lambda)})\delta^{(\lambda)}}$
    $W_{k+1}^{(\lambda)} \leftarrow W_k^{(\lambda)} - t_k^{(\lambda)} d_k^{(\lambda)}$
  Set $t \leftarrow t + 1$

3.5 Experiments

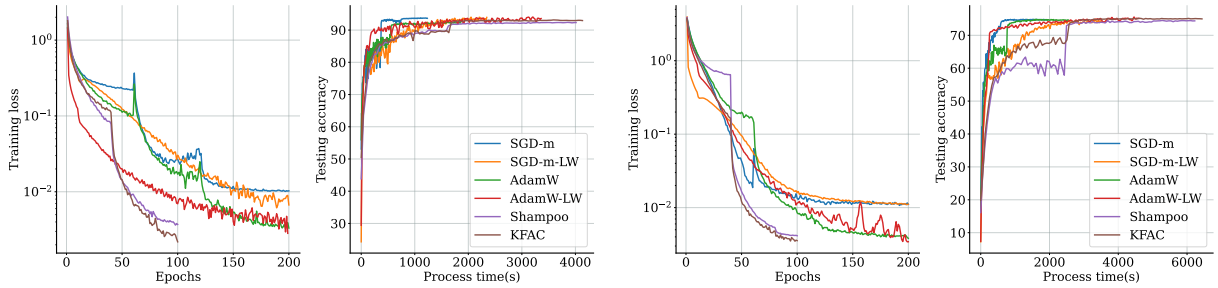
In this section, we compare SGD-m-LW and AdamW-LW with some SOTA **fine-tuned learning rate** first-order (SGD-m, Adam) and second-order (KFAC, Shampoo) methods. (See previous chapter on how these methods were implemented.) Since SGD-m-LW and AdamW-LW use layer-wise step sizes to scale the directions, fine-tuned SGD-m and AdamW were obvious choices for comparison. We also included KFAC and Shampoo in our results as SGD-m-LW and AdamW-LW

Algorithm 3.3: AdamW-LW: Per-Layer Adaptive Step-Size for AdamW

Input: Initial iterate $\boldsymbol{W}_0$, batch size m, and max number of iterations N.
Betas $\beta_1, \beta_2, \beta_t = 0.9, 0.999, 0.99$, Damping $\epsilon = 10^{-8}$, Weight-decay γ, Update-frequency T.
$k \leftarrow 0$
while $k < N$ **do**
  Sample m samples S_k.
  Compute $g_k = \nabla F_{S_k}(\boldsymbol{W}_k)$.
  Compute $m_k = \beta_1 m_k + (1-\beta_1) g_k, \quad v_k = \beta_2 v_k + (1-\beta_2) g_k^2$
  Compute $\hat{m}_k = \frac{m_k}{(1-\beta_1^k)}, \quad \hat{v}_k = \frac{v_k}{(1-\beta_2^k)}$
  for $l = 1 \ldots L$ **do**
    Compute $d_k^{(\lambda)} = \frac{\hat{m}_k^{(\lambda)}}{\sqrt{\hat{v}_k^{(\lambda)}+\epsilon}} + \gamma \boldsymbol{W}_k^{(\lambda)}$
    if $k \mod T = 0$ **then**
      Compute $\rho^{(\lambda)} = {d_k^{(\lambda)}}^\top g_k^{(\lambda)}$.
      Compute $\delta^{(\lambda)} = \sqrt{{d_k^{(\lambda)}}^\top H_k^{(\lambda)} d_k^{(\lambda)}}$.
      $t_k^{(\lambda)} \leftarrow \beta_t t_{k-1}^{(\lambda)} + (1-\beta_t)\frac{1}{L}\frac{\rho^{(\lambda)}}{(\rho^{(\lambda)}+\delta^{(\lambda)})\delta^{(\lambda)}}$
    $W_{k+1}^{(\lambda)} \leftarrow W_k^{(\lambda)} - t_k^{(\lambda)} d_k^{(\lambda)}$
  Set $t \leftarrow t+1$

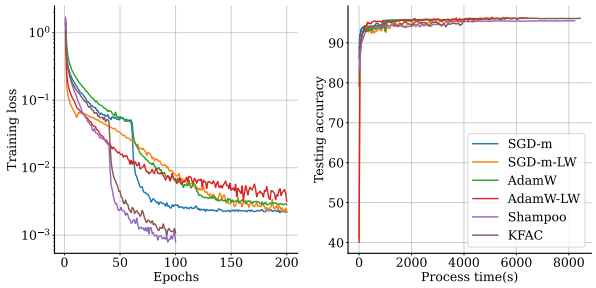
also use information about the local curvature of the loss function. We used the most popular version of Adam, AdamW [69] as a representative of adaptive first-order methods. Our experiments were run on a machine with one V100 GPU and eight Xeon Gold 6248 CPUs using PyTorch [70]. Each algorithm was run using the best hyper-parameters, determined by a grid search (the same grids specified in the previous chapter).

CNN problems: We first compared the performance of SGD-m-LW and AdamW-LW to SGD-m, Adam, KFAC and Shampoo on three CNN models, namely, VGG16 [68], ResNet-18 [72], and DenseNet [112], respectively, on the datasets CIFAR-10, CIFAR-100 and SVHN [73]. The first two have 50,000 training data and 10,000 testing data (used as the validation set in our experiments), while SVHN has 73,257 training data and 26,032 testing data. For all algorithms, we used a batch size of 512. In training, we applied data augmentation as described in [74], including random horizontal flip and random crop, since these setting choices have been used and endorsed in many



(a) CIFAR-10, VGG16

(b) CIFAR-100, ResNet-18



(c) SVHN, DenseNet

Figure 3.3: Performance of KFAC, Shampoo, Adam, Adam-LW, SGD-m and SGD-m-LW on three CNN problems.

previous research papers, e.g. [75, 76, 41]. (see the previous chapter for more details about the CNN experimental set-up)

All methods(except SGD-m-LW and AdamW-LW) employed a tunable learning rate(LR) schedule that decayed LR by a factor of 0.1 every K epochs, where K was set to 60 and 40, for the first-order methods, and second-order methods, respectively, on all problems. Moreover, weight decay, which has been shown to improve generalization across different optimizers [69, 75], was employed by all of the algorithms, and a grid search on the weight decay factor and the initial learning rate based on the criteria of maximal validation classification accuracy was performed. Finally, the damping parameter was set to 1e-8 for Adam (following common practice), and 0.03 for KFAC (`https://github.com/alecwangcq/KFAC-Pytorch`). For Shampoo, we set $\epsilon = 0.01$. For SGD-m-LW and AdamW-LW, we set $T = 20$ and $\beta_t = 0.99$.

From Figure **??**, we see that SGD-m-LW and AdamW-LW have a similar (and sometimes better) optimization and generalization performance compared to their fine-tuned versions(i.e SGD-m and

AdamW) and do not require any tuning procedure for the learning rate. Moreover, in terms of process time, SGD-m-LW and AdamW-LW are roughly 1.8 factor slower than SGD-m and AdamW and are competitive with all of the SOTA first and second-order methods in our experiments.

We also reported the computed adaptive step sizes per layer in Figure 3.4. There are two main interesting observations that we identified from these results: the adaptive learning rates have different scales across different layers and the values of the latter are stationary in the sense that one could "learn" a fixed "good" stationary learning rate for each layer without having to recompute it at every iteration.

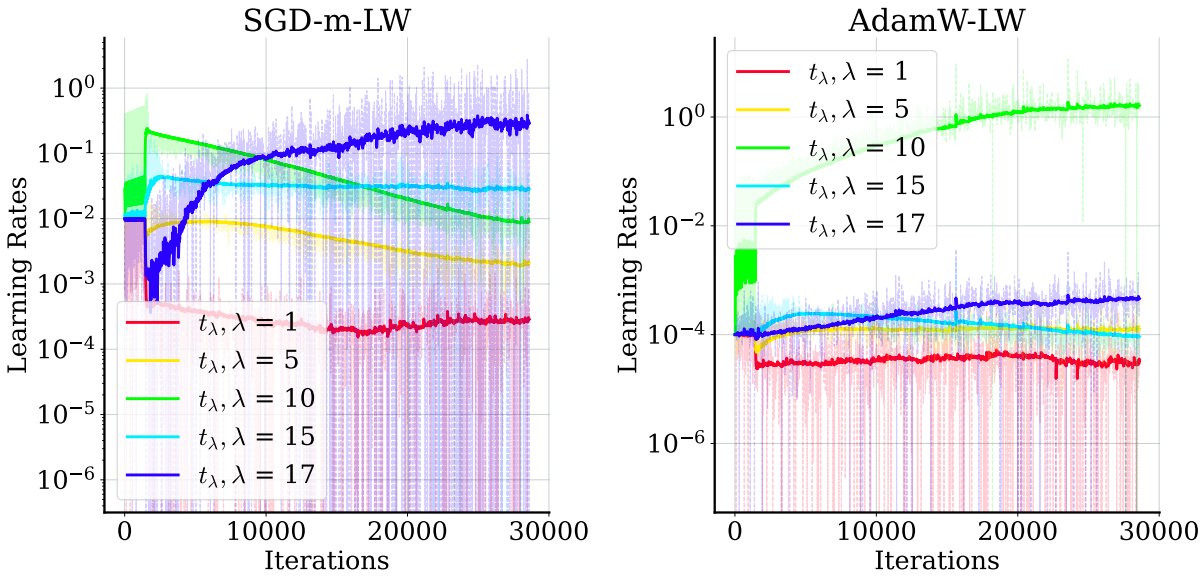


Figure 3.4: Layer-wise learning rates of SGD-LW and Adam-LW on CIFAR-10 VGG16 problem.

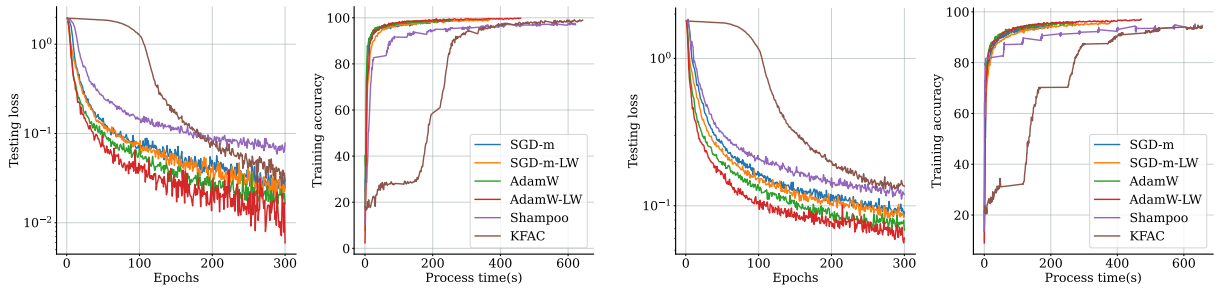
Graph Convolutional Networks (GCN) Problems: In this section, we compare the algorithms on a 3-layer GCN for the task of node classification in graphs applied to three citation datasets, Cora, CiteSeer, and PubMed(see [78]). In Table 3.1, nodes and edges correspond to documents and citation links, respectively, for these datasets. A sparse feature vector of document keywords, and a class label are associated with each node. For each dataset we used all of the nodes for training.

In our experiments, we used a 3-layer GCN with the following node-sizes $[I, 128, 64, O]$, where I and O are the numbers of input features and classes, respectively. In the first and second layers

Table 3.1: Citation network datasets statistics

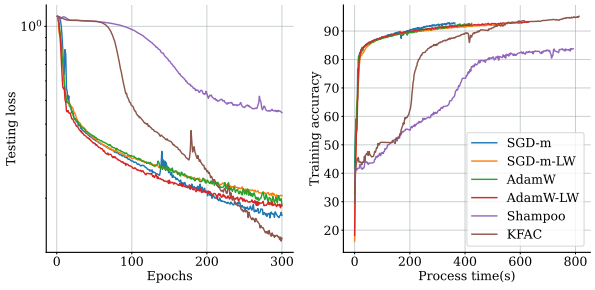
Dataset	Nodes	Edges	Classes	Features
Citeseer	3,327	4732	6	3,703
Cora	2,708	5,429	7	1,433
Pubmed	19,717	44,338	3	500

of this GCN, the activation function ReLU was followed by a dropout function with a rate of 0.5. The loss function was evaluated as the negative log-likelihood of Softmax of the last layer. The models were trained for 300 epochs. The hyperparameter search space was the same as that used for the CNN problems with no LR schedule and no weight decay, as in this section, we want to focus on optimization performance of the proposed methods. We set the inverse update frequency $T_2 = 25$ for KFAC and Shampoo and $T = 1$ for SGD-m-LW and AdamW-LW. From Figure 3.5, we see that SGD-m-LW and AdamW-LW were competitive with all of the SOTA first and second-order methods.



(a) Cora, GCN

(b) Citeseer, GCN



(c) Pubmedm, GCN

Figure 3.5: Performance of KFAC, Shampoo, Adam, Adam-LW, SGD-m and SGD-m-LW on three GCN problems.

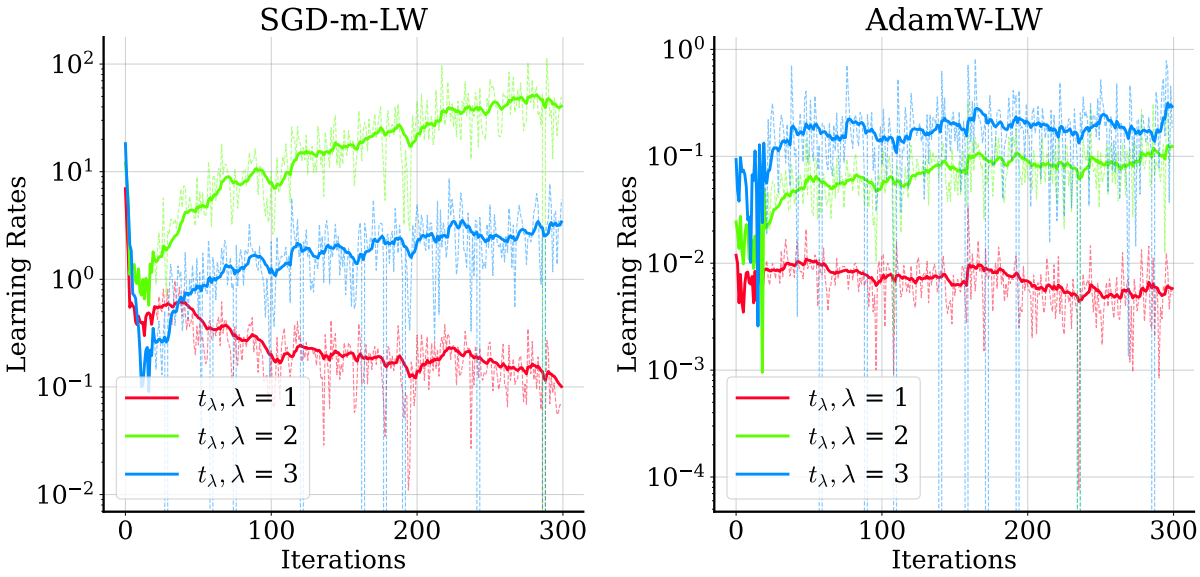


Figure 3.6: Layer-wise learning rates of SGD-LW and Adam-LW on Cora GCN problem.

3.6 Conclusion and next steps

We presented a layer-wise step-sizes procedure for first-order optimization methods that we believe is a valuable tool for fast and practical optimization without learning rate tuning, especially in DNN applications. Studying theoretical convergence guarantees of our method in DNNs which generate non-convex loss functions, and the convergence of our adaptive framework with other variants of SGD in the stochastic setting methods suggest interesting avenues for future research.

Chapter 4: Optimal Pricing with a Single Point

4.1 Introduction

Pricing is a central concept across a large spectrum of industries, ranging from e-commerce to transportation. A key informational dimension faced by decision-makers is the level of knowledge of customers' values. In classical settings in the literature, monopoly pricing problems are studied under the assumption that sellers have an accurate knowledge of consumer preferences through the value distribution (or the prior on values). In those cases, the seller may optimize pricing to maximize the expected revenues.

In practice, however, such information is rarely, if ever, available, and pricing must be conducted not based on the value distribution, but based on historical data. Typical historical data structures in the context of pricing include the prices posted and the responses of consumers observed at those prices: either a customer purchases or not. As a motivating example, consider an e-commerce firm that has been offering a product at an incumbent price w over the past quarter to a set of heterogeneous consumers, all with values drawn from a value distribution F . The firm observes the fraction of customers who have bought the offered product at the price w; in other words the firm has an estimate of the probability of sale or conversion rate, the fraction of customers whose values are greater than or equal to w, i.e. , an estimate of $\overline{F}(w) = q$ in $[0, 1]$. How should the seller decide on the pricing policy in the following quarter? Can the seller take advantage of the partial demand information extracted (conversion rate at w) to refine her pricing policy? Such historical data structures are commonplace in practice, and typically introduce different challenges. The number of past prices that were posted is often very limited and if one only accounts for recent data, can be as low as one, as in the example above. In other words, many historical data structures have very limited price dispersion. This renders elasticity-based price optimization very challenging

if not impossible in practice (without further experimentation) when trying to move from data to pricing decisions. A natural question is then if, in the absence of price dispersion, historical data is useful in any way in order to refine pricing decisions. The present chapter offers a resounding “yes" to this question and develops a framework to optimize prices given such limited data, and quantify the value of such data.

In more detail, we focus on a seller optimizing her pricing mechanism based on historical data with limited price dispersion. The seller does not know the value distribution of the buyer. She only knows that it belongs to some broad non-parametric class. In terms of the historical data, we anchor this chapter around the setting in which the seller has only access to the conversion rate $q = \overline{F}(w)$ at one historical price w, or potentially an interval I to which $\overline{F}(w)$ belongs. The questions the seller faces are then: what is an optimal pricing mechanism given the information at hand? And how valuable is the information/data at hand?

To answer these questions, we adopt a maximin ratio formulation in which performance is measured in comparison to the highest revenue the seller could have obtained with full information on the value distribution. The seller optimizes over general pricing mechanisms (we study both deterministic and randomized mechanisms). And nature may select any distribution in the class of interest to counter a pricing strategy. We are interested in characterizing the value of the maximin ratio as well as understanding the structure of optimal mechanisms. The latter quantifies the value of the collected information and the former offers concrete prescriptions.

This fundamental problem can be viewed as a foundational building block of offline data-driven pricing and the framework we will propose will be fairly general, enabling one to add information at other points in the future. Mathematically speaking, this leads to a problem in which the set of possible underlying value distributions is infinite dimensional, and so can be the set of possible pricing strategies (for randomized mechanisms). Hence, evaluating such an object is not possible without further understanding of structural properties of the problem.

Our main contributions lie in developing a general tractable characterization of deterministic and randomized *optimal* performances against any distribution in two widely used classes of

distributions: the class of regular distributions (distributions with increasing virtual values) and the class of monotone non-decreasing hazard rate (mhr) distributions. The latter is a subclass of the former and contains a wide variety of distributions (e.g., uniform, truncated normal, logistic, extreme value, exponential, subsets of Weibull, Gamma and Beta,...); [113] provides a review of the broad set of known subclasses of mhr distributions. The class of regular distributions further incorporates additional distributions (e.g., subsets of Pareto, log-normal, log-logistic,...); [114] provides an overview of such classes. Our analysis is general and the exact same analysis applies to regular and mhr distributions as special cases.

From a methodological perspective, our main contributions lie in a set of problem reductions that lead to *a closed form characterization of the maximin ratio for deterministic mechanisms*, and associated price prescriptions, and a sequence of *finite dimensional linear programs* that can approximate arbitrarily closely the maximin ratio for randomized mechanisms, leading to both optimal performance and associated near-optimal randomized mechanisms.

A first set of reductions lies in simplifying nature's optimization problem. As stated earlier, for any fixed mechanism chosen by the seller, nature's problem is an infinite dimensional problem over the class of regular (or mhr) distributions, which is non-convex. As such, evaluating the performance of a particular mechanism cannot be simply "brute-forced" numerically. As a first key reduction, we establish in Theorem 4.1 that against any mechanism, nature's worst-case optimization problem can be reduced from an infinite dimensional problem over a non-convex space to a one dimensional minimization problem over an interval. This reduction relies on exploiting the regularity (or mhr) structure to narrow down the set of candidate worst-cases to a "small" set.

Leveraging nature's problem reduction, we are able to derive a closed form (Theorem 4.2) for the maximin ratio for deterministic mechanisms against the classes of regular and mhr distributions. The results that one obtains through these closed forms highlight three different "regimes" of historical probability of sale values, and are quite illuminating with respect to the value associated with exact conversion rate information. Table 4.1 provides examples of such results. An example of a striking theoretical result is that, with knowledge of the median, it is possible to use a simple

deterministic pricing mechanism to guarantee a substantial fraction, 85.23%, of the oracle revenue when the value distribution is mhr; when the value distribution is regular, 66.62% of the oracle revenue can be guaranteed if we observe the 3rd quartile. Another highly notable theoretical result is associated with the value of low conversion rates that we uncover. We show that even if one only knows that 1% of customers purchase at a particular price, then a deterministic pricing mechanism guarantees more than 47% of oracle performance against mhr distribution and 18% against regular distributions. As a matter of fact, our closed form formulas indicate that, while the the maximin ratio converges to zero as the known conversion rate converges to zero, it does so at a supra-linear, very slow, rate: $\sqrt{q}$ for regular distributions, and $1/\log(q^{-1})$ for mhr distributions.

		Maximin ratio	
Distribution Class	Conversion Rate	Randomized mechanisms	Deterministic mechanisms
Regular	$\overline{F}(w) = 0.01$	31.12%	18.18%
	$\overline{F}(w) = 0.25$	67.75%	66.62%
	$\overline{F}(w) = 0.50$	55.99%	50.00%*
	$\overline{F}(w) = 0.75$	41.35%	25.00%
mhr	$\overline{F}(w) = 0.01$	51.17%	47.55%
	$\overline{F}(w) = 0.25$	74.71%	74.35%
	$\overline{F}(w) = 0.50$	85.30%	85.23%
	$\overline{F}(w) = 0.75$	64.14%	58.65%

Table 4.1: Maximin Performance: The table provides examples of the results obtained regarding the optimal performance one may achieve as a function of the admissible set of distributions and the class of pricing mechanisms one considers. The maximin ratio is characterized exactly for deterministic mechanisms and up to at most 1% error for randomized mechanisms. * indicates the only known result to date [115] .

In a second step, we study the performance of general randomized mechanisms. To characterize such performance, we first leverage the reduction above of Nature's problem, but also establish that one can focus on mechanisms with finite support, while controlling the potential losses in performance (Proposition 4.2). In turn, we are able to derive a sequence of linear programs with order N variables and order N constraints (Theorem 4.3) that yields: i.) an approximation to

the maximin ratio within $O(1/\sqrt{N})$ and $ii.$) provides a candidate randomized mechanism with near-optimal performance and support over order N points. Given these, one can evaluate for every history $(w, \{q\})$ the performance that the seller can achieve.

The above characterizes the theoretical developments needed to obtain an exact characterization of the maximin ratio for randomized mechanisms against regular or mhr distributions. The results we obtain through this analysis offer novel insights on the value of information and the additional value stemming from the expanded set of randomized pricing strategies, compared to deterministic ones. In particular, the value stemming from randomization is most prominent for values of the conversion rate close to 0 and 1. Intuitively, with historical prices providing less information, the seller can use randomization to counter uncertainty. For example, against regular distributions with a conversion rate of 1%, the seller can increase its guaranteed performance from about 18% with a deterministic price to 31% with a randomized mechanism, and with a conversion rate 75%, it can increase performance from 25% to about 41%.

Table 4.1 presents examples of the results obtained, but the framework developed is not specific to any probability of sale value and applies to any historical price and associated probability of sale. Figure 4.1 depicts the maximin ratio for randomized mechanisms for various values of the conversion rate ranging from 0.01 to 0.99.

We establish that randomization drastically affects the value that one can extract from information, leading to a fundamentally different rate of convergence of performance as q approaches zero or one. In that former regime, we establish that the rate convergence improves from order $\sqrt{q}$ to order $1/\log(1/q)$ (Proposition 4.3), and in the latter case, we show that it goes from a linear rate to order $1/\log(1/(1-q))$ (Proposition 4.4). We also show that for mhr distributions, while the rate of convergence is not affected by randomization around 0, the convergence rate is significantly affected for values of q that are close to 1, going from a linear rate to $1/\log(1/(1-q))$ (Proposition 4.3, Proposition 4.4). Table 4.2 summarizes these findings.

In addition, the framework we develop is general and allows to incorporate uncertainty (or noise) in the probability of sale estimate. We establish a parallel characterization for the maximin perfor-

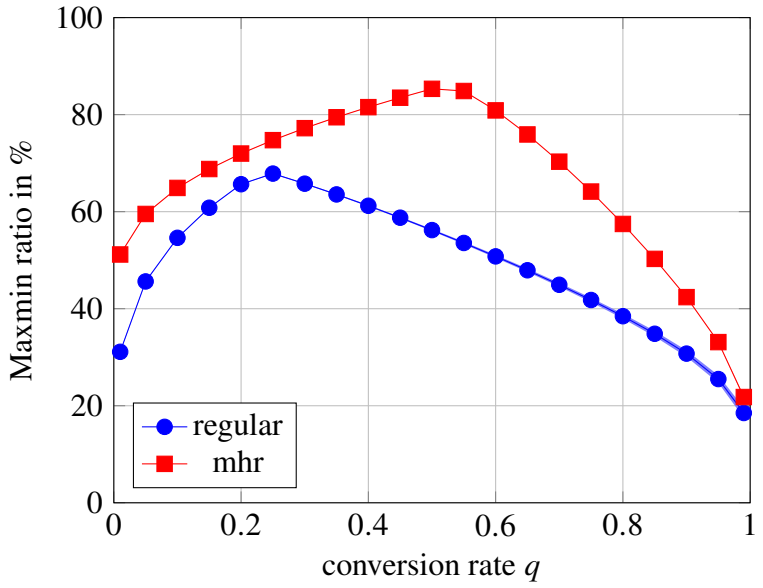


Figure 4.1: Maximin ratio for randomized mechanisms against regular and mhr distribution.

		Maximin ratio	
Distribution Class	regime	Randomized mechanisms	Deterministic mechanisms
regular	$q \to 0$	$\Theta\left(1/\log\left(1/q\right)\right)$	$\Theta\left(\sqrt{q}\right)$
	$q \to 1$	$\Theta\left(1/\log\left(1/(1-q)\right)\right)$	$\Theta\left(1-q\right)$
mhr	$q \to 0$	$\Theta\left(1/\log\left(1/q\right)\right)$	$\Theta\left(1/\log\left(1/q\right)\right)$
	$q \to 1$	$\Theta\left(1/\log\left(1/(1-q)\right)\right)$	$\Theta\left(1-q\right)$

Table 4.2: Maximin Performance rates: This table summarizes the rate of convergence to zero when the conversion rate q approaches 0 or 1 for the optimal randomized and deterministic mechanisms.

mance for randomized mechanisms in Theorem 4.4. We develop a sequence of finite dimensional linear programs that can approximate with arbitrary accuracy the maximin ratio.

Stepping back, the present chapter and associated framework enable to understand the value of one measurement for pricing purposes. While such a measurement provides very limited information about the entire value distribution, we establish that it provides significant value for pricing purposes. As such, this leads to an important building block for future research to better understand the value

associated with an arbitrary number of measurements, or the best way to experiment and collect such responses to prices.

Literature review: In this section, we position our work in the landscape of related past research efforts. Our work relates and contributes to the literature on data-driven pricing with limited value distribution knowledge. A setting that has been studied is one in which the seller only knows the support of the underlying distribution. Early studies are [42] and [43], in which the authors characterize the optimal pricing policy as well as the worst-case demand distribution with respect a min-max regret objective in the former and a competitive ratio in the latter. [44] studies a case in which the seller has access to the maximum price at which she would still expect non-zero demand, and the authors propose to use a simple deflation mechanism and characterize its performance against some subsets of parametric families. [116] characterize optimal pricing strategies in a dynamic setting where myopic or strategic customers arrive over time and only the support of their value distribution is known to the seller.

In contrast to this stream of work, we study the setting in which the seller has access to some information about the conversion rate at an incumbent price w, a typical data structure, and can adjust its decisions based on such information. We also allow value distributions with arbitrary support within central non-parametric classes (regular or mhr). [43] study a related setting and analyzes randomized mechanisms against general discrete distributions, but with known support bounds information. [117] study a general robust decision problem while specifying a shape-preserving set of univariate functions using a constrained B-spline approximation. The framework developed in [117] can be applied to pricing in environments with limited measurements and the authors illustrate their ideas using an optimal debt-settlement example. In [45], the authors studied the related problem of reconstructing demand curves when only a single point has been historically observed and showed how a second point can be extracted from the sales of discounted bundles and use it to estimate linear demand curve parameters. [46] document a setting at a large OEM where the problem is exactly one with no price dispersion in the historical data. There, the authors

applied some parametrization approach in conjunction with a robustification of prices. Recently, [118] study model-free assortment pricing decisions from transaction data, by leveraging incentive compatibility constraints.

In a related setting, [115] and [119] assume that the seller has only access to limited statistical information about the valuation distributions (such as the median, mean and variance). In [115] for the single-bidder pricing problem, which is related to our problem, the authors analyze the case with median information and provide a tight upper bound (50%) on the best achievable competitive ratio for regular distribution using deterministic mechanisms. This can be seen as a special case of the general framework we develop. Our results establish the exact performance of deterministic for *any* probability of sale for both regular and mhr distributions, establishing two phase transitions (small, moderate, and high probability of sale). For these settings, we also characterize the optimal performance of randomized mechanisms. While studying a different set of questions (the performance of a Vickrey auction with duplicate bidders), [120] also considers a setting in which the information available to the seller consists of a percentile of the value distribution, akin to the information structure in the present chapter.

A number of studies look at how to collect and incorporate data on the fly for pricing purposes, in which case an exploration-exploitation trade-off emerges. See [52], [53], [54], [55], [56]. The present study establishes that when initial offline data is available (such as in [55] or [56]), it can be possible to exploit such information, even with no price variability in the data. As such, the ideas presented here might also have implications for dynamic learning algorithms.

An alternative data structure that has received attention is one based on samples of the value distribution, as opposed to buy/no buy feedback. [47] studies the sample complexity needed to achieve near-optimal performance; see also, e.g., [121], [122], in the context of auctions. The setting where the seller has access to a limited number of observed samples has also been studied; see [47], [48], [49], [123], [50], and [51]. These studies demonstrate that a few samples can be very informative for pricing purposes. Relatedly, the present work characterizes the value that a different type of information/data, a single percentile, has for pricing purposes.

4.2 Problem formulation and approach overview

We consider a seller trying to sell one indivisible good to one buyer. We assume that the buyer's value v is drawn from some distribution F with support included in $[0, \infty)$. The seller does not know F and only knows some class information as well as partial information associated with it based on the historical conversion rate observed at a price w. More specifically, we study the setting in which the seller knows that the probability of sale belongs to some interval, i.e., $\overline{F}(w)$ belong s to I with $I \subseteq [0, 1]$. An important building block is when the decision-maker knows the exact probability of sale q at w, i.e., the seller knows that $\overline{F}(w) = 1 - F(w) = q$. In what follows, we will use the notation $\overline{F} := 1 - F$ to denote the complementary cumulative distribution function (ccdf).

The problem we are interested in is the following: how can the seller leverage the information observed at the price w to maximize her revenue. More formally, we model the problem as a game between nature and the seller, in which the seller selects a selling mechanism and nature may choose any admissible distribution F that is consistent with the observed information. We denote by $\mathcal{D}$ the set of cumulative distribution functions (cdf) on $[0, +\infty)$, i.e., the set of non-decreasing right continuous with left limits functions from $[0, +\infty)$ into $[0, 1]$ such that the limit at infinity is one.

Pricing and performance: A (potentially) randomized pricing strategy will be characterized by the cdf of prices the seller posts. We let $\mathcal{P} = \{\Psi \text{ in } \mathcal{D}\}$ to be the set of randomized prices that a mechanism can choose from, given the conversion rate information.

The expected revenue of the seller using a price distribution Ψ in $\mathcal{P}$, if nature is selecting a distribution F, is given by

$$\int_0^\infty \left[\int_0^\infty p \mathbf{1}\{v \geq p\} dF(v) \right] d\Psi(p) = \int_0^\infty p\overline{F}(p) d\Psi(p) = \int_0^\infty Rev\left(p|\overline{F}\right) d\Psi(p),$$

where we introduce the notation

$$Rev\left(p|\overline{F}\right) = p\overline{F}(p).$$

We define $\text{opt}(F)$ to be the maximal performance one could achieve *with knowledge* of the exact distribution of buyer's values. It is known that it is a posted price [124], and is given by

$$\text{opt}(F) := \sup_{p \geq 0} Rev\left(p|\overline{F}\right). \tag{4.2.1}$$

For an arbitrary distribution Ψ in $\mathcal{P}$, we define its performance against a distribution F such that $\text{opt}(F) > 0$ as follows

$$R(\Psi, F) = \frac{\int_0^\infty Rev\left(p|\overline{F}\right) d\Psi(p)}{\text{opt}(F)}.$$

Let $\mathcal{G}(w, I)$ denote the set of distributions with support included in $[0, \infty)$ with finite and non-zero expectation such that $\overline{F}(w)$ belongs to I where I is an interval in $[0, 1]$, i.e.,

$$\mathcal{G}(w, I) = \left\{F : [0, \infty) \to [0, 1] : F \text{ is in } \mathcal{D} \text{ and } 0 < \mathbb{E}_F[v] < \infty \text{ and } \overline{F}(w) \text{ in } I\right\}. \tag{4.2.2}$$

Note that $\text{opt}(F)$ is in $(0, \infty)$ for all F in $\mathcal{G}(w, I)$ and hence the ratio $R(\Psi, F)$ is well defined for any element of the class $\mathcal{G}(w, I)$. For an arbitrary price distribution Ψ in $\mathcal{P}$ and for a subclass $\mathcal{F} \subseteq \mathcal{G}(w, I)$, we define nature's problem as:

$$\inf_{F \in \mathcal{F}} R(\Psi, F).$$

The objective in this chapter is to characterize the maximin ratio for sub-classes of distributions $\mathcal{F} \subseteq \mathcal{G}(w, I)$ and subclasses $\mathcal{P}' \subseteq \mathcal{P}$

$$\mathcal{R}(\mathcal{P}', \mathcal{F}) = \sup_{\Psi \in \mathcal{P}'} \inf_{F \in \mathcal{F}} R(\Psi, F).$$

Note that this objective is always between 0 and 1 and can be interpreted as a measurement of the "value of information" when using the subclass $\mathcal{P}'$.

Mechanisms classes. We will be interested in the performance of general randomized mechanisms $\mathcal{P}$ but also with the performance associated with the subclass of deterministic pricing mechanisms $\mathcal{P}_d \subset \mathcal{P}$, defined as the set of dirac delta, i.e.,

$$\mathcal{P}_d = \{\delta_\gamma : \gamma \geq 0\}.$$

Focal classes of distributions. Recall the definition of the set of general distributions consistent with the data, $\mathcal{G}(w, I)$, given in Equation (4.2.2). Proposition 4.1 below formalizes that it is impossible to design any randomized mechanism with a positive competitive ratio when competing against $\mathcal{G}(w, I)$ for any non-empty interval I.

Proposition 4.1 (maximin ratio against general distributions)**.** *For any mechanism* Ψ *in* $\mathcal{P}$*, and non-empty interval* I *in* $[0, 1]$*, we have*

$$\inf_{F \in \mathcal{G}(w,I)} R(\Psi, F) = 0.$$

In the rest of the chapter, we focus on widely studied subclasses of $\mathcal{G}(w, I)$ in the pricing context. In particular, we focus on two broad subclasses. The first subclass we analyze is the class of monotone hazard rate (mhr) distributions, i.e., distributions that admit a density, except potentially at the maximum of their support, and that have a non-decreasing hazard rate. As mentioned in the introduction, this class contains a wide variety of distributions and typical models fitted in the literature belong to subclasses of mhr distributions. See, e.g., [113].

A second notable class of distributions that generalizes mhr distributions is the class of *regular* distributions; these admit a density, except potentially at the maximum of their support, and have a non-decreasing virtual value $v - \overline{F}(v)/f(v)$. This class of distributions contains all mhr distributions but also a host of additional distributions; see, e.g., [114] for a summary of widely used regular distributions. In particular, the class of regular distributions allows for heavier tails than mhr distributions. This class is central to the pricing and mechanism design literatures and can be

alternatively described as the class of distributions which induce a concave revenue function in the quantity space.

While we will focus on the two classes above given their central role in the literature, our analysis will be unified. In particular, the two classes above can be seen as special cases of α-strongly regular distributions (see, e.g., [114], [121], [125]). These are distributions with positive density function f on its support $[a, b]$, where $0 \leq a < \infty$ and $a \leq b \leq \infty$, such that $(1-\alpha)v - \overline{F}(v)/f(v)$ is non-increasing. When $\alpha = 0$, this corresponds to regular distributions and when $\alpha = 1$, this corresponds to mhr distributions. We define

$$\mathcal{F}_\alpha(w, I) \quad = \quad \{F \text{ in } \mathcal{G}(w, I) : F \text{ is } \alpha\text{-strongly regular}\}$$

to be the set of distributions that are α-strongly regular *and* consistent with the information at hand.

It is possible to establish that when the interval I contains 0 or 1, no pricing mechanism can guarantee a positive fraction of revenues. We formalize this result in Lemma 4.B-3 (presented in Appendix 4.B). We assume throughout that the interval I does not contain 0 or 1, i.e., $I \cap \{0, 1\} = \emptyset$.

Approach overview: We start by analyzing the case when the seller has access to the probability of sale at one price, i.e., $I = \{q\}$ is a singleton, with q in $(0, 1)$, and its associated price w. This will be the focus of Sections 4.3-4.5. In what follows, whenever the percentile is known, we will use, with some abuse of notation, $\mathcal{F}_\alpha(w, q)$ instead of $\mathcal{F}_\alpha(w, \{q\})$. We return in Section 4.6 to the case with interval uncertainty.

The first step in analyzing $\mathcal{R}(\mathcal{P}', \mathcal{F}_\alpha(w, q))$ resides in noting that it can reformulated as an equivalent mathematical program

$$\begin{aligned} \sup_{\Psi(\cdot) \text{ in } \mathcal{P}', c \text{ in } [0,1]} \quad & c & (\mathcal{MP}) \\ s.t. \quad & \int_0^\infty Rev\left(p|\overline{F}\right) d\Psi(p) \geq c \ \text{opt}(F) \qquad \text{for all } F \text{ in } \mathcal{F}_\alpha(w, \{q\}). \end{aligned}$$

The value of this problem is exactly equal to the maximin ratio $\mathcal{R}(\mathcal{P}', \mathcal{F}_\alpha(w, q))$ and any optimal solution to the former is also optimal for $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q))$. When $\mathcal{P}' = \mathcal{P}$, this is a linear program. However, the key challenge in solving such a problem and designing optimal or near-optimal pricing mechanisms resides in the fact both $\mathcal{P}$ and $\mathcal{F}_\alpha(w, q)$ are infinite dimensional spaces. In turn, this is a linear program with an infinite number of variables and constraints. (When $\mathcal{P}' = \mathcal{P}_d$, the set of feasible mechanisms is not convex anymore.)

To characterize $\mathcal{R}(\mathcal{P}', \mathcal{F}_\alpha(w, q))$, we will proceed in two steps. We first establish in Section 4.3 a key reduction, that many of the constraints are "redundant" and as a result, one can restrict attention, without loss of optimality to an alternative to $(\mathcal{MP})$ with significantly fewer constraints.

For deterministic mechanisms $\mathcal{P}_d$, analyzed in Section 4.4, we leverage the fundamental reduction in the space of distributions to establish that the problem can be directly reframed and solved in closed form. In turn, optimal deterministic mechanisms and optimal performance over this subclass of mechanisms can be derived explicitly against regular and mhr distributions.

In Section 4.5, we tackle the challenge stemming from the infinite dimensional nature of the space of mechanisms of the seller in the context of general randomized mechanisms. For that, we establish that mechanisms with bounded and discrete support, can approximate (from below) the performance of general randomized mechanisms with arbitrarily high accuracy. We combine the reductions in both the space of distributions and mechanisms to derive a sequence of finite dimensional linear programs whose value converges (from below) to the original quantity of interest, $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q))$. Furthermore, the optimal solution of any such linear program provides a (discrete) pricing distribution with a certificate of performance given by the value of the linear program and this value approaches the optimal maximin ratio. In Section 4.6, we extend the ideas above to the case when the probability of sale is only known to belong to an interval I and characterize optimal performance in this more general case.

Notation. With some abuse, to avoid introducing special notation at various junctions, we will interpret any ratio of a positive quantity divided by zero as ∞. Furthermore, we will use the notation $a \vee b := \max\{a, b\}$ and $a \wedge b := \min\{a, b\}$.

4.3 Reduction of Nature's problem

In this section, we focus on Nature's problem associated with selecting a worst-case distribution against an arbitrary mechanism. For any mechanism Ψ in $\mathcal{P}$, Nature will select a worst-case distribution in the non-convex infinite dimensional space of distributions $\mathcal{F}_\alpha(w, q)$. Our first main result establishes a fundamental reduction: one may restrict attention to a "small" set of candidate worst-case distributions. In particular, we will establish that Nature's problem can be reduced to a one-dimensional optimization problem.

For any α in $[0, 1]$, we introduce notation for Generalized Pareto Distributions (GPD). This class of distributions plays a central role in pricing problems in the context of α-strongly regular distributions (see, e.g., [121] and [125]). Indeed, the α-strongly regularity condition can be interpreted as a curvature restriction captured by the fact that α virtual value function $(1-\alpha)v - f(v)/\overline{F}(v)$ is non-increasing. For a given value of α, the function defined below, Γ_α, can be seen to be on the "boundary" of this space as it has constant α virtual value function. In particular, for any $v \geq 0$, we define

$$\Gamma_\alpha(v) \quad = \quad \begin{cases} (1 + (1-\alpha)\ v)^{-1/(1-\alpha)} & \text{if } \alpha \text{ in } [0, 1), \\ e^{-v} & \text{if } \alpha = 1. \end{cases}$$

In addition, Γ_α^{-1} denotes the inverse of Γ_α and we set $\Gamma_\alpha^{-1}(0) := +\infty$ and $\Gamma_\alpha(+\infty) := 0$.

We next introduce some notation that will allow us to define an appropriate subclass of distributions.

For any pair of values (s, s') such that $0 \leq s \leq s'$ and $1 \geq q_s \geq q_{s'} > 0$, and for any $t \geq s'$, we define on $[0, \infty)$

$$\overline{G}_{\alpha,t}(v|(s,q_s),(s',q_{s'})) = \begin{cases} \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_s)\,\frac{v}{s}\right), & \text{if } v \in [0,s), \\ q_s\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)\frac{v-s}{s'-s}\right) & \text{if } v \in [s,t], \\ 0 & \text{if } v > t. \end{cases} \tag{4.3.1}$$

The function $\overline{G}_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ is a complementary cumulative distribution function (ccdf) that has conversion rate q_s at price s and $q_{s'}$ at price s'[1] and satisfies the restriction on the curvature implied by α-strong regularity with equality locally, on $[0, s)$, and on $[s, t]$. Furthermore, it has support $[0, t]$.

We next define the following family of distributions, through their complementary cumulative distribution function

$$\overline{F}_\alpha(v|r,(w,q)) := \begin{cases} \overline{G}_{\alpha,w}(v|(r,1),(w,q)) & \text{if } r \text{ in } [0,w), \\ \overline{G}_{\alpha,r}(v|(0,1),(w,q)) & \text{if } r \geq w. \end{cases} \tag{4.3.2}$$

The associated cdf $F_\alpha(v|r,(w,q))$ has the upper end of its support at $r \vee w$ and the lower end of its support at either 0 if $r > w$ or r if $r \leq w$, and has a conversion rate of q at w. Figure 4.2 depicts this distribution for two sets of parameters. This corresponds to a family of translated and truncated GPD distributions. Let

$$\underline{r}_\alpha(w,q) = \frac{w}{\Gamma_\alpha^{-1}(q)+1}, \quad \overline{r}_\alpha(w,q) = \frac{w}{\alpha\Gamma_\alpha^{-1}(q)}, \text{ with } \overline{r}_\alpha(w,q) = +\infty \text{ for } \alpha = 0. \tag{4.3.3}$$

[1]Note that when $s = s'$ and $q_s > q_{s'}$, the ccdf $\overline{G}_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ has a mass of $1 - q_s$ at $s = s'$. In this case, with some abuse of terminology, we continue to say that it has a conversion rate of $q_{s'}$ at s' as it can be approximated arbitrarily closely by a ccdf that has this property.

We are now in a position to define the following subset of distributions, which is parametrized by a single parameter r:

$$\mathcal{S}_{\alpha,w,q} \quad = \quad \left\{F_\alpha(\cdot|r,(w,q)) : r \text{ in } [\underline{r}_\alpha(w,q),w) \cup [w,\overline{r}_\alpha(w,q)]\right\}, \tag{4.3.4}$$

where we use the convention that whenever $\overline{r}_\alpha(w,q) < w$, $[w,\overline{r}_\alpha(w,q)] := \emptyset$.

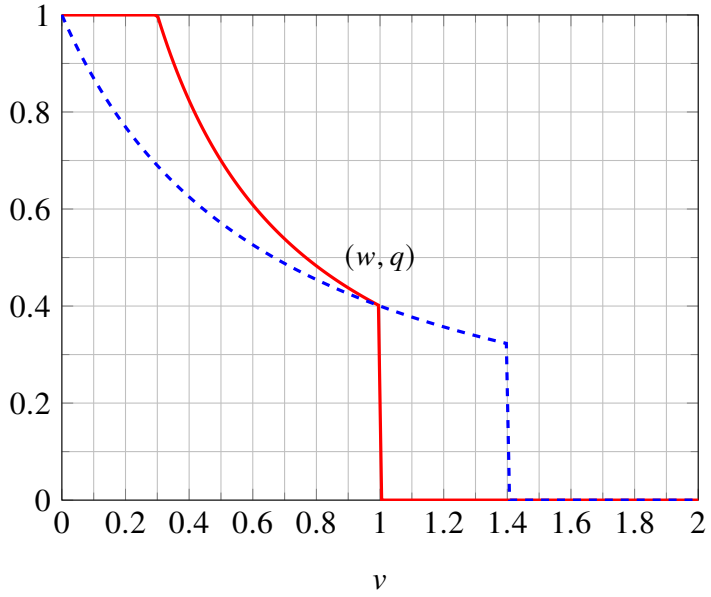


Figure 4.2: Examples of distributions in $\mathcal{S}_{\alpha,w,q}$: The figure depicts examples of functions $\overline{F}_\alpha(\cdot|r,(w,q))$: $\overline{F}_0(\cdot|0.3,(1,0.4))$ in red and $\overline{F}_0(\cdot|1.5,(1,0.4))$ in dashed blue.

It is possible to establish that $\mathcal{S}_{\alpha,w,q} \subset \mathcal{F}_\alpha(w,q)$ as every element of this set has constant α-virtual value on the interior of its support (see Lemma 4.A-1).

We next state our first main result.

Theorem 4.1 (Fundamental Reduction)**.** *Fix α in $[0,1]$. For any q in $(0,1)$, for any subset of mechanisms $\mathcal{P}' \subseteq \mathcal{P}$,*

$$\mathcal{R}(\mathcal{P}',\mathcal{F}_\alpha(w,q)) = \mathcal{R}(\mathcal{P}',\mathcal{S}_{\alpha,w,q}).$$

This result provides a central structural property for this class of problems. When analyzing the possible response of nature to a particular mechanism, it is sufficient to only consider translated

and truncated GPD families with a special structure. In particular, the candidate worst-cases are parametrized by a single parameter r as outlined in the definition of $\mathcal{S}_{\alpha,w,q}$. The only candidate worst-cases to consider are either: i.) distributions whose lower bound starts at some r in $[0, w]$, decreases according to a GPD piece up to w and admit a mass at w; for those, the optimal oracle price is r (cf. Lemma 4.C-2); or ii.) distributions that have a support starting at zero, decrease according to a GPD, and admit a mass at $r \geq w$; for those distributions, the optimal oracle price is again at r, but exceeds w. Intuitively, these distributions capture exactly the difficulty of not knowing the distribution of values. Indeed, when fixing any mechanism, the result above implies that one "can think" of nature as selecting an optimal oracle price as opposed to a distribution, as conditional on the former, one can now compute the worst-case distribution. As we will see later, this structural result will be central to characterize optimal performance and derive near-optimal mechanisms for both the classes of deterministic and randomized mechanisms.

Based on the fundamental reduction in Theorem 4.1, the set of constraints in $(\mathcal{MP})$ can be significantly reduced and the problem can be equivalently stated as follows

$$\begin{aligned} \sup_{\Psi\in\mathcal{P}',c\in[0,1]} \quad & c \\ s.t. \quad & \int_0^w \frac{Rev\left(u|\overline{G}_{\alpha,w}(\cdot|(r,1),(w,q))\right)}{Rev\left(r|\overline{G}_{\alpha,w}(\cdot|(r,1),(w,q))\right)} d\Psi(u) \geq c, \text{for all } r \text{ in } \left[\underline{r}_\alpha(w,q), w\right) \\ & \int_0^r \frac{Rev\left(u|\overline{G}_{\alpha,r}(\cdot|(0,1),(w,q))\right)}{Rev\left(r|\overline{G}_{\alpha,r}(\cdot|(0,1),(w,q))\right)} d\Psi(u) \geq c, \text{for all } r \text{ in } \left[w, \overline{r}_\alpha(w,q)\right]. \end{aligned}$$

We note that Theorem 4.1 admits a generalization to the case in which q is only known to belong to an interval (cf. Proposition 4.F-1).

Next, we present the proof of Theorem 4.1 together with the intuition associated with the various steps that enable this fundamental reduction.

Key ideas and proof of Theorem 4.1 We fix a mechanism Ψ throughout. The proof is organized around two main steps. In a first step, we show that nature's optimization problem can be reduced to a two-dimensional optimization problem, one of selecting the location the optimal oracle price r_F and the corresponding "quantity" q_F. In other words, these two quantities can be seen as "sufficient statistics" from the perspective of nature, given the limited knowledge of the seller. This step is enabled by using the local "extremality" of generalized pareto distributions in the set of α-strongly regular distributions. In a second step, we establish that for a given value of r_F, the worst-case q_F can be characterized explicitly, and in turn, one can further reduce the problem to a one-dimensional optimization problem, over the set of possible oracle optimal prices r_F. In particular, we establish that one can reduce attention to translated and truncated generalized pareto distributions.

Step 1. Fix q in $(0, 1)$. In this first step, we develop a reduction of nature's problem to a two-dimensional optimization problem parametrized by the set of possible values that the optimal oracle price r_F and quantity q_F can take. We first define the set of feasible values for r_F and q_F given the information at hand. To that end, let $\mathcal{B}_\alpha(w, q)$ denote the set of feasible pairs, i.e.,

$$\mathcal{B}_\alpha(w,q) \quad := \quad \{(r^*, q^*) \text{ in } \mathbb{R}_+ \times [0,1] : \text{there exists } \ F \text{ in } \mathcal{F}_\alpha(w,q) \text{ with } r_F = r^*, q_F = q^*\}\,.$$

Given such a definition, we have

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi, F) &= \inf_{F\in\mathcal{F}_\alpha(w,q)} \frac{\mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)]}{\text{opt}(F)} \\
&= \inf_{(r^*,q^*)\in\mathcal{B}_\alpha(w,q)} \inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \frac{\mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)]}{\text{opt}(F)} \\
&= \inf_{(r^*,q^*)\in\mathcal{B}_\alpha(w,q)} \frac{1}{r^*q^*} \inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)]. \qquad (4.3.5)
\end{aligned}$$

The reduction above has allowed to “decouple" the problem, where the denominator is fully controlled and the numerator can be minimized in the inner minimization, independently of the denominator.

Fix (r^*, q^*) in $\mathcal{B}_\alpha(w, q)$. Next, we characterize $\inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)]$. We first derive a lower bound.

$$
\begin{aligned}
& \inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)] \\
= & \inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \int_0^\infty Rev\left(p|\overline{F}\right) d\Psi(p) \\
\geq & \inf_{\substack{F\in\mathcal{F}_\alpha(w,q):\\ r_F=r^*, q_F=q^*}} \left[\int_0^{r^*\wedge w} Rev\left(p|\overline{F}\right) d\Psi(p) + \int_{r^*\wedge w}^{r^*\vee w} Rev\left(p|\overline{F}\right) d\Psi(p)\right].
\end{aligned}
$$

Next, we leverage the following single crossing property result from Lemma 2 in [51].

Lemma 4.1 (Single Crossing Property)**.** *Fix α in $[0, 1]$, F in $\mathcal{F}_\alpha$ and a pair of values (s, s') such that $0 \leq s \leq s'$ and $q_{s'} = \overline{F}(s') > 0$. Then*

$$
\begin{aligned}
\overline{F}(v) &\geq q_s \Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)\frac{v-s}{s'-s}\right) \qquad \text{if } v \text{ in } [s, s'].\\
\overline{F}(v) &\leq q_s \Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)\frac{v-s}{s'-s}\right) \qquad \text{if } v \text{ in } (s', +\infty).
\end{aligned}
$$

Lemma 4.1 provides a systematic way to obtain local lower and upper bounds on the ccdf of any α-regular distribution as a function of $\overline{H}_\alpha(\cdot)$. The bound coincides with the original function at the extreme points of the interval $[s, s']$, and provides a lower bound on the interval $[s, s']$ and an upper bound on $[s', +\infty)$ that coincides with the function at s'. Furthermore, the bounds are only parameterized by α and the quantiles at the interval extremes. For further intuition about this lemma, we refer the reader to [51]. Applying Lemma 4.1 to the pairs $(s, s') = (0, r^* \wedge w)$ and

$(s, s') = (r^* \wedge w, r^* \vee w)$, we obtain the following lower bound

$$
\begin{aligned}
& \inf_{\substack{F \in \mathcal{F}_\alpha(w,q): \\ r_F = r^*, q_F = q^*}} \mathbb{E}_\Psi[Rev\left(p|\overline{F}\right)] \\
\geq & \int_0^{r^* \wedge w} p\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(q^* \vee q\right)\frac{v}{r^* \wedge w}\right) d\Psi(p) \\
& + \mathbb{1}_{r^* \neq w} \int_{r^* \wedge w}^{r^* \vee w} (q^* \vee q)\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^* \wedge q}{q^* \vee q}\right)\frac{v - r^* \wedge w}{r^* \vee w - r^* \wedge w}\right) d\Psi(p) \\
= & \mathbb{E}_\Psi\left[p\, \overline{G}_{\alpha, r^* \vee w}(p|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q))\right] \\
= & \mathbb{E}_\Psi[Rev\left(p|\overline{G}_{\alpha, r^* \vee w}\left(\cdot|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q)\right)\right)],
\end{aligned}
$$

where we used the definition of $\overline{G}_{\alpha, r^* \vee w}(p|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q))$ given in (4.3.1).

In Lemma 4.C-3, stated and proved in Section 4.C, we establish that the distributions $\overline{G}_{\alpha, r^* \vee w}(p|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q))$ always belong to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$. In turn, this implies that the inequality above is tight. Returning to (4.3.5), we have established that the problem of nature can be written as

$$
\begin{aligned}
& \inf_{F \in \mathcal{F}_\alpha(w,q)} R(\Psi, F) = \\
& \inf_{(r^*, q^*) \in \mathcal{B}_\alpha(w,q)} \int_0^\infty \frac{Rev\left(p|\overline{G}_{\alpha, r^* \vee w}\left(\cdot|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q)\right)\right)}{Rev\left(r^*|\overline{G}_{\alpha, r^* \vee w}\left(\cdot|(r^* \wedge w, q^* \vee q), (r^* \vee w, q^* \wedge q)\right)\right)} d\Psi(p) \quad (4.3.6)
\end{aligned}
$$

In other words, conditional on $r_F = r^*$ and $q_F = q^*$, one can pin down the worst-case distribution, and associated revenue curve. In Figure 4.3, we illustrate the construction of the worst-case revenue curves.

Step 2. In a second step, we will further reduce the minimization problem stated in Equation (4.3.6) to a one dimensional minimization problem by solving exactly for the worst-case q^* across instances. To that end, we first develop an alternative characterization of the set $\mathcal{B}_\alpha(w,q)$. In particular, in

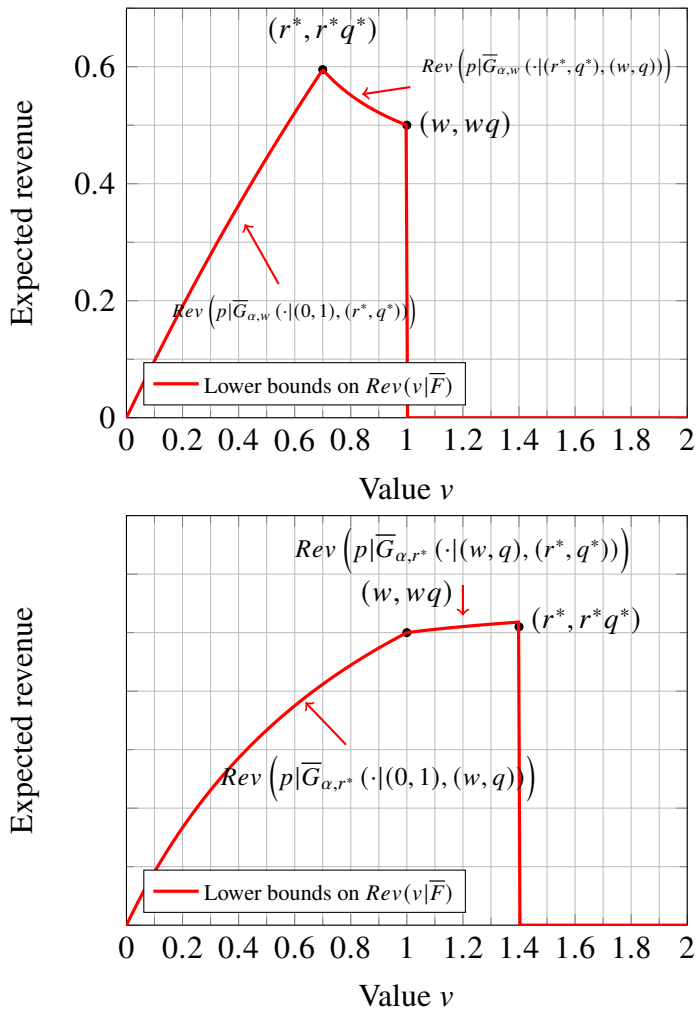


Figure 4.3: Parametrized worst-case revenue curves: The figure depicts, conditional on the optimal oracle price r^* and revenue r^*q^*, the worst-case revenue functions obtained in the proof using the single crossing property Lemma 4.1. The left panel corresponds to a case where $r^* < w$ and the right one to a case $r^* > w$. For these figures, α is set to zero.

Lemma 4.C-4, stated and proved in Section 4.C, we establish that $\mathcal{B}_\alpha(w,q) = \mathcal{B}_l \cup \mathcal{B}_h$, where

$$\mathcal{B}_l = \left\{(r^*,q^*) \text{ in } [0,w)\times[0,1] :\ q^* \geq \max\left\{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right), \Gamma_\alpha\left(\frac{1}{\alpha}\right), \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)}\right\}\right\}$$

$$\mathcal{B}_h = \left\{(r^*,q^*) \text{ in } [w,+\infty)\times[0,1] :\ q^* \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right),\ q^* \geq q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right)\right\}.$$

Next, define the function for each r^*, w, q,

$$\widetilde{R}_{r^*,w,q} : q^* \;\mapsto\; \int_0^{\infty} \frac{Rev\left(p|\overline{G}_{\alpha,r^*\vee w}\left(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q)\right)\right)}{Rev\left(r^*|\overline{G}_{\alpha,r^*\vee w}\left(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q)\right)\right)} d\Psi(p).$$

By (4.3.6), and the definition of $\widetilde{R}_{r^*,w,q}(\cdot)$, we have

$$\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) \;=\; \min\left\{\inf_{(r^*,q^*)\in\mathcal{B}_l} \widetilde{R}_{r^*,w,q}(q^*), \inf_{(r^*,q^*)\in\mathcal{B}_h} \widetilde{R}_{r^*,w,q}(q^*)\right\}.$$

In Lemma 4.C-5, stated and proved in Section 4.C, we establish the following monotonicity result: for any r^* that is consistent with a pair in $\mathcal{B}_\alpha(w,q)$, i.e, that belongs to

$$J_{w,q} = \{r \text{ s.t. there exists } q^* \text{ s.t. } (r^*,q^*) \text{ is in } \mathcal{B}_\alpha(w,q)\},$$

the function $\widetilde{R}_{r^*,w,q}(\cdot)$ is decreasing in the set $\{q^* : (r^*,q^*) \text{ is in } \mathcal{B}_\alpha(w,q)\}$. This monotonicity, in conjunction with the explicit characterization of the sets $\mathcal{B}_l$ and $\mathcal{B}_h$, implies that that if $r^* < w$, fixing a feasible r^*, the worst-value of q^* is 1; and if $r^* \geq w$, fixing a feasible r^*, the worst-case value of q^* is $\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)$.

In turn, the problem reduces to finding the domain of possible values of r^*, i.e., characterizing $J_{w,q}$. We show in Lemma 4.C-5 that, for $r^* \leq w$, there exists a value q^* such that $(r^*,q^*) \in \mathcal{B}_l$ if and only if $r^* \geq \underline{r}_\alpha(w,q)$. Hence

$$\inf_{(r^*,q^*)\in\mathcal{B}_l} \widetilde{R}_{r^*,w,q}(q^*) = \inf_{r^*\in\left[\underline{r}_\alpha(w,q),w\right)} \int_0^{w} \frac{Rev\left(p|\overline{G}_{\alpha,w}\left(\cdot|(r^*,1),(w,q)\right)\right)}{Rev\left(r^*|\overline{G}_{\alpha,w}\left(\cdot|(r^*,1),(w,q)\right)\right)} d\Psi(p).$$

In turn, we show in Lemma 4.C-5 that for $r^* > w$, there exists a value q^* such that $(r^*, q^*) \in \mathcal{B}_h$ if and only if $r^* \leq \overline{r}_\alpha(w, q)$. Hence

$$\inf_{(r^*,q^*)\in\mathcal{B}_h} \widetilde{R}_{r^*,w,q}(q^*) = \inf_{r^*\in[w,\overline{r}_\alpha(w,q)]} \int_0^{r^*} \frac{Rev\left(p|\overline{G}_{\alpha,r^*}(\cdot|(0,1),(w,q))\right)}{Rev\left(r^*|\overline{G}_{\alpha,r^*}(\cdot|(0,1),(w,q))\right)} d\Psi(p).$$

Therefore we have established

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi, F) &= \min\Bigg\{ \inf_{r^*\in[\underline{r}_\alpha(w,q),w)} \int_0^{w} \frac{Rev\left(p|\overline{G}_{\alpha,w}(\cdot|(r^*,1),(w,q))\right)}{Rev\left(r^*|\overline{G}_{\alpha,w}(\cdot|(r^*,1),(w,q))\right)} d\Psi(p), \\
&\qquad \inf_{r^*\in[w,\overline{r}_\alpha(w,q)]} \int_0^{r^*} \frac{Rev\left(p|\overline{G}_{\alpha,r^*}(\cdot|(0,1),(w,q))\right)}{Rev\left(r^*|\overline{G}_{\alpha,r^*}(\cdot|(0,1),(w,q))\right)} d\Psi(p)) \Bigg\} \\
&\overset{(a)}{=} \min\left\{ \inf_{r\in[\underline{r}_\alpha(w,q),w)\cup[w,\overline{r}_\alpha(w,q)]} R(\Psi, F_\alpha(\cdot|r,(w,q))) \right\} \\
&\overset{(b)}{=} \inf_{F\in\mathcal{S}_{\alpha,w,q}} R(\Psi, F),
\end{aligned}$$

where (a) follows from the fact that the optimal oracle price associated with $F_\alpha(\cdot|r,(w,q))$ is r for all r in $[\underline{r}_\alpha(w,q), w) \cup [w, \overline{r}_\alpha(w,q)]$ (a fact established in Lemma 4.C-2); in (b), we use the definition of $\mathcal{S}_{\alpha,w,q}$ in Equation (4.3.4). This concludes the proof.

4.4 Optimal performance for deterministic mechanisms

We are now in a position to investigate the performance of general classes of mechanisms and their associated performance. In this section, we investigate the optimal performance when one restricts attention to mechanisms that post a deterministic price.

4.4.1 Optimal prices and performance

Using Theorem 4.1, it is possible to obtain the following reduction.

$$\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) \;=\; \mathcal{R}(\mathcal{P}_d, \mathcal{S}_{\alpha,w,q}) \;=\; \sup_{p\in[0,w]} \min_{r\in[\underline{r}_\alpha(w,q),w)\cup[w,\overline{r}_\alpha(w,q)]} \frac{Rev\left(p|F_\alpha(\cdot|r,(w,q))\right)}{\text{opt}(F_\alpha(\cdot|r,(w,q)))},$$

where for the last equality, we use the fact that one can restrict attention to mechanisms that post a price that is less or equal than the incumbent price w, as any deterministic mechanism that posts a price strictly above the incumbent price yields zero competitive ratio in the worst-case[2]. In turn, we may split the possible worst-cases into different regions to obtain

$$\begin{aligned}
&\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) \\
=\;& \sup_{p\in[0,w]} \min\left\{ \min_{r\in[\underline{r}_\alpha(w,q),w)} \frac{Rev\left(p|F_\alpha(\cdot|r,(w,q))\right)}{\text{opt}(F_\alpha(\cdot|r,(w,q)))}, \min_{r\in[w,\overline{r}_\alpha(w,q))} \frac{Rev\left(p|F_\alpha(\cdot|r,(w,q))\right)}{\text{opt}(F_\alpha(\cdot|r,(w,q)))} \right\} \\
\overset{(a)}{=}\;& \sup_{p\in[0,w]} \min\left\{ \min_{r\in[\underline{r}_\alpha(w,q),p)} \frac{p\overline{F}_\alpha(p|r,(w,q))}{r}, \min_{r\in[p,w)} \frac{p}{r}, \min_{r\in[w,\overline{r}_\alpha(w,q)]} \frac{p\overline{F}_\alpha(p|r,(w,q))}{\text{opt}(F_\alpha(\cdot|r,(w,q)))} \right\} \\
\overset{(b)}{=}\;& \sup_{p\in[0,w]} \min\left\{ \min_{r\in[\underline{r}_\alpha(w,q),p)} \frac{p\overline{F}_\alpha(p|r,(w,q))}{r}, \frac{p}{w}, \min_{r\in[w,\overline{r}_\alpha(w,q)]} \frac{p\overline{F}_\alpha(p|r,(w,q))}{\text{opt}(F_\alpha(\cdot|r,(w,q)))} \right\}, \quad (4.4.1)
\end{aligned}$$

where (a) follows from the fact that $\text{opt}(F_\alpha(\cdot|r,(w,q))) = r$ if $r < w$ (cf. Lemma 4.C-2) and noting that conditional on r belonging to $[p,w)$, the conversion rate is equal to 1 at p. (b) follows from noting that the worst-case in the latter case is for nature to select $r = w$. The reduction above highlights three "regimes" of worst cases that may emerge, driven by the location of the oracle optimal price.[3] For regular and mhr distributions, we establish that one can actually explicitly solve the problem above and characterize the spectrum of optimal transformations from data to decisions and the associated performance.

Theorem 4.2 (Maximin Ratio for deterministic mechanisms)**.** *Fix the set of mechanisms to be $\mathcal{P}_d$.*

[2]Any deterministic price above w would yield a performance of zero against a distribution that puts all the mass at w.

[3]Note that in cases when $\overline{r}_\alpha(w,q) < w$ (which happens when $\alpha \in (0,1]$ and $q < \Gamma_\alpha(1/\alpha)$), there are only two regimes as the last term in the brackets does not affect the worst-case.

- *For regular distributions ($\alpha = 0$), the optimal price is given by*

$$p_d^*(w,q) = w\left(\frac{2\sqrt{q}}{1+\sqrt{q}}\mathbf{1}\left\{q \in \left(0, \frac{1}{4}\right]\right\} + \frac{q(3-4q)}{1-q}\mathbf{1}\left\{q \in \left(\frac{1}{4}, \frac{1}{2}\right]\right\} + \mathbf{1}\left\{q \in \left(\frac{1}{2}, 1\right)\right\}\right),$$

and the maximin ratio is characterized as follows

$$\begin{aligned}\mathcal{R}(\mathcal{P}_d, \mathcal{F}_0(w,q)) &= \frac{2\sqrt{q}}{1+\sqrt{q}}\mathbf{1}\left\{q \in \left(0, \frac{1}{4}\right]\right\} + \frac{3-4q}{4(1-q)}\mathbf{1}\left\{q \in \left(\frac{1}{4}, \frac{1}{2}\right]\right\} \\ &\quad +(1-q)\mathbf{1}\left\{q \in \left(\frac{1}{2}, 1\right)\right\}.\end{aligned}$$

- *For mhr distributions ($\alpha = 1$), the optimal price is given by*

$$\begin{aligned}p_d^*(w,q) &= w\Bigg(\beta_q\left(\frac{e}{q}\right)\mathbf{1}\{q \in (0, \hat{q}]\} + \beta_q\left(\frac{1}{\log(q^{-1})}\right)\mathbf{1}\{q \in (\hat{q}, e^{-e^{-1}}]\} \\ &\quad +\mathbf{1}\{q \in (e^{-e^{-1}}, 1)\}\Bigg),\end{aligned}$$

and the maximin ratio is characterized as follows

$$\begin{aligned}\mathcal{R}(\mathcal{P}_d, \mathcal{F}_1(w,q)) &= \beta_q\left(\frac{e}{q}\right)\mathbf{1}\{q \in (0, \hat{q}]\} + \rho(q)\mathbf{1}\{q \in (\hat{q}, e^{-e^{-1}}]\} \\ &\quad +eq\log(q^{-1})\mathbf{1}\{q \in (e^{-e^{-1}}, 1)\},\end{aligned}$$

where if W is the Lambert function defined as the inverse of $x \to xe^x$ in $[0, +\infty)$, $\beta_q(x) = 1 - \frac{1}{\log(q^{-1})}(W(x) + \frac{1}{W(x)} - 2)$, $\rho(q) = \beta_q\left(\frac{1}{\log(q^{-1})}\right) e\log(q^{-1}) e^{-\log(q^{-1})\beta_q\left(\frac{1}{\log(q^{-1})}\right)}$ and $\hat{q}$ is the unique solution in $[0,1]$ to the equation $W\left(\frac{1}{\log(q^{-1})}\right) W\left(\frac{e}{q}\right) = 1$. Numerically $\hat{q} \in [0.52, 0.53]$.

The proof is presented in Appendix 4.D. When using deterministic mechanisms, this result enables one, quite notably, to obtain in closed form the exact value associated with the conversion rate at one price when using deterministic mechanisms, but also the optimal price to post. The structure of the result is also quite instructive. There are three “regimes": high, intermediate and

low conversion rates (where those regions depend of the focal class). For high conversion rates, the seller's optimal price is simply to continue to post the incumbent price. Intuitively, the seller may want to explore higher prices, but the seller runs the risk of losing all customers when pricing higher. In this case, the hard cases for the seller are masses at or around w. For intermediate conversion rates and low conversion rates, the situation is different as in those cases, a more subtle interplay arises, and the seller needs to carefully select a price below the incumbent price to optimize its competitive ratio. We next analyze some implications of this result.

4.4.2 Performance analysis

While Theorem 4.2 provides a full characterization, there are various notable observations with regard to the implications of the result.

In Figure 4.4, we plot the optimal price to post given the data at hand. As highlighted in

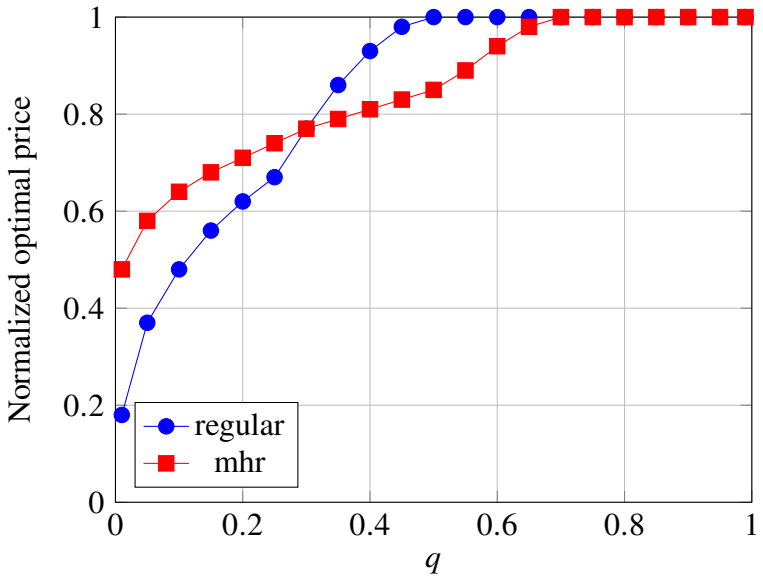


Figure 4.4: Optimal deterministic normalized price $p_d^*(w,q)/w$ as a function of the probability of sale q.

Theorem 4.2, there are various regimes. When q is "high", then the optimal price for the seller is simply to post the incumbent price w. However, when the observed probability of sale decreases, the seller will price below the incumbent price w, potentially much below the latter.

In Figure 4.5, we plot the maximin ratio against regular and mhr distributions as a function of the conversion rate rate observed for the incumbent price. This value can be interpreted as measuring the value of information associated with the data when using deterministic mechanisms.

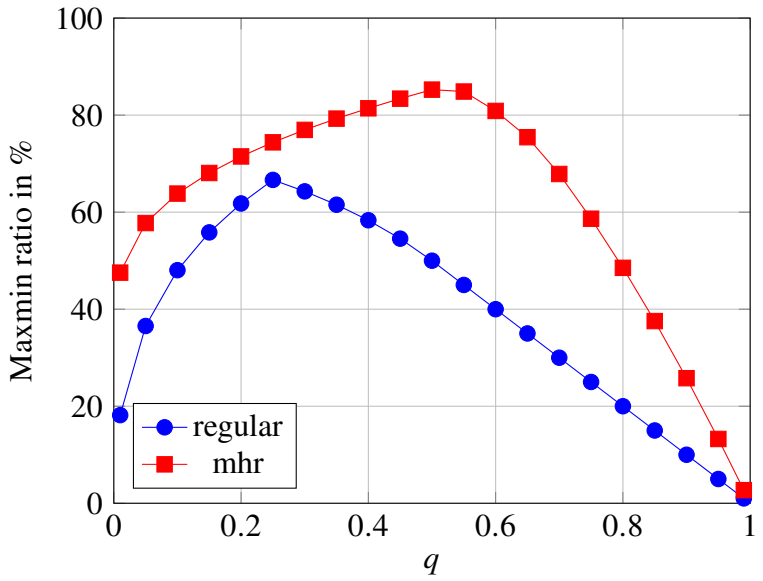


Figure 4.5: Performance of deterministic mechanisms as a function of the probability of sale q.

A first striking implication is associated with the levels of performance that one can achieve with a simple deterministic mechanism, despite the very limited information available at hand. For example, when simply knowing that the conversion rate of customers is 50% at a particular price, and that the value distribution is mhr, there exists a deterministic pricing mechanism that can guarantee more than 85% of oracle performance! For regular distribution, with a 25% conversion rate, the seller can guarantee more than 66% of oracle performance using an adequate deterministic mechanism. Our results provide a full mapping from historical conversion rate to achievable performance and prescription.

As shown in Figure 4.5, the maximin ratio can be quite different depending on the underlying class of distributions. The difference between the two curves highlights the “price" of heavier tails that one could face under regular distributions.

While the optimal maximin ratio converges to zero as $q \to 0$ (i.e simply knowing that no customer purchases beyond a given price does not guide pricing decisions), quite strikingly, the rate of convergence is quite *slow*. Indeed, this is illustrated in Figure 4.5. For example, with a conversion rate of 1%, it is still possible to guarantee more than 47% against mhr distributions and more than 18% against regular distributions with appropriate prices. As a matter fact, the closed form formulas in Theorem 4.2 show that the maximin ratio, while converging to zero as q becomes small, it does so only at rate $\Theta(1/\log(q^{-1}))$ for mhr and rate $\Theta(\sqrt{q})$ for regular distributions. In other words, this highlights that even knowing only that a "small" fraction of customers purchases at a given price is very informative. We will further see that significant more value can be captured through randomized mechanisms.

The limiting behavior at 0 has also other implications with regard to the earlier literature. The fact that the performance converges to zero as q tends to zero shows that the results in [44], that a simple pricing rule can guarantee a fraction of revenues with knowledge of the exact value of the upper bound of the support for particular parametric families (corresponding to $q = 0^+$ in the current chapter), do not extend to non-parametric classes such as mhr or regular. At the same time, as highlighted above, against such distributions, the convergence of performance to zero is very slow.

4.5 Optimal performance for randomized mechanisms

In this section, we now turn to the analysis of general randomized mechanisms. These will allow to measure the full value of information associated with percentile data.

4.5.1 Near optimal mechanisms and performance

Reduction to bounded discrete mechanisms. As mentioned following the statement of Problem $\mathcal{MP}$, one of the challenges in analyzing randomized mechanisms stems from the infinite dimensional nature of the space of the seller's strategies. Next, we establish that bounded discrete mechanisms can approximate arbitrary closely general randomized mechanisms.

More specifically, consider an increasing sequence of N positive reals $\mathbb{A} = \{a_i\}_{i=1}^N$ for N in $\mathbb{N}^*$ and define the set of discrete mechanisms on $\mathbb{A}$ as

$$\mathcal{P}_{\mathbb{A}} = \left\{ \Psi \in \mathcal{P} \ : \ \Psi(x) = \sum_{j=1}^{N} p_j \mathbf{1}\{x \geq a_i\}, \text{ for some } 0 \leq p_i \leq 1, \ \sum_{j=1}^{N} p_i = 1 \right\}.$$

Proposition 4.2. *Fix Ψ in $\mathcal{P}$, q in $(0,1)$, $N > 1$, and any finite sequence of increasing reals $\mathbb{A} = \{a_i\}_{i=1}^N$ such that $0 < a_1 \leq w \leq a_N$. Then there exists $\Psi_{\mathbb{A}}$ in $\mathcal{P}_{\mathbb{A}}$ such that*

$$\inf_{F \in \mathcal{F}_\alpha(w,q)} R(\Psi_{\mathbb{A}}, F) \geq \inf_{F \in \mathcal{F}_\alpha(w,q)} R(\Psi, F) - \frac{\Delta(\mathbb{A})}{a_1} - \frac{1}{q(1 + (q^{-1} - 1)a_N)} \mathbf{1}\{a_N < \overline{r}_\alpha(w,q)\},$$

where $\Delta(\mathbb{A}) = \sup_i \{a_i - a_{i-1}\}$.

The proof of Proposition 4.2 is presented in Section 4.E. This result implies two main points. First, it is possible to approximate arbitrarily closely the performance of general randomized mechanisms through *discrete and bounded* mechanisms. Second, the optimality gap between the two classes can be quantified and is driven by two terms: a discretization term $(\Delta(\mathbb{A})/a_1)$associated with how fine the grid is, and a truncation term. For mhr distributions, since $\overline{r}_\alpha(w,q)$ is finite, the truncation term can be eliminated by selecting a_N high enough.

The key ideas underlying the result revolve around, first, quantifying how much the seller looses by restricting the support of the mechanism to a bounded interval $[0, b]$ with $b > w$. We quantify this error by leveraging the concavity of the revenue function in the quantity space as well as the upper-bound on the tail of the distribution obtained from the regularity assumption. The second step consists of quantifying how much the seller looses by restricting to the class of mechanisms that randomize over a finite set of prices using the single crossing property of regular distributions applied to local intervals (Lemma 4.1). It is important to note that here, because there is no exogenously imposed uniform positive lower bound on $\text{opt}(F)$, to appropriately control losses, it is key to perform an analysis that maintains the coupling between the achieved revenues and the oracle revenues.

A family of factor revealing finite dimensional linear programs. We are now ready to present the sequence of finite dimensional linear programs that will be central to our analysis. For any q in $(0,1)$, and α in $[0,1]$, we will define a linear program parametrized by a finite sequence of increasing positive reals $\mathbb{A} = \{a_i\}_{i=0}^{2N}$, where $N > 1$, such that $a_N < w, a_{N+1} = w$. In particular, we define the following linear program.

$$\begin{aligned} \underline{\mathcal{L}}_{\alpha,q,\mathbb{A}} \;=\; & \max_{\mathbf{p},c} \; c \qquad\qquad (\mathcal{LP}) \\ & s.t. \;\; \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q)))} \sum_{j=0}^{2N} a_j \overline{F}_\alpha(a_j|a_i,(w,q)) p_j \geq c \quad i = 0,\ldots 2N, \\ & \sum_{j=0}^{2N} p_j \leq 1, \quad p_i \geq 0 \quad i = 0,\ldots 2N, \end{aligned}$$

with $\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q))) = \lim_{x\to a_{i+1}^-} \text{opt}(F_\alpha(\cdot|x,(w,q)))$ for any $i = 0,\cdots,2N$.

Theorem 4.3 (Maximin Ratio for Randomized Mechanisms)**.** *Fix q in $(0,1)$, and α in $[0,1]$.*

(i) For any sequence of increasing positive reals $\mathbb{A} = \{a_i\}_{i=0}^{2N}$, where $N > 1$, such that $a_{N+1} = w$, the solution to Problem $(\mathcal{LP})$ provides a feasible distribution of prices and its performance is lower bounded by $\underline{\mathcal{L}}_{\alpha,q,\mathbb{A}}$, implying that

$$\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,q)) \geq \underline{\mathcal{L}}_{\alpha,q,\mathbb{A}}.$$

(ii) Furthermore, there exists a sequence of increasing prices in $= \{a_i\}_{i=0}^{N}$ such that:

$$\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,q)) \leq \underline{\mathcal{L}}_{\alpha,q,\mathbb{A}} + O\left(\frac{1}{\sqrt{N}}\right),$$

where the O notation includes constants that depend only on α and q.

The proof of Theorem 4.3 is presented in Appendix 4.E. Theorem 4.3 is notable in two respects. First, it provides a systematic procedure to obtain a lower bound on performance through a linear program but also an associated pricing distribution that guarantees such performance. The second

notable point is that, by judiciously constructing a discrete grid, the values associated with a sequence of linear programs constructed converge to the maximin ratio of interest as the grid becomes finer. The proof is constructive and provides such a sequence. In addition, the result implies that it suffices to solve a linear program with order N variables and order N constraints to yield an approximation within order $1/\sqrt{N}$ of the maximin ratio and an associated near-optimal prescription.

Remark (alternative feasible price sets and constraints). We note that above, we assumed that the set of feasible prices to post was any non-negative number. In practice, often, there are are constraints on the set of feasible prices to use. Such constraints can be encoded in the framework. Indeed, one can still apply Theorem 4.1 as it applies to any subclass of mechanisms $\mathcal{P}' \subseteq \mathcal{P}$. For example, if the set of feasible prices is a discrete set $\{\phi_1, \phi_2, \cdots, \phi_K\}$, the result applies when the subset $\mathcal{P}'$ is the set of mechanisms that can only put mass over a subset of these prices. With this result in hand, one may then develop a special case of the Linear Program $\mathcal{LP}$ in which one replaces the sequence $\mathbb{A} = \{a_i\}_{i=0}^{2N}$ by the sequence $\{\phi_1, \phi_2, \cdots, \phi_K\}$ (and partition the latter between values below w and above w to partition the constraints). In turn, the result would follow from Theorem 4.3. The main difference is that if we start with a discrete set of prices, one does not need to call on Proposition 4.2 as the sequence on which one randomizes is pre-determined, and the discretization error would only stem from the discretization of the set of constraints. In general, the framework is flexible and could allow other constraints such as, e.g., to never put too much weight on prices above w (as these "riskier"). Such a constraint could be easily added to Problem $\mathcal{LP}$.

4.5.2 Performance analysis

We next discuss the implications of Theorem 4.3 in terms of performance. In Figure 4.6, we plot the maximin ratio for randomized mechanisms superimposed with that for deterministic mechanisms. For randomized mechanisms, we note that we plot a lower bound that is obtained by selecting the sequence used in Theorem 4.3 (we provide further details in Section 4.G). Furthermore, all the lower bounds depicted can be shown to be within 1% of the maximin ratio by solving an

alternative, but related, linear program that can be shown to yield an upper bound (such an LP is presented in Appendix 4.G).

The figure highlights various points. First the value of randomization is limited for a historical price with a “moderate" probability of sale and is more critical against distributions with heavier tails (regular) versus mhr. At the same time, the value of randomization can be quite significant for low and high conversion rates. For example, with access to a price with a probability of sale of 1% against regular distributions, the performance improves from 18% for deterministic mechanisms to 31% for randomized ones. For a probability of sale of 75%, the performance improves from 25% to 41%. Against mhr distributions, for the previous probabilities of sale of 1% and 75% the performance improves from 47% to 51% and from 58% to 64%, respectively.

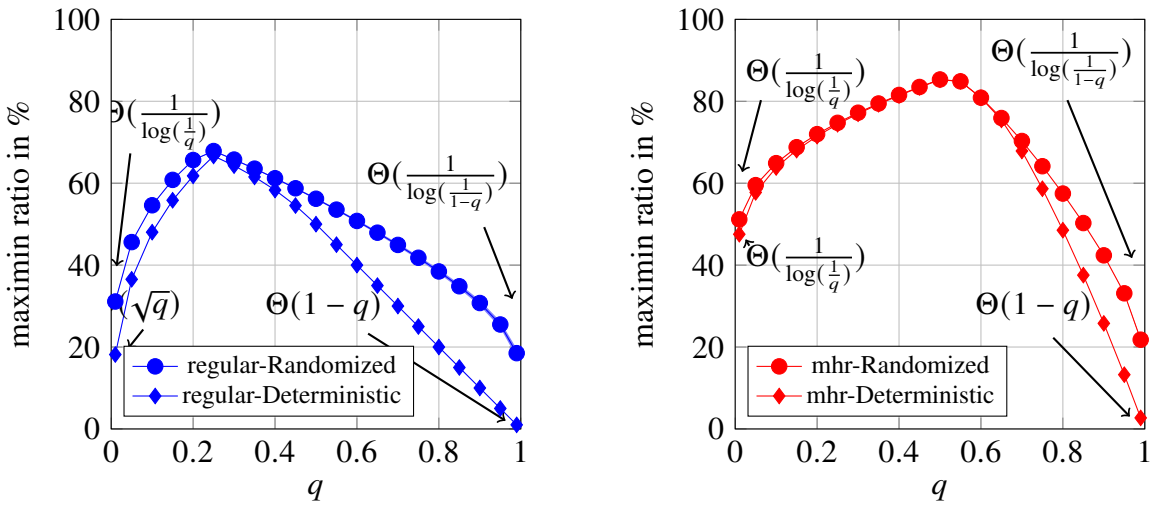
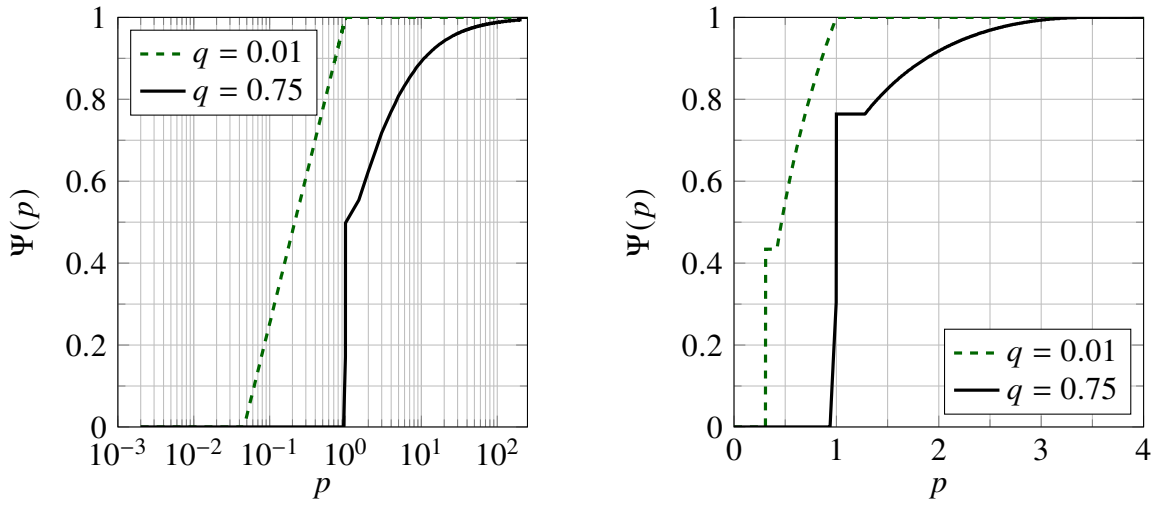


Figure 4.6: Maximin ratio as a function of the probability of sale: The figure depicts the performance of optimal randomized and deterministic mechanisms, as well as the rates of convergence to zero when q approaches 0 or 1. The left panel corresponds to regular distributions and the right one to mhr distributions.

In Figure 4.7, we illustrate the structure of the cdf associated with near-optimal mechanisms obtained by solving Problem ($\mathcal{LP}$) for two values of the probability of sale, one in which a very small fraction of customers purchases ($q = 0.01$) and another in which a large fraction of customers purchase ($q = 0.75$). Some examples for other values of q are presented in Appendix 4.H. For these, without loss of generality, we fix $w = 1$. Recall from Figure 4.4 that for $q = 0.01$, against



regular distributions **mhr distributions**

Figure 4.7: Illustration of near optimal mechanisms: The figure depicts near optimal pricing distributions for $w = 1$, $q = 0.01$ and $q = 0.75$. The left panel corresponds to regular distributions (plotted using a log scale) and the right panel to mhr distributions (on a regular scale).

regular distributions, the optimal deterministic price was about 0.18. Indeed, intuitively, with so few customers purchasing at the incumbent price, the seller should consider decreasing her price. When randomization is allowed, the (near) optimal mechanism puts mass over values between 0.05 and 1. This careful randomization yields an improvement in performance from 18% to more than 31%. When $q = 0.75$, the optimal deterministic price was simply given by $w = 1$. Indeed, with so many customers already purchasing, it seems natural that the seller would not want to consider a decrease in price. For deterministic mechanisms, she cannot increase the price as nature could counter such a price to yield zero performance (with as mass at w). A (near) optimal randomized mechanism puts significant mass right around 1 (about 50% of the mass), but also inflates the current price and puts the remaining mass between 1 and ∞. Here, the benefits of randomization are substantial (from a ratio of 25% for deterministic prices to about 41.35% for randomized prices). Against mhr distributions, the structure of (near) optimal mechanisms is equally rich, with mass spread below 1, right around 1 or above 1 depending on the cases.

On the values of small and large probabilities of sale. We next explore in more detail the value of randomization for low and high values of q. Our next result provides theoretical lower and upper bounds on the optimal performance $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q))$ as $q \to 0$.

Proposition 4.3. *For any q in $(0, 0.4)$, and α in $[0, 1]$ there exist $c_1, c_2 > 0$ such that*

$$\frac{c_1}{\log(q^{-1})} \leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q)) \leq \frac{c_2}{\log(q^{-1})}.$$

The proof of Proposition 4.3 is presented in Appendix 4.E. While the optimal maximin ratio converges to zero as $q \to 0$, quite strikingly, the rate of convergence is *extremely slow*, $\Theta(\log(q^{-1}))$ for both mhr and regular distributions. In other words, this shows that even very low conversion rates are quite informative for pricing purposes. Furthermore, recalling the result for deterministic mechanisms, we see that, for regular distributions, randomization allows to fundamentally alter the rate of convergence as q approaches zero, from $\Theta(\sqrt{q})$ to $\Theta(\log(q^{-1}))$, altering the value that can be extracted from the data. Randomization is extremely valuable with very low conversion rates. Such an effect is less pronounced for mhr distributions as the rate of convergence to zero was already extremely slow for deterministic mechanisms.

We now explore the value of randomization for high values of q. Our next result provides theoretical lower and upper bounds on the optimal performance $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q))$ as $q \to 1$.

Proposition 4.4. *For any q in $(0.5, 1)$, and α in $[0, 1]$, there exist $c_3, c_4 > 0$ such that*

$$\frac{c_3}{\log((1-q)^{-1})} \leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, q)) \leq \frac{c_4}{\log((1-q)^{-1})}.$$

The proof of Proposition 4.4 is presented in Section 4.E. This result further highlights the significant value of randomization against both regular and mhr distributions. Indeed, with deterministic mechanisms, the performance decreased linearly with q as q approached one. Now, the performance only decreases at the significantly slower rate of $\Theta(1/\log((1-q)^{-1}))$. The value of randomization is again extremely high, as illustrated in Figure 4.6.

4.6 Optimal pricing with uncertainty on the probability of sale

In this section, we show how the ideas established in the previous sections can be generalized when the seller does not know the exact value of the probability of sale but only an interval to which it belongs $[q_l, q_h]$. In particular, we focus on general randomized mechanisms and the object of interest is now

$$\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, [q_l, q_h])) \;:=\; \sup_{\Psi \in \mathcal{P}} \inf_{F \in \mathcal{F}_\alpha(w, [q_l, q_h])} R(\Psi, F).$$

For any $q_l < q_h$ in $(0,1)^2$, α in $[0,1]$, $N > 1$ and any finite sequence of increasing prices $\mathbb{A} = \{a_i\}_{i=1}^{2N}$, such that $a_N < w, a_{N+1} = w$, define a generalized version of $\mathcal{LP}$ given by

$$\begin{aligned}
\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;=\; \max_{\mathbf{p},c}\; & c && (\mathcal{LP}\text{-int})\\
s.t.\; & \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))}\sum_{j=1}^{2N} a_j \overline{F}_\alpha(a_j|a_i,(w,q_l))p_j \geq c \quad i = 1,\dots N,\\
& \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))}\sum_{j=1}^{2N} a_j \overline{F}_\alpha(a_j|a_i,(w,q_h))p_j \geq c \quad i = N+1,\dots 2N,\\
& \sum_{j=1}^{2N} p_j \leq 1, \quad p_i \geq 0 \quad i = 1,\dots 2N,
\end{aligned}$$

with $\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l))) = \lim_{x \to a_{i+1}^-} \mathrm{opt}(F_\alpha(\cdot|x,(w,q_l)))$ for any $i = 0, \cdots, N$.

Theorem 4.4. *Fix $q_l < q_h$ in $(0,1)^2$, and α in $[0,1]$.*

1. *For any sequence of increasing positive reals $\mathbb{A} = \{a_i\}_{i=1}^{2N}$, where $N > 1$, such that $a_{N+1} = w$, the solution to Problem ($\mathcal{LP}$-int) provides a feasible distribution of prices and its performance is lower bounded by $\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}$, yielding that*

$$\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, [q_l, q_h])) \geq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}.$$

2. *Furthermore, there exists a sequence of increasing prices* $\mathbb{A} = \{a_i\}_{i=0}^{2N+1}$ *with* $a_0 = \underline{r}_\alpha(w, q_l)$, $a_{N+1} = w$, *such that:*

$$\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, [q_l, q_h])) \leq \underline{\mathcal{L}}_{\alpha, q_l, q_h, \mathbb{A}} + O\left(\frac{1}{\sqrt{N}}\right),$$

where the O *notation includes constants that depend only on* α *and* q_l, q_h.

The full Proof of Theorem 4.4 is presented in Appendix 4.E. The result highlights that uncertainty in the probability of sale can be incorporated and optimal performance can again be approximated with arbitrary accuracy by solving a finite dimensional linear program.

Sensitivity Analysis. We next illustrate the impact of uncertainty on the maximin performance. While the framework above applies to any interval, to anchor ideas we focus on the following experiment to parametrize the interval with a more "physical" quantity. Consider a setting in which the seller has access to N buy/no-buy decisions of customers at the fixed price w. The seller can then use this data to estimate the conversion rate $\overline{F}(p)$ through the following estimator

$$\hat{q} = \frac{\text{\#buy decisions}}{N},$$

We will fix an uncertainty interval through

$$[q_l, q_h] = \left[\hat{q} - 1.96\frac{\sqrt{\hat{q}(1-\hat{q})}}{\sqrt{N}}, \hat{q} + 1.96\frac{\sqrt{\hat{q}(1-\hat{q})}}{\sqrt{N}}\right],$$

inspired by a 95% confidence interval. In Figure 4.8, we present the maximin ratio as a function of the mid-point $\hat{q}$ and N for various values of N in $[100, 500, 1000, \infty]$, for both mhr and regular distributions.

The figure quantifies the impact of uncertainty in the probability of sale on performance degradation. The case $N = \infty$ corresponds the case of known probability of sale. We observe the seller can still leverage the noisy information to achieve high levels of performance despite the uncertainty in the conversion rate.

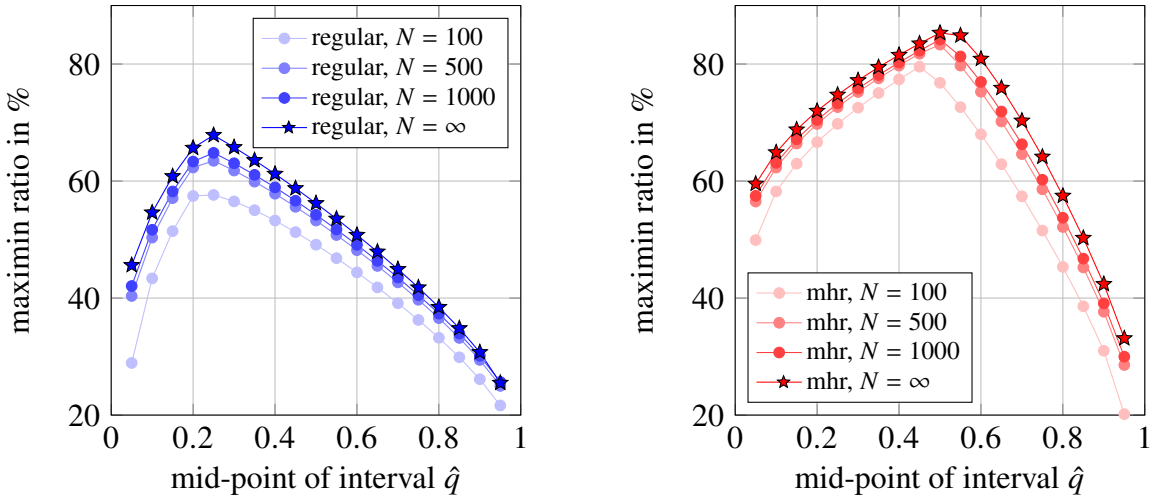


Figure 4.8: Maximin ratio as a function of the uncertainty: The figure depicts the performance of optimal randomized mechanisms in face of uncertainty in the probability of sale. The left panel corresponds to regular distributions and the right one to mhr distributions.

4.7 Conclusion

In this present chapter, we presented a systematic analysis of pricing in the presence of a common data structure historical data at a single price. We propose a novel and general framework that allows to obtain how such data should be optimally used and the best performance one can achieve. The novel framework is powerful but also yields novel insights on the value of such information, the power of simple deterministic mechanisms, but also the incremental value of randomized mechanisms and the regimes in which it is most significant.

There are many avenues of future research that this work opens up. This framework offers a framework to quantify in a robust manner the value of a single measurement. As such, it offers a foundation for future work that can tackle how to leverage measurements at multiple prices. There, a key question would be how to find a parallel reduction to Theorem 4.1. More broadly, a promising direction is also to leverage such analyses to inform the the design of static and dynamic price experiments.

4.A Preliminaries and properties of Generalized Pareto Distributions

Throughout the rest of the chapter, whenever a distribution F is defined, and when clear from context, we use q_w to denote $\overline{F}(w)$ to lighten the notation. We also use the generalized inverse of a distribution F in $\mathcal{D}$, defined by $F^{-1}(1-q) := \inf\{v \text{ in } \mathbb{R}^+ \text{ s.t. } F(v) \geq 1-q\}$ for all q in $[0,1]$.

For a distribution with positive density function f on its support $[a,b]$, where $0 \leq a < \infty$ and $a \leq b \leq \infty$, we denote the α-virtual value function for $v \in [a,b]$ by

$$\phi_F^\alpha(v) := (1-\alpha)v - \frac{\overline{F}(v)}{f(v)}.$$

Lemma 4.A-1. *Fix two scalars $\beta > 0$ and $s \geq 0$. The cumulative distribution function $\Gamma_\alpha(\beta(v-s))$ for $v \geq s$ admits a constant α-virtual value function given by*

$$(1-\alpha)s - \frac{1}{\beta}.$$

Proof of Lemma 4.A-1. Let us first explicitly compute the α-virtual value function. The derivative of $\Gamma_\alpha(\beta(v-s))$, for any $v \geq s$, is given by

$$-\beta\left(\Gamma_\alpha(\beta(v-s))\right)^{2-\alpha}.$$

Therefore, the α-virtual value function evaluated at $v \geq s$ is given by

$$\begin{aligned}
(1-\alpha)v - \frac{\Gamma_\alpha(\beta(v-s))}{\beta\left(\Gamma_\alpha(\beta(v-s))\right)^{2-\alpha}} &= (1-\alpha)v - \frac{1}{\beta}\left(\Gamma_\alpha(\beta(v-s))\right)^{\alpha-1} \\
&= (1-\alpha)v - \frac{1}{\beta}\left(1 + (1-\alpha)\Gamma_\alpha^{-1}(\beta(v-s))\right) \\
&= (1-\alpha)v - \frac{1}{\beta} - (1-\alpha)(v-s) \\
&= (1-\alpha)s - \frac{1}{\beta}.
\end{aligned}$$

This completes the proof. □

We will also need the following result derived in [51, Lemma E-1].

Lemma 4.A-2. *Fix $\alpha \in [0,1]$, two scalars $\beta \geq 0$ and $w' \geq 0$. The revenue function $v\Gamma_\alpha(\beta(v-w'))$ for $v \geq w' - \frac{1}{(1-\alpha)\beta}$ is unimodal and attains its maximum at*

$$r = \max\left\{\frac{1-(1-\alpha)\beta w'}{\beta\alpha}, w' - \frac{1}{(1-\alpha)\beta}\right\}.$$

With the following conventions: $\max\{+\infty, v\} = +\infty$*, and* $\max\{-\infty, v\} = v$ *for any real number* v*.*

4.B Proofs and auxiliary results for Section 4.2

Proof of Proposition 4.1. If q in $\{0,1\}$, then the result follows from Lemma 4.B-3. For q in $(0,1)$, let Ψ a mechanism in $\mathcal{P}$. We know that

$$\lim_{u\to\infty} \bar{\Psi}(u) := 1 - \Psi(u) = 0.$$

Fix $\epsilon > 0$. By definition of the limit, there exists $M \geq w$ such that for any $u \geq M$, we have:

$$\bar{\Psi}(u) \leq \frac{\epsilon}{2}. \tag{4.B-1}$$

For any integer N, consider the following distribution $F_{\Psi,N}$ defined through its Complementary Cumulative Distribution Function $\overline{F}_{\Psi,N}$:

$$\overline{F}_{\Psi,N}(v) \quad = \quad \begin{cases} 1 & \text{if } v < 0, \\ q & \text{if } v \text{ in } [0,w), \\ \frac{q}{N} & \text{if } v \text{ in } [w, N^2M), \\ 0 & \text{if } v \text{ in } [N^2M, +\infty). \end{cases}$$

Note that $F_{\Psi,N}$ in $\mathcal{G}(w,I)$ and that $F_{\Psi,N}$ represents a three point mass distribution with mass at points 0, w and N^2M.

Note also that $\text{opt}(F_{\Psi,N}) = \max\{qw, qNM\}$. Since $M \geq w$ and $N \geq 1$, we have $\text{opt}(F_{\Psi,N}) = qNM$. Thus the performance of mechanism Ψ is given by

$$\begin{aligned}
R(\Psi, F_{\Psi,N}) &= \frac{1}{qNM}\left(\int_{[0,w)} u\overline{F}_{\Psi,N}(u)d\Psi(u)\right. \\
&\quad \left. + \int_{[w,N^2M)} u\overline{F}_{\Psi,N}(u)d\Psi(u) + \int_{[N^2M,+\infty)} u\overline{F}_{\Psi,N}(u)d\Psi(u)\right) \\
&= \frac{1}{NM}\left(\int_{[0,w)} ud\Psi(u) + \frac{1}{N}\int_{[w,N^2M)} ud\Psi(u)\right) \\
&= \frac{1}{NM}\left(\int_{[0,w)} ud\Psi(u) + \frac{1}{N}\int_{[w,M)} ud\Psi(u) + \frac{1}{N}\int_{[M,N^2M)} ud\Psi(u)\right) \\
&\leq \frac{1}{NM}\left(w\Psi(w) + \frac{1}{N}M(\Psi(M) - \Psi(0)) + \frac{1}{N}N^2M(\Psi(N^2M) - \Psi(M))\right) \\
&\leq \frac{w}{NM} + \frac{1}{N^2} + (1 - \Psi(M)),
\end{aligned}$$

where in the last step we use the fact that for any $u \geq 0$ $\Psi(u)$ is in $[0, 1]$.

Let us now choose N large enough such that

$$\frac{w}{NM} + \frac{1}{N^2} \leq \frac{\epsilon}{2},$$

Combining the latter with (4.B-1), we get

$$R(\Psi, F_{\Psi,N}) \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

Taking $\epsilon \to 0$ concludes the proof. □

Additional result for the cases q in $\{0, 1\}$

Lemma 4.B-3. *For any mechanism Ψ in $\mathcal{P}$, and any α in $[0, 1]$, if q in $\{0, 1\}$, then*

$$\inf_{F \in \mathcal{F}_\alpha(w,\{q\})} R(\Psi, F) = 0.$$

Proof of Lemma 4.B-3. We will first show the case when $q = 0$ then the case when $q = 1$. For both cases, we will exhibit worst case families of distributions for which the seller cannot achieve a non-trivial guarantee.

For any $r > 0$, let us introduce the following distribution through its Complementary Cumulative Distribution Function:

$$\overline{F}_r(v) \quad = \quad \begin{cases} 1 & \text{if } v \text{ in } [0, r), \\ 0 & \text{if } v \text{ in } [r, +\infty). \end{cases}$$

The latter distribution represents a point mass at r.

Case $q = 0$. We have, for any $r < w$, F_r in $\mathcal{F}_\alpha(w, \{0\})$ and $\text{opt}(F_r) = r$. Furthermore, for any mechanism Ψ in $\mathcal{P}$, we have:

$$\begin{aligned} \inf_{F \in \mathcal{F}_\alpha(w,\{0\})} R(\Psi, F) &\leq \frac{\mathbb{E}_\Psi[p\overline{F}_r(p)]}{\text{opt}(F_r)} = \frac{1}{r}\int_0^r u d\Psi(u) = \frac{1}{r}\int_0^r \int_0^u ds d\Psi(u) \\ &\overset{(a)}{=} \frac{1}{r}\int_0^r \int_s^r d\Psi(u) ds = \frac{1}{r}\int_0^r (\Psi(r) - \Psi(s))\, ds \\ &\leq \Psi(r) - \Psi(0). \end{aligned}$$

Where in equality (a), we used Fubini–Tonelli theorem as $(s, u) \to 1$ is a non-negative measurable function and $([0, r], d\Psi)$ and $([0, r], dx)$ are σ-finite measure spaces.

The right hand side above converges to zero as $r \to 0^+$ since Ψ in $\mathcal{P} = \mathcal{D}$ and is therefore right continuous. We conclude the case $q = 0$.

Case $q = 1$. We have, for any $r > w$, F_r in $\mathcal{F}_\alpha(w, \{1\})$ and $\text{opt}(F_r) = r$. Furthermore, for any mechanism Ψ, we have:

$$\inf_{F \in \mathcal{F}_\alpha(w,\{1\})} R(\Psi, F) \quad \leq \quad \frac{\mathbb{E}_\Psi[p\overline{F}_r(p)]}{\text{opt}(F_r)} = \frac{1}{r}\int_0^r u d\Psi(u) = \frac{1}{r}\int_0^r (\Psi(r) - \Psi(u)) du.$$

Since Ψ in $\mathcal{P}$, we have $\lim_{r\to+\infty}\Psi(r)=1$, therefore, for any $\epsilon>0$, there exists $A>0$ such that:

$$1-\Psi(r)=|\Psi(r)-1|<\epsilon \quad \text{if } r\in[A,+\infty). \tag{4.B-2}$$

Let r in $(A,+\infty)$, we have:

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,\{1\})} R(\Psi,F) &\leq \frac{1}{r}\int_0^r(\Psi(r)-\Psi(u))du \\
&= \frac{1}{r}\left(\int_{[0,A)}(\Psi(r)-\Psi(u))\,du+\int_{[A,r]}(\Psi(r)-\Psi(u))\,du\right) \\
&\overset{(a)}{\leq} \frac{1}{r}\left(_{[0,A)}du+\int_{[A,r]}(1-\Psi(u))\,du\right) \\
&= \frac{A}{r}+\frac{1}{r}\int_A^r(1-\Psi(u))\,du \\
&\overset{(b)}{\leq} \frac{A}{r}+\frac{r-A}{r}\epsilon \overset{(c)}{\leq} \frac{A}{r}+\epsilon,
\end{aligned}$$

where in (a) we use the fact that for any $u\geq 0$, we have $0\leq\Psi(u)\leq 1$. And in (b) we use (4.B-2). In (c), we used the fact that $r-A\leq r$.

Hence we conclude that for any $r\geq A/\epsilon$, we get that

$$\inf_{F\in\mathcal{F}_\alpha(w,\{1\})} R(\Psi,F)\leq 2\epsilon.$$

Since ϵ was arbitrary, this completes the proof for the case $q=1$. □

4.C Proofs and auxiliary results for Section 4.3

Lemma 4.C-1. *For any $(s,q_s),(s',q_{s'})$ in $([0,+\infty)\times[0,1])^2$ such that $s\leq s'$ and $q_s\geq q_{s'}>0$, the distribution $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ with $t\geq s'$, defined in* (4.3.1)*, belongs to $\mathcal{F}_\alpha(s,q_s)\cap\mathcal{F}_\alpha(s',q_{s'})$ if and only if*

$$q_s\geq\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_{s'})\,\frac{s}{s'}\right).$$

Proof of Lemma 4.C-1. Let us show each direction.

$\Longrightarrow$) If the distribution $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ belongs to $\mathcal{F}_\alpha(s,q_s)\cap\mathcal{F}_\alpha(s',q_{s'})$ then $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ is $\alpha-$regular, therefore by Lemma 4.1 applied to the interval $[0,s']$, we have that

$$q_s=\overline{G}_{\alpha,t}(s|(s,q_s),(s',q_{s'}))\geq\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(q_{s'}\right)\frac{s}{s'}\right),$$

and hence the first direction is established. Let us now show the other direction.

$\Longleftarrow$) Suppose now that $q_s\geq\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(q_{s'}\right)\frac{s}{s'}\right)$. By definition of $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$, we have:

$$\begin{aligned}\overline{G}_{\alpha,t}(s|(s,q_s),(s',q_{s'}))&=q_s\\ \overline{G}_{\alpha,t}(s'|(s,q_s),(s',q_{s'}))&=q_{s'}.\end{aligned}$$

Therefore, we only have to show that the distribution $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ is α-regular. Using Lemma 4.A-1, the associated α-virtual value function is given by

$$\phi^\alpha_{G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))}(v)=\begin{cases}-\frac{s}{\Gamma_\alpha^{-1}(q_s)} & \text{if } v \text{ in } [0,s],\\ (1-\alpha)s-\frac{s'-s}{\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)} & \text{if } v \text{ in } (s,s'].\end{cases}$$

Thus the α-virtual value function is piece-wise constant. Now we need to show that $\phi^\alpha_{G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))}(v)$ is non-decreasing. Next, we evaluate the difference between the two constant

values that the virtual value function is taking.

$$
\begin{aligned}
& (1-\alpha)s - \frac{s'-s}{\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)} - (-\frac{s}{\Gamma_\alpha^{-1}(q_s)}) \\
= \quad & s\frac{(1-\alpha)\Gamma_\alpha^{-1}(q_s)\,\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right) + \Gamma_\alpha^{-1}(q_s) + \Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right) - \frac{s'}{s}\Gamma_\alpha^{-1}(q_s)}{\Gamma_\alpha^{-1}(q_s)\,\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)} \\
\overset{(a)}{=} \quad & s\frac{\Gamma_\alpha^{-1}\left(q_s\frac{q_{s'}}{q_s}\right) - \frac{s'}{s}\Gamma_\alpha^{-1}(q_s)}{\Gamma_\alpha^{-1}(q_s)\,\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)} \\
= \quad & s\frac{\Gamma_\alpha^{-1}(q_{s'}) - \frac{s'}{s}\Gamma_\alpha^{-1}(q_s)}{\Gamma_\alpha^{-1}(q_s)\,\Gamma_\alpha^{-1}\left(\frac{q_{s'}}{q_s}\right)} \overset{(b)}{\geq} 0,
\end{aligned}
$$

where (a) stems from the fact that $\Gamma_\alpha^{-1}(uv) = \Gamma_\alpha^{-1}(u) + \Gamma_\alpha^{-1}(v) + (1-\alpha)\Gamma_\alpha^{-1}(u)\,\Gamma_\alpha^{-1}(v)$ and (b) is due to the fact that by assumption $q_s \geq \Gamma_\alpha\left(\frac{s}{s'}\Gamma_\alpha^{-1}(q_{s'})\right)$ and that the function $\Gamma_\alpha(\cdot)$ is non increasing. This shows that the α-virtual value function of $G_{\alpha,t}(\cdot|(s,q_s),(s',q_{s'}))$ is non decreasing. This concludes the proof. □

Lemma 4.C-2. *Let α in $[0,1]$, $w>0$, q in $(0,1)$, and r in $[\underline{r}_\alpha(w,q),w)\cup[w,\overline{r}_\alpha(w,q)]$. Then the optimal price associated with $F_\alpha(\cdot|r,(w,q))$ is given by r.*

Proof of Lemma 4.C-2. We compute the virtual value function for the function $F_\alpha(\cdot|r,(w,q))$. Since the definition of $F_\alpha(\cdot|r,(w,q))$ depends on whether $r<w$ or $r\geq w$, we treat each case separately.

Case 1: $r\in[\underline{r}_\alpha(w,q),w)$: By applying Lemma 4.A-1 for the pair $((r,1),(w,q))$, we get the virtual value function at $v\geq r$ satisfies

$$
\begin{aligned}
\phi^0_{F_\alpha(\cdot|r,(w,q))}(v) &= \alpha v + (1-\alpha)r - \frac{w-r}{\Gamma_\alpha^{-1}(q)} \\
&\geq r\left(1+\frac{1}{\Gamma_\alpha^{-1}(q)}\right) - \frac{w}{\Gamma_\alpha^{-1}(q)} \\
&= \left(1+\frac{1}{\Gamma_\alpha^{-1}(q)}\right)\left(r-\underline{r}_\alpha(w,q)\right),
\end{aligned}
$$

since $r \geq \underline{r}_\alpha(w,q)$, then we conclude that $\phi^0_{F_\alpha(\cdot|r,(w,q))}(v) \geq 0$. Now, since r is the lower support of the distribution $F_\alpha(\cdot|r,(w,q))$ in the case $r < w$ and $F_\alpha(\cdot|r,(w,q))$ is regular, we conclude that necessarily the optimal price is at r.

Case 2: $r \in [w, \overline{r}_\alpha(w,q)]$**:** In this case, we assume $\overline{r}_\alpha(w,q) \geq w$, otherwise the set is empty. Similarly, by applying Lemma 4.A-1 for the pair $((0,1),(w,q))$, we get that the virtual value function at $v < r$ satisfies

$$\phi^0_{F_\alpha(\cdot|r,(w,q))}(v) \;=\; \alpha v + \left(0 - \frac{w}{\Gamma_\alpha^{-1}(q)}\right) \;=\; \alpha\left(v - \overline{r}_\alpha(w,q)\right).$$

Since $v < r \leq \overline{r}_\alpha(w,q)$, we conclude that $\phi^0_{F_\alpha(\cdot|r,(w,q))}(v) \leq 0$. Now, since r is the upper support of the distribution $F_\alpha(\cdot|r,(w,q))$ in the case $r \geq w$ and $F_\alpha(\cdot|r,(w,q))$ is regular, we conclude that necessarily the optimal price is given by r.

□

Lemma 4.C-3. *The distribution* $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$*, defined in Eq.* (4.3.1)*, belongs to* $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$ *if and only if* (r^*,q^*) *belongs to* $\mathcal{B}_\alpha(w,q)$.

Proof of Lemma 4.C-3. One direction of the proof is direct. In particular, if the distribution $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ belongs to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$ then by definition we have (r^*,q^*) in $\mathcal{B}_\alpha(w,q)$.

Let us now show the other direction, and suppose that (r^*,q^*) belongs to $\mathcal{B}_\alpha(w,q)$ and let F in $\mathcal{F}_\alpha(w,q)$ be a corresponding distribution with $r_F = r^*$ and $q_F = q^*$. We will first show that $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ belongs to $\mathcal{F}_\alpha$ and that the revenue curve of F is lower bounded by the revenue curve of $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$. In a second step, we will show that the optimal revenue of $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ is achieved at r^*.

Step 1: We separate the cases $r^* < w$ and $r^* \geq w$.

Case 1: $r^* < w$. By Lemma 4.1, note that we have that $q^* = \overline{F}(r^*) \geq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)$. By Lemma 4.C-1, applied to the following parameters $(s,q_s) = (r^*,q^*)$ and $(s',q_{s'}) = (w,q)$,

$G_{\alpha,w}(\cdot|(r^*,q^*),(w,q))$ belongs to $\mathcal{F}_\alpha$. Furthermore, by Lemma 4.1 again, we have that

$$v\overline{F}(v) \geq \begin{cases} v\overline{G}_{\alpha,w}(v|(r^*,q^*),(w,q)) & \text{if } v \in [0,r^*], \\ v\overline{G}_{\alpha,w}(v|(r^*,q^*),(w,q)) & \text{if } v \in (r^*,w]. \end{cases}$$

Case 2: $r^* \geq w$. By Lemma 4.1, we have that $q = \overline{F}(w) \geq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\frac{w}{r^*}\right)$ and hence, by Lemma 4.C-1 applied to the following parameters $(s,q_s) = (w,q)$ and $(s',q_{s'}) = (r^*,q^*)$, $G_{\alpha,r^*}(\cdot|(w,q),(r^*,q^*))$ belongs to $\mathcal{F}_\alpha$. Furthermore, by Lemma 4.1 again, we have that

$$v\overline{F}(v) \geq \begin{cases} v\overline{G}_{\alpha,r^*}(v|(w,q),(r^*,q^*)) & \text{if } v \text{ in } [0,w], \\ v\overline{G}_{\alpha,r^*}(v|(w,q),(r^*,q^*)) & \text{if } v \text{ in } [w,r^*]. \end{cases}$$

Therefore in both cases, we have that $v\overline{F}(v) \geq v\overline{G}_{\alpha,r^*\vee w}(v|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ for all v in $[0,+\infty)$ and $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ is α-regular.

Step 2: To conclude the proof we will show that the optimal revenue associated with the distribution $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ is achieved at r^*. We will show that by contradiction.

Since $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ is α-regular, then the associated revenue function is unimodal and achieves its maximum at some point r_G in $[0,+\infty)$. Suppose for a moment $r_G\overline{G}_{\alpha,r^*\vee w}(r_G|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q)) > r^*\overline{G}_{\alpha,r^*\vee w}(r^*|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$. Then, using the above lower-bounds, one would have

$$\begin{aligned} r_G\overline{F}(r_G) &\geq r_G\overline{G}_{\alpha,r^*\vee w}(r_G|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q)) \\ &> r^*\overline{G}_{\alpha,r^*\vee w}(r^*|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q)) \\ &= r^*q^* = r_F\overline{F}(r_F), \end{aligned}$$

which would contradict the optimality of r_F. Hence $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ belongs to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$.

□

Lemma 4.C-4. *A pair (r^*, q^*) in $\mathbb{R}_+ \times [0,1]$ belongs to $\mathcal{B}_\alpha(w,q)$ if and only if it belongs to $\mathcal{B}_l \cup \mathcal{B}_h$, where*

$$\begin{aligned}
\mathcal{B}_l &= \left\{(r^*,q^*) \text{ in } [0,w)\times[0,1] :\ q^* \geq \max\left\{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right), \Gamma_\alpha\left(\frac{1}{\alpha}\right), \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)}\right\}\right\},\\
\mathcal{B}_h &= \left\{(r^*,q^*) \text{ in } [w,+\infty)\times[0,1] :\ q^* \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right),\ q^* \geq q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right)\right\}.
\end{aligned}$$

Proof of Lemma 4.C-4. By Lemma 4.C-3, we have that (r^*, q^*) in $\mathcal{B}_\alpha(w,q)$ if and only if the distribution $G_{\alpha,r^*\vee w}(\cdot|(r^*\wedge w, q^*\vee q),(r^*\vee w, q^*\wedge q))$ belongs to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$.

Case 1: Suppose $r^* < w$. By definition, we have that

$$v\overline{G}_{\alpha,w}(v|(r^*,q^*),(w,q)) = \begin{cases} v\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\,\frac{v}{r^*}\right) & \text{if } v\in[0,r^*],\\ vq^*\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)\frac{v-r^*}{w-r^*}\right) & \text{if } v\in(r^*,w],\\ 0 & \text{if } v\in[w,\infty). \end{cases}$$

By applying Lemma 4.C-1 to the following parameters $(s,q_s) := (r^*,q^*)$ and $(s',q_{s'}) := (w,q)$, we have that $G_{\alpha,w}(\cdot|(r^*,q^*),(w,q))$ belongs to $\mathcal{F}_\alpha(w,q)$ and $\mathcal{F}_\alpha(r^*,q^*)$ if and only if $q^* = \overline{F}(r^*) \geq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)$.

Furthermore, using Lemma 4.A-2, the revenue function $v \mapsto v\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{v}{w}\right)$ is maximized at $r_1 = r^*/(\alpha\Gamma_\alpha^{-1}(q^*))$ and the revenue function $v \mapsto vq^*\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)\frac{v-r^*}{w-r^*}\right)$ is maximized at

$$r_2 = \frac{1}{\alpha}\left(\frac{w-r^*}{\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)} - (1-\alpha)r^*\right).$$

Thus, when $G_{\alpha,w}(\cdot|(r^*,q^*),(w,q))$ belongs to $\mathcal{F}_\alpha(w,q)$, the optimal revenue associated with $G_{\alpha,w}(\cdot|(r^*,q^*),(w,q))$ is achieved at r^* if and only if $r_2 \le r^* \le r_1$. We have $r_2 \le r^*$ if and only if:

$$\frac{1}{\alpha}\left(\frac{w-r^*}{\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)}-(1-\alpha)r^*\right)\le r^* \quad \text{iff} \quad \frac{w-r^*}{\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)}\le r^* \quad \text{iff} \quad \frac{w}{r^*}-1\le \Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right) \text{ iff } q^*\ge \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)},$$

and $r^* \le r_1$ if and only if:

$$r^* \le \frac{r^*}{\alpha\Gamma_\alpha^{-1}(q^*)} \quad \text{iff} \quad \Gamma_\alpha^{-1}(q)\le\frac{1}{\alpha} \quad \text{iff} \quad q^*\ge\Gamma_\alpha\left(\frac{1}{\alpha}\right).$$

Therefore $r_2 \le r^* \le r_1$ is equivalent to

$$q^* \ge \max\left\{\Gamma_\alpha\left(\frac{1}{\alpha}\right), \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)}\right\}.$$

We have established that when $r^* < w$, $G_{\alpha,w}(\cdot|(r^*,q^*),(w,q))$ belongs to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$ if and only if

$$q^* \ge \max\left\{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{r^*}{w}\right), \Gamma_\alpha\left(\frac{1}{\alpha}\right), \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)}\right\}.$$

We have hence established that when $r^* \le w$, (r^*,q^*) belongs to $\mathcal{B}_\alpha(w,q)$ if and only if (r^*,q^*) belongs to $\mathcal{B}_l$.

Case 2: Suppose now $r^* \ge w$. By definition, we have that

$$v\overline{G}_{\alpha,r^*}(v|(w,q),(r^*,q^*)) = \begin{cases} v\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{v}{w}\right) & \text{if } v\in[0,w], \\ vq\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)\frac{v-w}{r^*-w}\right) & \text{if } v\in(w,r^*], \\ 0 & \text{if } v \text{ in } (r^*,\infty). \end{cases}$$

By applying Lemma 4.C-1 to the following parameters $(s, q_s) := (w, q)$ and $(s', q_{s'}) := (r^*, q^*)$, we have that $G_{\alpha,r^*}(\cdot|(w,q),(r^*,q^*))$ belongs to $\mathcal{F}_\alpha(w,q)$ and $\mathcal{F}_\alpha(w^*,\{q^*\})$ if and only if $q = \overline{F}(w) \geq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\frac{w}{r^*}\right)$ which can be rewritten as $q^* \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{r^*}{w}\right)$.

Using Lemma 4.A-2, the oracle price for the revenue function $v \mapsto vq\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)\frac{v-w}{r^*-w}\right)$ is achieved at

$$r' = \frac{1}{\alpha}\left(\frac{r^*-w}{\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)} - (1-\alpha)w\right).$$

Given that r^* is at the end of its support and that the revenue curve is unimodal, the optimal revenue associated $G_{\alpha,r^*}(\cdot|(w,q),(r^*,q^*))$ is achieved at r^* if and only if $r^* \leq r'$, which, in turn, is true if and only if:

$$\begin{aligned} r^* \leq \frac{1}{\alpha}\left(\frac{r^*-w}{\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)} - (1-\alpha)w\right) \quad &\text{iff} \quad \alpha r^* + (1-\alpha)w \leq \frac{r^*-w}{\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)} \\ &\text{iff} \quad \Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right) \leq \frac{r^*-w}{w+\alpha(r^*-w)} \\ &\text{iff} \quad \Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq \frac{q^*}{q} \quad \text{iff} \quad q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq q^*. \end{aligned}$$

We have established that when $r^* \geq w$, $G_{\alpha,r^*}(\cdot|(w,q),(r^*,q^*))$ belongs to $\{F \text{ in } \mathcal{F}_\alpha(w,q) : r_F = r^*, q_F = q^*\}$ if and only if

$$q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq q^* \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{r^*}{w}\right).$$

In turn, we have hence established that when $r^* \geq w$, (r^*,q^*) in $\mathcal{B}_\alpha(w,q)$ if and only if (r^*,q^*) in $\mathcal{B}_h$. This concludes the proof. □

Lemma 4.C-5. *Let $J_{w,q} = \{r :$ there exists q^* s.t. (r^*,q^*) in $\mathcal{B}_\alpha(w,q)\}$. We have $J_{w,q} = [\underline{r}_\alpha(w,q), w) \cup [w, \overline{r}_\alpha(w,q)]$ and for any r^* in $J_{w,q}$, the function $\widetilde{R}_{r^*,w,q}(\cdot)$ is decreasing in the set $\{q^* : (r^*,q^*)$ in $\mathcal{B}_\alpha(w,q)\}$.*

Proof of Lemma 4.C-5. We will show that $J_{w,q} = [\underline{r}_\alpha(w,q), w) \cup [w, \overline{r}_\alpha(w,q)]$ by analyzing the cases when $r^* < w$ and $r^* \geq w$ separately.

Suppose first that $r^* < w$. In this case, we show that there exists a value q^* such that $(r^*, q^*) \in \mathcal{B}_l$ if and only if $r^* \geq \underline{r}_\alpha(w,q)$.

If there exists $(r^*, q^*) \in \mathcal{B}_l$ then we have y Lemma 4.C-4 that $\frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)} \leq q^*$ and since $q^* \leq 1$, we have $\frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)} \leq 1$, which implies that $r^* \geq \frac{w}{1+\Gamma_\alpha^{-1}(q)} = \underline{r}_\alpha(w,q)$. Now if $r^* \in [\underline{r}_\alpha(w,q), w)$, note that $(r^*, 1) \in \mathcal{B}_l$, as we have seen above that $1 \geq \frac{q}{\Gamma_\alpha\left(\frac{w}{r^*}-1\right)}$. Furthermore, we have

$$1 \geq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right) \quad \text{and} \quad 1 \geq \Gamma_\alpha\left(\frac{1}{\alpha}\right)$$

since $\Gamma_\alpha(x) \leq 1$ for all $x \geq 0$.

Now suppose that $r^* \geq w$. In this case, we show that there exists a value q^* such that $(r^*, q^*) \in \mathcal{B}_h$ if and only if $\overline{r}_\alpha(w,q) \geq w$ (which is equivalent to $q \geq \Gamma_\alpha\left(\frac{1}{\alpha}\right)$) and $r^* \leq \overline{r}_\alpha(w,q)$.

If there exists $(r^*, q^*) \in \mathcal{B}_h$ then, by Lemma 4.C-4, we have $q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq q^* \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)$. Note that we have

$$
\begin{aligned}
& q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right) \\
\text{iff}\quad & q^{\alpha-1}\left(\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right)\right)^{\alpha-1} \geq \left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)\right)^{\alpha-1} \\
\text{iff}\quad & \left(1+(1-\alpha)\,\Gamma_\alpha^{-1}(q)\right)\left(1+\frac{1-\alpha}{\alpha+\frac{w}{r^*-w}}\right) \geq 1+(1-\alpha)\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w} \\
\text{iff}\quad & \frac{1-\alpha}{\alpha+\frac{1}{\frac{r^*}{w}-1}}+(1-\alpha)\,\Gamma_\alpha^{-1}(q)\left(1+\frac{1-\alpha}{\alpha+\frac{1}{\frac{r^*}{w}-1}}\right) \geq (1-\alpha)\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w} \\
\text{iff}\quad & \frac{1}{\alpha+\frac{1}{\frac{r^*}{w}-1}}+\Gamma_\alpha^{-1}(q)\left(1+\frac{1-\alpha}{\alpha+\frac{1}{\frac{r^*}{w}-1}}\right) \geq \Gamma_\alpha^{-1}(q)\,\frac{r^*}{w} \\
\text{iff}\quad & \frac{1}{\alpha+\frac{1}{\frac{r^*}{w}-1}}+\Gamma_\alpha^{-1}(q)\left(\frac{1-\alpha}{\alpha+\frac{1}{\frac{r^*}{w}-1}}-(\frac{r^*}{w}-1)\right) \geq 0 \\
\text{iff}\quad & \frac{1+\Gamma_\alpha^{-1}(q)\left(1-\alpha-\alpha\frac{r^*}{w}+\alpha\right)}{\alpha+\frac{1}{\frac{r^*}{w}-1}} \geq 0 \\
\text{iff}\quad & \frac{1-\Gamma_\alpha^{-1}(q)\,\alpha\frac{r^*}{w}}{\alpha+\frac{1}{\frac{r^*}{w}-1}} \geq 0 \\
\text{iff}\quad & 1-\Gamma_\alpha^{-1}(q)\,\alpha\frac{r^*}{w} \geq 0 \quad \text{iff} \quad r^* \leq \frac{w}{\alpha\Gamma_\alpha^{-1}(q)} = \overline{r}_\alpha(w,q).
\end{aligned}
$$

Additionally, since $r^* \geq w$, the above inequality implies that $\overline{r}_\alpha(w,q) \geq w$ (which in turns implies that $q \geq \Gamma_\alpha\left(\frac{1}{\alpha}\right)$).

Now if $q \geq \Gamma_\alpha\left(\frac{1}{\alpha}\right)$ then $\overline{r}_\alpha(w,q) \geq w$ and therefore, if $r^* \in [w, \overline{r}_\alpha(w,q)]$, we always have that $(r^*, \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)) \in \mathcal{B}_h$, as we showed above that $q\Gamma_\alpha\left(\frac{1}{\alpha+\frac{w}{r^*-w}}\right) \leq \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{r^*}{w}\right)$.

Next, we show the $\widetilde{R}_{r^*,w,q}(\cdot)$ monotonicity property by analyzing the cases when $r^* < w$ and $r^* \geq w$ separately.

Suppose first that $r^* < w$. In this case, we have

$$\begin{aligned}
&\widetilde{R}_{r^*,w,q}(q^*) \\
&= \int_0^{r^*} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\frac{p}{r^*}\right)}{q^*} d\Psi(p) + \int_{r^*}^{w} \frac{p}{r^*} \Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)\frac{p-r^*}{w-r^*}\right) d\Psi(p) \\
&= \int_0^{r^*} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\frac{p}{r^*}\right)}{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q^*)\right)} d\Psi(p) + \int_{r^*}^{w} \frac{p}{r^*} \Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)\frac{p-r^*}{w-r^*}\right) d\Psi(p) \\
&= \int_0^{r^*} \frac{p}{r^*} \left(\Gamma_\alpha\left(\frac{1-p/r^*}{1/\Gamma_\alpha^{-1}(q^*)+(1-\alpha)}\right)\right)^{-1} d\Psi(p) + \int_{r^*}^{w} \frac{p}{r^*} \Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q}{q^*}\right)\frac{p-r^*}{w-r^*}\right) d\Psi(p),
\end{aligned}$$

where the last equality follows from the fact that for $u \geq v, \Gamma_\alpha(u)/\Gamma_\alpha(v) = \Gamma_\alpha((u-v)/(1+(1-\alpha)v))$. Each term on the RHS above is decreasing in q^*, since $\Gamma_\alpha(\cdot)$ is decreasing. Hence $\widetilde{R}_{r^*,w,q}(q^*)$ is decreasing in this case.

Suppose now that $r^* \geq w$ and $q \geq \Gamma_\alpha\left(\frac{1}{\alpha}\right)$ so that the interval $[w, \overline{r}_\alpha(w,q)]$ is non-empty. In this case, we have

$$\begin{aligned}
&\widetilde{R}_{r^*,w,q}(q^*) \\
&= \int_0^{w} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p}{w}\right)}{q^*} d\Psi(p) + \int_{w}^{r^*} \frac{p}{r^*} \frac{q\,\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)\frac{p-w}{r^*-w}\right)}{q^*} d\Psi(p) \\
&= \int_0^{w} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p}{w}\right)}{q^*} d\Psi(p) + \int_{w}^{r^*} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)\frac{p-w}{r^*-w}\right)}{\Gamma_\alpha\left(\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)\right)} d\Psi(p) \\
&= \int_0^{w} \frac{p}{r^*} \frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p}{w}\right)}{q^*} d\Psi(p) + \int_{w}^{r^*} \frac{p}{r^*} \left(\Gamma_\alpha\left(\frac{\frac{r^*-p}{r^*-w}}{1/\Gamma_\alpha^{-1}\left(\frac{q^*}{q}\right)+(1-\alpha)\frac{p-w}{r^*-w}}\right)\right)^{-1} d\Psi(p),
\end{aligned}$$

where the last equality follows from the fact that for
$u \geq v, \Gamma_\alpha(u)/\Gamma_\alpha(v) = \Gamma_\alpha((u-v)/(1+(1-\alpha)v))$. Each term in the above equality is decreasing in q^*, therefore the result also holds for this case. This concludes the proof. □

4.D Proofs and auxiliary results for Section 4.4

We define the following useful notation used throughout this section:

$$\tilde{v}_\alpha := \begin{cases} 1 & \text{if } \alpha = 0, \\ \alpha^{\frac{\alpha}{1-\alpha}} & \text{if } \alpha \text{ in } (0,1), \\ e^{-1} & \text{if } \alpha = 1, \end{cases} \tag{4.D-1}$$

$$\underline{q}_\alpha := \begin{cases} 0 & \text{if } \alpha = 0, \\ \Gamma_\alpha\left(\frac{1}{\alpha}\right) & \text{if } \alpha \text{ in } (0,1]. \end{cases} \tag{4.D-2}$$

One can easily check that, for any α in $[0,1]$, $\underline{q}_\alpha < \Gamma_\alpha(\tilde{v}_\alpha)$.

Proof of Theorem 4.2. The proof is divided into three separate parts.

In a first step, we simplify the problem given in (4.4.1) in Section 4.4. We show in Proposition 4.D-1 that the seller's posted price has to counter at most 3 worst-case distributions, where two of these are fixed, and the third one is a function of the price selected. In a second step, we analyze the case of regular distributions, and in a third step, we analyze the case of mhr distributions.

Recall the definition of $\underline{q}_\alpha$ introduced in (4.D-2).

Proposition 4.D-1 (Worst-case Distributions against Deterministic Mechanisms)**.** *For any α in $[0,1]$, we have*

$$\begin{aligned} &\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)\right) \\ &= \begin{cases} \sup_{p\in[0,1]} \min\left\{ \frac{Rev\left(p|\overline{F}_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)}{\text{opt}\left(F_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)}, \frac{Rev(p|\delta_1)}{Rev(1|\delta_1)}, \frac{Rev\left(p|\overline{F}_\alpha(\cdot|\overline{r}_\alpha(1,q),(1,q))\right)}{\text{opt}(F_\alpha(\cdot|\overline{r}_\alpha(1,q),(1,q)))} \right\}, & \text{if } q \in [\underline{q}_\alpha, 1), \\ \sup_{p\in[0,1]} \min\left\{ \frac{Rev\left(p|\overline{F}_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)}{\text{opt}\left(F_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)}, \frac{Rev(p|\delta_1)}{Rev(1|\delta_1)} \right\}, & \text{if } q \in (0, \underline{q}_\alpha), \end{cases} \end{aligned}$$

with

$$\begin{aligned}\mu_{\alpha,q}(p) &= 1-\frac{\sqrt{\Delta_{\alpha,q}(p)}-\alpha\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}\\ \Delta_{\alpha,q}(p) &= \left(\alpha\Gamma_\alpha^{-1}(q)(1-p)\right)^2+4\Gamma_\alpha^{-1}(q)(1-p)q^{\alpha-1},\end{aligned}$$

In the above result, the initial price w is normalized to 1 without loss of generality. The above establishes that, when restricting attention to deterministic prices, one can restrict attention to worst-case distributions consisting of a GPD distribution with support starting at $\mu_{\alpha,q}(p)$ and truncated at 1, a mass at 1, or a GPD distribution truncated at $\overline{r}_\alpha(1,q)$ (when $q \geq \underline{q}_\alpha$). The proof is deferred to Section 4.D.

We now leverage the above reduction to explicitly derive optimal deterministic mechanisms against regular and mhr distributions.

Let us introduce the following functions that represent the ratios of the worst families for $p \geq \underline{r}_\alpha(1,q)$

$$\begin{aligned}R_{1,\alpha}(p,q) &:= \frac{Rev\left(p|\overline{F}_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)}{\text{opt}\left(F_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right)\right)},\\ R_{2,\alpha}(p,q) &:= \frac{Rev(p|\delta_1)}{Rev(1|\delta_1)},\\ R_{3,\alpha}(p,q) &:= \frac{Rev\left(p|\overline{F}_\alpha\left(\cdot|\overline{r}_\alpha(1,q),(1,q)\right)\right)}{\text{opt}\left(F_\alpha\left(\cdot|\overline{r}_\alpha(1,q),(1,q)\right)\right)}, \quad \text{defined only for } q \geq \underline{q}_\alpha.\end{aligned}$$

We next analyze some properties of the above defined ratios. To that end, recall the definitions of $\tilde{v}_\alpha$ introduced in (4.D-1) and of $\underline{q}_\alpha$ in (4.D-2).

Lemma 4.D-1. *We have the following properties:*

1. *If $q \in [\underline{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha))$, then there exists $p_{13,\alpha,q}$ in $\left[\underline{r}_\alpha(1,q),1\right]$ such that $R_{1,\alpha}(\cdot,q) \geq R_{3,\alpha}(\cdot,q)$ in $\left[\underline{r}_\alpha(1,q),p_{13,\alpha,q}\right]$ and $R_{1,\alpha}(\cdot,q) \leq R_{3,\alpha}(\cdot,q)$ in $\left[p_{13,\alpha,q},1\right]$. Else, if $q \in [\Gamma_\alpha(\tilde{v}_\alpha),1]$, then $R_{1,\alpha}(p,q) \geq R_{3,\alpha}(p,q)$ for all p in $\left[\underline{r}_\alpha(1,q),1\right]$.*

2. *For any* $q \in (0,1)$*, there exists* $p_{12,\alpha,q}$ *such that* $R_{1,\alpha}(\cdot,q) \geq R_{2,\alpha}(\cdot,q)$ *in* $\left[\underline{r}_{\alpha}(1,q), p_{12,\alpha,q}\right]$ *and* $R_{1,\alpha}(\cdot,q) \leq R_{2,\alpha}(\cdot,q)$ *in* $\left[p_{12,\alpha,q}, 1\right]$.

3. *If* $q \geq \underline{q}_{\alpha}$*, then we have that* $R_{3,\alpha}(\cdot,q)$ *is non decreasing in* $\left[\underline{r}_{\alpha}(1,q), 1\right]$.

4. *For* α *in* $\{0,1\}$*,* $R_{1,\alpha}(\cdot,q)$ *is non increasing in* $\left[\underline{r}_{\alpha}(1,q), 1\right]$*,* $R_{2,\alpha}(\cdot,q)$ *is non decreasing in* $\left[\underline{r}_{\alpha}(1,q), 1\right]$.

Lemma 4.D-2. *For* α *in* $\{0,1\}$*, there exists a unique* $\hat{q}_{\alpha}$ *in* $[\underline{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})]$ *solution to the equation* $p_{13,\alpha,q} = p_{12,\alpha,q}$*, and we have for* q *in* $[\underline{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})]$: $\quad p_{13,\alpha,q} \leq p_{12,\alpha,q} \quad$ *if and only if* $\quad q \leq \hat{q}_{\alpha}$*. Furthermore, we have the following expressions:*

$$\hat{q}_0 = \frac{1}{4}, \quad p_{13,0,q} = 1 - \frac{(1-2q)^2}{1-q}, \quad p_{12,0,q} = 1 - \frac{\left(1-\sqrt{q}\right)^2}{1-q},$$

$$\hat{q}_1 = \hat{q}, \quad p_{13,1,q} = \mu_{1,q}^{-1}\left(W\left(\frac{1}{\log\left(q^{-1}\right)}\right)\right), \quad p_{12,1,q} = \mu_{1,q}^{-1}\left(\frac{1}{W\left(\frac{e}{q}\right)}\right),$$

Where $\hat{q}$ *is the unique solution in* $[0,1]$ *to the equation* $W\left(\frac{1}{\log(q^{-1})}\right) W\left(\frac{e}{q}\right) = 1$*,* W *is the Lambert function defined as the inverse of* $x \to xe^x$ *in* $[0,+\infty)$*. Numerically* $\hat{q} \in [0.52, 0.53]$.

The proof can be found in Section 4.D. We proceed by analyzing three main cases q in $(0, \hat{q}_{\alpha}]$, q in $(\hat{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})]$ and q in $(\Gamma_{\alpha}(\tilde{v}_{\alpha}), 1)$.

Below, we fix α in $\{0,1\}$.

Case q in $(0, \hat{q}_{\alpha}]$:

We analyze the two sub-cases q in $(0, \underline{q}_{\alpha})$ and q in $[\underline{q}_{\alpha}, \hat{q}_{\alpha}]$, which both will lead the same final result. Note that the first sub-case is empty for $\alpha = 0$.

Sub-case: $\alpha = 1$ and q in $(0, \underline{q}_{\alpha})$: In this case, based on Proposition 4.D-1, we have:

$$\begin{aligned}\mathcal{R}(\mathcal{P}_d, \mathcal{F}_{\alpha}(w,q)) \;=\; \max\Bigg\{ & \max_{p \in \left[\underline{r}_{\alpha}(1,q), p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q)\right), \\ & \max_{p \in \left[p_{12,\alpha,q}, 1\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q)\right) \Bigg\}.\end{aligned}$$

Now let us simplify each term.

- Using Lemma 4.D-1-2, we have that, $R_{1,\alpha}(\cdot, q)$ is above $R_{2,\alpha}(\cdot, q)$ in $\left[\underline{r}_{\alpha}(1,q), p_{12,\alpha,q}\right]$, therefore:

$$\begin{aligned} & \max_{p\in\left[\underline{r}_{\alpha}(1,q),p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q)\right) \\ = & \max_{p\in\left[\underline{r}_{\alpha}(1,q),p_{12,\alpha,q}\right]} R_{2,\alpha}(p,q) \overset{(a)}{=} R_{2,\alpha}\left(p_{12,\alpha,q}, q\right), \end{aligned}$$

  where in (a), we used the result in Lemma 4.D-1-4 that states that $R_{2,\alpha}(\cdot, q)$ is non decreasing in $\left[\underline{r}_{\alpha}(1,q), 1\right]$.

- Using Lemma 4.D-1-2, we have that, $R_{1,\alpha}(\cdot, q)$ is below $R_{2,\alpha}(\cdot, q)$ in $\left[p_{12,\alpha,q}, 1\right]$, therefore:

$$\begin{aligned} & \max_{p\in\left[p_{12,\alpha,q},1\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q)\right) \\ = & \max_{p\in\left[p_{12,\alpha,q},1\right]} R_{1,\alpha}(p,q) \overset{(b)}{=} R_{1,\alpha}\left(p_{12,\alpha,q}, q\right), \end{aligned}$$

  where in (b), we used the result in Lemma 4.D-1-4 that states that $R_{1,\alpha}(\cdot, q)$ is non increasing in $\left[\underline{r}_{\alpha}(1,q), 1\right]$.

Therefore, we have

$$\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_{\alpha}(w,q)\right) = \max\{R_{2,\alpha}\left(p_{12,\alpha,q}, q\right), R_{1,\alpha}\left(p_{12,\alpha,q}, q\right)\} \overset{(c)}{=} p_{12,\alpha,q},$$

where in (c), we used the fact that, by definition, $R_{2,\alpha}\left(p_{12,\alpha,q}, q\right) = R_{1,\alpha}\left(p_{12,\alpha,q}, q\right)$. We also note that the value above is achieved at $p = p_{12,\alpha,q}$.

Sub-case: q in $[\underline{q}_\alpha, \hat{q}_\alpha]$: In this case, based on Lemma 4.D-2, we have that $p_{13,\alpha,q} \leq p_{12,\alpha,q}$, thus we have

$$\begin{aligned}\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) &= \max\Bigg\{ \max_{p\in\left[\underline{r}_\alpha(1,q), p_{13,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right), \\ &\qquad \max_{p\in\left[p_{13,\alpha,q}, p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right), \\ &\qquad \max_{p\in\left[p_{12,\alpha,q}, 1\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) \Bigg\}.\end{aligned}$$

Now let us simplify each term.

- Using Lemma 4.D-1-1, we have that, for $q \in [\underline{q}_\alpha, \hat{q}_\alpha] \subseteq [\underline{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)]$, $R_{1,\alpha}(\cdot, q)$ is above $R_{3,\alpha}(\cdot, q)$ in $\left[\underline{r}_\alpha(1,q), p_{13,\alpha,q}\right]$, therefore:

$$\begin{aligned}&\max_{p\in\left[\underline{r}_\alpha(1,q), p_{13,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) \\ = &\max_{p\in\left[\underline{r}_\alpha(1,q), p_{13,\alpha,q}\right]} \min\left(R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right).\end{aligned}$$

- Using Lemma 4.D-1-1 and Lemma 4.D-1-2, we have that both $R_{3,\alpha}(p,q) \geq R_{1,\alpha}(p,q) \geq R_{2,\alpha}(p,q)$ in $\left[p_{13,\alpha,q}, p_{12,\alpha,q}\right]$, therefore:

$$\begin{aligned}&\max_{p\in\left[p_{13,\alpha,q}, p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) \\ = &\max_{p\in\left[p_{13,\alpha,q}, p_{12,\alpha,q}\right]} R_{2,\alpha}\left(p,q\right).\end{aligned}$$

- Using Lemma 4.D-1-1 and Lemma 4.D-1-2, we have that both $R_{2,\alpha}(\cdot, q)$ and $R_{3,\alpha}(\cdot, q)$ are above $R_{1,\alpha}(\cdot, q)$ in $\left[p_{12,\alpha,q}, 1\right]$, therefore:

$$\begin{aligned}&\max_{p\in\left[p_{12,\alpha,q}, 1\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) \\ = &\max_{p\in\left[p_{12,\alpha,q}, 1\right]} R_{1,\alpha}\left(p,q\right).\end{aligned}$$

Therefore, we have

$$\begin{aligned}\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) &= \max\Bigg\{\max_{p\in\left[\underline{r}_\alpha(1,q),p_{13,\alpha,q}\right]} \min\left(R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right), \\ &\qquad \max_{p\in\left[p_{13,\alpha,q},p_{12,\alpha,q}\right]} R_{2,\alpha}\left(p,q\right), \max_{p\in\left[p_{12,\alpha,q},1\right]} R_{1,\alpha}\left(p,q\right)\Bigg\} \\ &\overset{(a)}{=} \max\Bigg\{\max_{p\in\left[p_{13,\alpha,q},p_{12,\alpha,q}\right]} R_{2,\alpha}\left(p,q\right), \max_{p\in\left[p_{12,\alpha,q},1\right]} R_{1,\alpha}\left(p,q\right)\Bigg\},\end{aligned}$$

where in (a), we used the fact that

$$\begin{aligned}\max_{p\in\left[\underline{r}_\alpha(1,q),p_{13,\alpha,q}\right]} \min\left(R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) &\leq \max_{p\in\left[\underline{r}_\alpha(1,q),p_{13,\alpha,q}\right]} R_{2,\alpha}\left(p,q\right) \\ &\overset{(b)}{\leq} \max_{p\in\left[p_{13,\alpha,q},p_{12,\alpha,q}\right]} R_{2,\alpha}\left(p,q\right),\end{aligned}$$

and in (b), we used the result in Lemma 4.D-1-4 that states that $R_{2,\alpha}\left(\cdot,q\right)$ is non decreasing in $\left[\underline{r}_\alpha(1,q),1\right]$.

Using the latter and also the fact that $R_{1,\alpha}\left(\cdot,q\right)$ is non increasing in $\left[\underline{r}_\alpha(1,q),1\right]$ we have

$$\begin{aligned}\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) &= \max\{R_{2,\alpha}\left(p_{12,\alpha,q},q\right), R_{1,\alpha}\left(p_{12,\alpha,q},q\right)\} \\ &\overset{(c)}{=} p_{12,\alpha,q},\end{aligned}$$

where in (c), we used the fact that, by definition, $R_{2,\alpha}\left(p_{12,\alpha,q},q\right) = R_{1,\alpha}\left(p_{12,\alpha,q},q\right)$. We also note that the value above is achieved at $p = p_{12,\alpha,q}$.

We conclude that, for q in $(0,\hat{q}_\alpha]$, $\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) = p_{12,\alpha,q}$ and is achieved at $p = p_{12,\alpha,q}$. For $\alpha = 0$, using the expressions in Lemma 4.D-2, we obtain $(0,\hat{q}_\alpha] \overset{\alpha=0}{=} (0,\frac{1}{4}]$ and:

$$\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) \overset{\alpha=0}{=} p_{12,0,q} = 1 - \frac{\left(1-\sqrt{q}\right)^2}{1-q}, \quad \text{which is achieved at } p = 1 - \tfrac{\left(1-\sqrt{q}\right)^2}{1-q}.$$

For $\alpha = 1$, using the expressions in Lemma 4.D-2, we obtain $(0, \hat{q}_\alpha] \overset{\alpha=1}{=} (0, \hat{q}]$ and:

$$\begin{aligned}\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) &\overset{\alpha=1}{=} p_{12,1,q} = \mu_{1,q}^{-1}\left(\frac{1}{W\left(\frac{e}{q}\right)}\right), \quad \text{which is achieved at } p = \mu_{1,q}^{-1}\left(\frac{1}{W\left(\frac{e}{q}\right)}\right) \\ &= 1 - \frac{1}{\log\left(q^{-1}\right)}\left(W\left(\frac{e}{q}\right) + \frac{1}{W\left(\frac{e}{q}\right)} - 2\right) := \beta_q\left(\frac{e}{q}\right).\end{aligned}$$

Case q in $(\hat{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)]$: In this case, based on Lemma 4.D-2, we have that $p_{12,\alpha,q} \leq p_{13,\alpha,q}$, thus we have

$$\begin{aligned}\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha\left(w,q\right)\right) = \max\Bigg\{ &\max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right), \\ &\max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right), \\ &\max_{p\in\left[p_{13,\alpha,q}, 1\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right)\Bigg\}.\end{aligned}$$

Now let us simplify each term.

- Using Lemma 4.D-1-1, we have that, for $q \in (\hat{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)] \subseteq [\underline{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)]$, $R_{1,\alpha}(\cdot, q) \geq R_{3,\alpha}(\cdot, q)$ in $\left[\underline{r}_\alpha(1,q), p_{13,\alpha,q}\right]$, therefore:

$$\begin{aligned}&\max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) \\ = &\max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} \min\left(R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right).\end{aligned}$$

- Using Lemma 4.D-1-1 and Lemma 4.D-1-2, we have that both $R_{2,\alpha}(p,q) \geq R_{1,\alpha}(p,q) \geq R_{3,\alpha}(p,q)$ in $\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]$, therefore:

$$\max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} \min\left(R_{1,\alpha}\left(p,q\right), R_{2,\alpha}\left(p,q\right), R_{3,\alpha}\left(p,q\right)\right) = \max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} R_{3,\alpha}\left(p,q\right).$$

- Using Lemma 4.D-1-1 and Lemma 4.D-1-2, we have that both $R_{2,\alpha}(\cdot, q)$ and $R_{3,\alpha}(\cdot, q)$ are above $R_{1,\alpha}(\cdot, q)$ in $\left[p_{13,\alpha,q}, 1\right]$, therefore:

$$\max_{p\in\left[p_{13,\alpha,q},1\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q), R_{3,\alpha}(p,q)\right) = \max_{p\in\left[p_{13,\alpha,q},1\right]} R_{1,\alpha}(p,q).$$

Therefore, we have

$$\begin{aligned}
\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)\right) &= \max\left\{ \max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} \min\left(R_{2,\alpha}(p,q), R_{3,\alpha}(p,q)\right), \right. \\
&\qquad \left. \max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} R_{3,\alpha}(p,q), \max_{p\in\left[p_{13,\alpha,q},1\right]} R_{1,\alpha}(p,q) \right\} \\
&\overset{(a)}{=} \max\left\{ \max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} R_{3,\alpha}(p,q), \max_{p\in\left[p_{13,\alpha,q},1\right]} R_{1,\alpha}(p,q) \right\},
\end{aligned}$$

where in (a), we used the fact that

$$\begin{aligned}
\max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} \min\left(R_{2,\alpha}(p,q), R_{3,\alpha}(p,q)\right) &\leq \max_{p\in\left[\underline{r}_\alpha(1,q), p_{12,\alpha,q}\right]} R_{3,\alpha}(p,q) \\
&\overset{(b)}{\leq} \max_{p\in\left[p_{12,\alpha,q}, p_{13,\alpha,q}\right]} R_{3,\alpha}(p,q),
\end{aligned}$$

and in (b), we used the result in Lemma 4.D-1-3 that states that $R_{3,\alpha}(\cdot, q)$ is non decreasing in $\left[\underline{r}_\alpha(1,q), 1\right]$. Therefore, using the property Lemma 4.D-1-4, we have, additionally, that $R_{1,\alpha}(\cdot, q)$ is non increasing in $\left[\underline{r}_\alpha(1,q), 1\right]$ and therefore

$$\begin{aligned}
\mathcal{R}\left(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)\right) &= \max\left\{R_{3,\alpha}\left(p_{13,\alpha,q}, q\right), R_{1,\alpha}\left(p_{13,\alpha,q}, q\right)\right\} \\
&\overset{(c)}{=} R_{3,\alpha}\left(p_{13,\alpha,q}, q\right), \quad \text{which is achieved at } p = p_{13,\alpha,q}.
\end{aligned}$$

where in (c), we used the fact that $R_{3,\alpha}\left(p_{13,\alpha,q}, q\right) = R_{1,\alpha}\left(p_{13,\alpha,q}, q\right)$.

For $\alpha = 0$, using the expressions in Lemma 4.D-2, we obtain $(\hat{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)] \overset{\alpha=0}{=} (\frac{1}{4}, \frac{1}{2}]$ and:

$$\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) \overset{\alpha=0}{=} R_{3,0}\left(p_{13,0,q}, q\right) = \frac{3-4q}{4(1-q)}, \quad \text{which is achieved at } p_{13,0,q} = \tfrac{q(3-4q)}{1-q}.$$

For $\alpha = 1$, using the expressions in Lemma 4.D-2, we obtain $(\hat{q}_\alpha, \Gamma_\alpha(\tilde{v}_\alpha)] \overset{\alpha=1}{=} (\hat{q}, e^{-e^{-1}}]$ and:

$$\begin{aligned}
\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) &\overset{\alpha=1}{=} R_{3,1}\left(p_{13,1,q}, q\right) \\
&= \mu_{1,q}^{-1}\left(W\left(\frac{1}{\log\left(q^{-1}\right)}\right)\right) e \log\left(q^{-1}\right) e^{-\log\left(q^{-1}\right)\mu_{1,q}^{-1}\left(W\left(\frac{1}{\log\left(q^{-1}\right)}\right)\right)} := \rho(q) \\
&\quad \text{which is achieved at } p = \mu_{1,q}^{-1}\left(W\left(\tfrac{1}{\log\left(q^{-1}\right)}\right)\right) = \beta_q\left(\tfrac{1}{\log\left(q^{-1}\right)}\right).
\end{aligned}$$

Case q in $(\Gamma_\alpha(\tilde{v}_\alpha), 1)$: Using Lemma 4.D-1-1, we have that for q in $(\Gamma_\alpha(\tilde{v}_\alpha), 1)$, $R_{1,\alpha}(\cdot, q)$ is above $R_{3,\alpha}(\cdot, q)$ in $\left[\underline{r}_\alpha(1,q), 1\right]$. Therefore, we have

$$\begin{aligned}
\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) &= \max_{p \in \left[\underline{r}_\alpha(1,q), 1\right]} \min\left(R_{1,\alpha}(p,q), R_{2,\alpha}(p,q), R_{3,\alpha}(p,q)\right) \\
&= \max_{p \in \left[\underline{r}_\alpha(1,q), 1\right]} \min\left(R_{2,\alpha}(p,q), R_{3,\alpha}(p,q)\right) \\
&\overset{(a)}{=} \min\left(R_{2,\alpha}(1,q), R_{3,\alpha}(1,q)\right) \\
&= \min\left(1, R_{3,\alpha}(1,q)\right) \overset{(b)}{=} R_{3,\alpha}(1,q).
\end{aligned}$$

In (a), we used the results in Lemma 4.D-1-3 that states that $R_{3,\alpha}(\cdot, q)$ is non decreasing in $\left[\underline{r}_\alpha(1,q), 1\right]$ and Lemma 4.D-1-4, that states that $R_{2,\alpha}(\cdot, q)$ is non decreasing in $\left[\underline{r}_\alpha(1,q), 1\right]$ as in this case $q \geq \Gamma_\alpha(\tilde{v}_\alpha) \geq \underline{q}_\alpha$. In (b), we used the fact that $R_{3,\alpha}(1,q) \leq 1$.

For $\alpha = 0$, using the expressions in Lemma 4.D-2, we obtain $(\Gamma_\alpha(\tilde{v}_\alpha), 1) \overset{\alpha=0}{=} (\frac{1}{2}, 1)$ and:

$$\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w, \{q\})) \overset{\alpha=0}{=} R_{3,0}(1,q) = 1 - q, \quad \text{which is achieved at } p = 1.$$

For $\alpha = 1$, using the expressions in Lemma 4.D-2, we obtain $(\Gamma_\alpha(\tilde{v}_\alpha), 1) \overset{\alpha=1}{=} (e^{-e^{-1}}, 1)$ and:

$$\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) \overset{\alpha=1}{=} R_{3,1}(1,q) = eq \log\left(q^{-1}\right), \quad \text{which is achieved at } p = 1.$$

This completes the proof of Theorem 4.2. □

Proofs of auxiliary results

Proof of Proposition 4.D-1. Following the reasoning in Section 4.4 in Eq.(4.4.1), we have that:

$$\begin{aligned}
&\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) \\
&= \sup_{p\in[0,w]} \min\left\{ \min_{r\in[\underline{r}_\alpha(w,q),p)} \frac{p\overline{F}_\alpha(p|r,(w,q))}{r}, \frac{p}{w}, \min_{r\in[w,\bar{r}_\alpha(w,q)]} \frac{p\overline{F}_\alpha(p|r,(w,q))}{\mathrm{opt}(F_\alpha(\cdot|r,(w,q)))} \right\} \\
&= \sup_{\frac{p}{w}\in[0,1]} \min\left\{ \inf_{\frac{r}{w}\in[\underline{r}_\alpha(1,q),\frac{p}{w})} \frac{\frac{p}{w}\overline{F}_\alpha(p/w|r/w,(1,q))}{\frac{r}{w}}, \frac{p}{w}, \inf_{\frac{r}{w}\in[1,\bar{r}_\alpha(1,q)]} \frac{\frac{p}{w}\overline{F}_\alpha(p/w|r/w,(1,q))}{\frac{r}{w}\overline{F}_\alpha(r/w|r/w,(1,q))} \right\} \\
&\overset{(a)}{=} \sup_{\tilde{p}\in[0,1]} \min\left\{ \inf_{\tilde{r}\in[\underline{r}_\alpha(1,q),\tilde{p})} \frac{\tilde{p}\overline{F}_\alpha(\tilde{p}|\tilde{r},(1,q))}{\tilde{r}}, \tilde{p}, \inf_{\tilde{r}\in[1,\bar{r}_\alpha(1,q)]} \frac{\tilde{p}\overline{F}_\alpha(\tilde{p}|\tilde{r},(1,q))}{\tilde{r}\overline{F}_\alpha(\tilde{r}|\tilde{r},(1,q))} \right\},
\end{aligned}$$

where in (a) we used two changes of variables to remove the dependency on w, namely $\tilde{p} = p/w$ and $\tilde{r} = r/w$. Note that when $q < \underline{q}_\alpha$, the last term in the brackets does not affect the worst-case. Thus we conclude that

$$\mathcal{R}(\mathcal{P}_d, \mathcal{F}_\alpha(w,q)) = \sup_{p\in[0,1]} \min\left\{ \inf_{r\in[\underline{r}_\alpha(1,q),p)} \frac{p\overline{F}_\alpha(p|r,(1,q))}{r}, p, \inf_{r\in[1,\bar{r}_\alpha(1,q)]} \frac{p\overline{F}_\alpha(p|r,(1,q))}{r\overline{F}_\alpha(r|r,(1,q))} \right\} \quad (4.\text{D-3})$$

For each (normalized) price p in $[0,1]$, we have three terms that determine the worst case performance. We analyze each term separately. The second term is just the identity stemming from nature selecting a point mass at 1. We next analyze the first and third terms with the brackets in (4.D-3).

Third term. The third term is only present if $q \geq \underline{q}_\alpha$ (ensuring that $[1, \overline{r}_\alpha(1,q)] \neq \emptyset$). In this case, for any p in $[0,1]$ and , the third term can be shown to be equal to

$$\inf_{r \in [1, \overline{r}_\alpha(1,q)]} \frac{p\overline{F}_\alpha\left(p|r,(1,q)\right)}{r\overline{F}_\alpha\left(r|r,(1,q)\right)} = \frac{p\overline{F}_\alpha\left(p|\overline{r}_\alpha(1,q),(1,q)\right)}{\sup_{r \in [1, \overline{r}_\alpha(1,q)]} r\overline{F}_\alpha\left(r|r,(1,q)\right)}.$$

Indeed, fix $q \geq \underline{q}_\alpha$. For any $r \in [1, \overline{r}_\alpha(1,q)]$, $\overline{F}_\alpha\left(p|r,(1,q)\right) = \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\, p\right) = \overline{F}_\alpha\left(p|\overline{r}_\alpha(1,q),(1,q)\right)$. By Lemma 4.A-2 applied to $\beta := \Gamma_\alpha^{-1}(q)$ and $w' := 0$, we have that the function $v \to v\overline{F}_\alpha\left(v|\overline{r}_\alpha(1,q),(1,q)\right)$ is maximized at $\overline{r}_\alpha(1,q)$ thus it achieves its maximum at $\overline{r}_\alpha(1,q)$ on the interval $[1, \overline{r}_\alpha(1,q)]$. Hence we get that

$$\begin{aligned} & \inf_{r \in [1, \overline{r}_\alpha(1,q)]} \frac{p\overline{F}_\alpha\left(p|r,(1,q)\right)}{r\overline{F}_\alpha\left(r|r,(1,q)\right)} && \text{(4.D-4)} \\ = & \inf_{r \in [1, \overline{r}_\alpha(1,q)]} \frac{p\overline{F}_\alpha\left(p|\overline{r}_\alpha(1,q),(1,q)\right)}{r\overline{F}_\alpha\left(r|\overline{r}_\alpha(1,q)\,(1,q)\right)} = \frac{p\overline{F}_\alpha\left(p|\overline{r}_\alpha(1,q),(1,q)\right)}{\sup_{r \in [1, \overline{r}_\alpha(1,q)]} r\overline{F}_\alpha\left(r|\overline{r}_\alpha(1,q)\,(1,q)\right)} \\ = & \frac{p\overline{F}_\alpha\left(p|\overline{r}_\alpha(1,q),(1,q)\right)}{\mathrm{opt}(F_\alpha\left(\cdot|\overline{r}_\alpha(1,q),(1,q)\right))}. && \text{(4.D-5)} \end{aligned}$$

One can easily check that

$$\mathrm{opt}(F_\alpha\left(\cdot|\overline{r}_\alpha(1,q),(1,q)\right)) = \lim_{v \to \overline{r}_\alpha(1,q)} v\overline{F}_\alpha\left(v|\overline{r}_\alpha(1,q),(1,q)\right) = \frac{\tilde{v}_\alpha}{\Gamma_\alpha^{-1}(q)},$$

where $\tilde{v}_\alpha$ was defined in (4.D-1).

First term. For any p in $[0,1]$, the first term in (4.D-3) can be rewritten as

$$\inf_{r \in [\underline{r}_\alpha(1,q), p)} \frac{p\overline{F}_\alpha\left(p|r,(1,q)\right)}{r} = \inf_{r \in [\underline{r}_\alpha(1,q), p)} \Phi(r),$$

where

$$\Phi(r) := \frac{p}{r}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{p-r}{1-r}\right).$$

We study the function $\Phi(\cdot)$ and by analyzing its derivative, determine exactly where its minimum is achieved. In particular, we will establish the following claim. On $[\underline{r}_\alpha(1,q), p)$, the function $\Phi(\cdot)$ is minimized at

$$\mu_{\alpha,q}(p) = 1 - \frac{\sqrt{\left(\alpha\Gamma_\alpha^{-1}(q)(1-p)\right)^2 + 4\Gamma_\alpha^{-1}(q)(1-p)q^{\alpha-1}} - \alpha\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}.$$

For any p in $[0,1]$, at any r in $[\underline{r}_\alpha(1,q), p)$, $\Phi(\cdot)$ is differentiable with derivative given by

$$\begin{aligned}
&\frac{d\Phi}{dr}(r) \\
&= -\frac{p}{r^2}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right) - \frac{p}{r}\left(\Gamma_\alpha^{-1}(q)\frac{-(1-r)+(p-r)}{(1-r)^2}\right)\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{2-\alpha} \\
&= -\frac{p}{r^2}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right) - \frac{p}{r}\left(\Gamma_\alpha^{-1}(q)\frac{-(1-p)}{(1-r)^2}\right)\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{2-\alpha} \\
&= -\frac{p}{r^2(1-r)^2}\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{2-\alpha}\left[\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{-(1-\alpha)}(1-r)^2 - \Gamma_\alpha^{-1}(q)\,r(1-p)\right] \\
&= -\frac{p}{r^2(1-r)^2}\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{2-\alpha}\left[\left(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)(1-r)^2 - \Gamma_\alpha^{-1}(q)\,r(1-p)\right] \\
&= -\frac{p}{r^2(1-r)^2}\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-r}{1-r}\right)\right)^{2-\alpha}\left[(1-r)^2 + (1-\alpha)\Gamma_\alpha^{-1}(q)(p-r)(1-r) - \Gamma_\alpha^{-1}(q)\,r(1-p)\right].
\end{aligned}$$

Note that the sign of the derivative of Φ is determined by that of the quadratic

$$\begin{aligned}
\varphi(r) &:= -\left[(1-r)^2 + (1-\alpha)\Gamma_\alpha^{-1}(q)(p-r)(1-r) - \Gamma_\alpha^{-1}(q)\,r(1-p)\right] \\
&= -\Bigg[\left(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\right)(1-r)^2 - (1-\alpha)\Gamma_\alpha^{-1}(q)(1-p)(1-r) \\
&\qquad +\Gamma_\alpha^{-1}(q)(1-r)(1-p) - \Gamma_\alpha^{-1}(q)(1-p)\Bigg] \\
&= -\left[\left(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\right)(1-r)^2 + \alpha\Gamma_\alpha^{-1}(q)(1-p)(1-r) - \Gamma_\alpha^{-1}(q)(1-p)\right] \\
&= -\left[q^{\alpha-1}(1-r)^2 + \alpha\Gamma_\alpha^{-1}(q)(1-p)(1-r) - \Gamma_\alpha^{-1}(q)(1-p)\right].
\end{aligned}$$

Let

$$\Delta_{\alpha,q}(p) \;=\; \left(\alpha\Gamma_{\alpha}^{-1}(q)\,(1-p)\right)^2 + 4q^{\alpha-1}\Gamma_{\alpha}^{-1}(q)\,(1-p).$$

The above is positive and hence the quadratic $\varphi(r)$ admits two roots given by

$$\begin{aligned} r_1 &= 1 + \frac{\alpha\Gamma_{\alpha}^{-1}(q)\,(1-p) + \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}}, \\ r_2 &= 1 + \frac{\alpha\Gamma_{\alpha}^{-1}(q)\,(1-p) - \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}}. \end{aligned}$$

It is clear that $r_1 \geq 1$. We next establish that r_2 belongs to $\left[\underline{r}_{\alpha}(1,q), p\right]$.

$$\begin{aligned} & p - r_2 \\ = \;& p - 1 - \frac{\alpha\Gamma_{\alpha}^{-1}(q)\,(1-p) - \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}} \\ = \;& \frac{-(2q^{\alpha-1} + \alpha\Gamma_{\alpha}^{-1}(q))(1-p) + \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}} \\ = \;& \frac{-\left(4q^{2(\alpha-1)} + \left(\alpha\Gamma_{\alpha}^{-1}(q)\right)^2 + 4q^{\alpha-1}\alpha\Gamma_{\alpha}^{-1}(q)\right)(1-p)^2 + \left(\alpha\Gamma_{\alpha}^{-1}(q)\,(1-p)\right)^2 + 4q^{\alpha-1}\Gamma_{\alpha}^{-1}(q)\,(1-p)}{2q^{\alpha-1}((2q^{\alpha-1} + \alpha\Gamma_{\alpha}^{-1}(q))(1-p) + \sqrt{\Delta_{\alpha,q}(p)})} \\ = \;& 4(1-p)q^{\alpha-1}\frac{-\left(q^{(\alpha-1)} + \alpha\Gamma_{\alpha}^{-1}(q)\right)(1-p) + \Gamma_{\alpha}^{-1}(q)}{2q^{\alpha-1}((2q^{\alpha-1} + \alpha\Gamma_{\alpha}^{-1}(q))(1-p) + \sqrt{\Delta_{\alpha,q}(p)})} \\ = \;& 4(1-p)q^{\alpha-1}\frac{-(1+\Gamma_{\alpha}^{-1}(q))(1-p) + \Gamma_{\alpha}^{-1}(q)}{2q^{\alpha-1}((2q^{\alpha-1} + \alpha\Gamma_{\alpha}^{-1}(q))(1-p) + \sqrt{\Delta_{\alpha,q}(p)})} \\ = \;& 4(1-p)q^{\alpha-1}\Gamma_{\alpha}^{-1}(q)\,\frac{p/\underline{r}_{\alpha}(1,q) - 1}{2q^{\alpha-1}((2q^{\alpha-1} + \alpha\Gamma_{\alpha}^{-1}(q))(1-p) + \sqrt{\Delta_{\alpha,q}(p)})} \\ \geq \;& 0, \end{aligned}$$

where the last inequality follows since $p \geq \underline{r}_{\alpha}(1,q)$.

Now, we also have

$$
\begin{aligned}
& r_2 - \underline{r}_\alpha(1,q) \\
= \;& 1 + \frac{\alpha\Gamma_\alpha^{-1}(q)(1-p) - \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}} - \frac{1}{1+\Gamma_\alpha^{-1}(q)} \\
= \;& \frac{\alpha\Gamma_\alpha^{-1}(q)(1-p) - \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}} + \Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) \\
= \;& \frac{\alpha\Gamma_\alpha^{-1}(q)(1-p) + 2q^{\alpha-1}\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) - \sqrt{\Delta_{\alpha,q}(p)}}{2q^{\alpha-1}} \\
= \;& \frac{(\alpha\Gamma_\alpha^{-1}(q))^2(1-p)^2 + 4q^{2(\alpha-1)}(\Gamma_\alpha^{-1}(q))^2(\underline{r}_\alpha(1,q))^2 + 4q^{\alpha-1}(\Gamma_\alpha^{-1}(q))^2\alpha\underline{r}_\alpha(1,q)(1-p)}{2q^{\alpha-1}(\alpha\Gamma_\alpha^{-1}(q)(1-p) + 2q^{\alpha-1}\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) + \sqrt{\Delta_{\alpha,q}(p)})} \\
& - \frac{\left(\alpha\Gamma_\alpha^{-1}(q)(1-p)\right)^2 + 4q^{\alpha-1}\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}(\alpha\Gamma_\alpha^{-1}(q)(1-p) + 2q^{\alpha-1}\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) + \sqrt{\Delta_{\alpha,q}(p)})} \\
= \;& 4\Gamma_\alpha^{-1}(q)\,q^{\alpha-1}\frac{q^{\alpha-1}\Gamma_\alpha^{-1}(q)(\underline{r}_\alpha(1,q))^2 + \alpha\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q)(1-p) - (1-p)}{2q^{\alpha-1}(\alpha\Gamma_\alpha^{-1}(q)(1-p) + 2q^{\alpha-1}\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) + \sqrt{\Delta_{\alpha,q}(p)})} \\
= \;& 4\Gamma_\alpha^{-1}(q)\,q^{\alpha-1}\frac{q^{\alpha-1}\underline{r}_\alpha(1,q)(p - \underline{r}_\alpha(1,q))}{2q^{\alpha-1}(\alpha\Gamma_\alpha^{-1}(q)(1-p) + 2q^{\alpha-1}\Gamma_\alpha^{-1}(q)\,\underline{r}_\alpha(1,q) + \sqrt{\Delta_{\alpha,q}(p)})} \\
\geq \;& 0,
\end{aligned}
$$

where the last inequality follows since $p \geq \underline{r}_\alpha(1,q)$. Hence we have established that r_2 belongs to $[\underline{r}_\alpha(1,q), p)$, and $r_1 \geq 1 \geq p$.

Now, note that the sign of φ is non-negative on $[r_2, r_1]$ and non-positive on $[0, r_2]$. We deduce that the function Φ is non increasing on $[\underline{r}_\alpha(1,q), r_2]$ and non decreasing on $[r_2, p)$, thus, on $[\underline{r}_\alpha(1,q), p)$, Φ achieves its minimum at $r_2 = \mu_{\alpha,q}(p)$. In other words we have established that for any p in $[0,1]$,

$$
\inf_{r \in [\underline{r}_\alpha(1,q), p)} \frac{p\overline{F}_\alpha(p|r,(1,q))}{r} = \frac{p\overline{F}_\alpha\left(p|\mu_{\alpha,q}(p),(1,q)\right)}{\mu_{\alpha,q}(p)} = \frac{p\overline{F}_\alpha\left(p|\mu_{\alpha,q}(p),(1,q)\right)}{\mathrm{opt}(F_\alpha\left(\cdot|\mu_{\alpha,q}(p),(1,q)\right))}. \quad \text{(4.D-6)}
$$

Combining equations (4.D-3), (4.D-5) and (4.D-6) yields the result. □

Proof of Lemma 4.D-1. We first start by studying the function $p \to \mu_{\alpha,q}(p)$. We have

$$\begin{aligned}
&\mu_{\alpha,q}(p) \\
&= 1 - \frac{\sqrt{\left(\alpha\Gamma_\alpha^{-1}(q)(1-p)\right)^2 + 4\Gamma_\alpha^{-1}(q)(1-p)q^{\alpha-1}} - \alpha\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}} \\
&= 1 + \alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}} - \sqrt{\left(\alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}\right)^2 + 2\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}} \\
&= \frac{1 - 2(1-\alpha)\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}}{1 + \alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}} + \sqrt{\left(\alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}\right)^2 + 2\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}}}.
\end{aligned}$$

The numerator of the above ratio $p \to 1-2(1-\alpha)\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}$ is clearly non-decreasing, and the denominator $p \to 1+\alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}} + \sqrt{\left(\alpha\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}\right)^2 + 2\frac{\Gamma_\alpha^{-1}(q)(1-p)}{2q^{\alpha-1}}}$ is clearly non-increasing. Therefore, by composition, $p \to \mu_{\alpha,q}(p)$ is non-decreasing and $\mu_{\alpha,q}(p)$ in $\left[\mu_{\alpha,q}\left(\frac{1}{1+\Gamma_\alpha^{-1}(q)}\right), \mu_{\alpha,q}(1)\right]$ with:

$$\begin{aligned}
&\mu_{\alpha,q}\left(\frac{1}{1+\Gamma_\alpha^{-1}(q)}\right) \\
&= 1 + \alpha\frac{\Gamma_\alpha^{-1}(q)^2}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)} - \sqrt{\left(\alpha\frac{\Gamma_\alpha^{-1}(q)^2}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}\right)^2 + 2\frac{\Gamma_\alpha^{-1}(q)^2}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}} \\
&= 1 + \alpha\frac{\Gamma_\alpha^{-1}(q)^2}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)} - \frac{\Gamma_\alpha^{-1}(q)}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}\sqrt{\alpha^2\Gamma_\alpha^{-1}(q)^2 + 4q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)} \\
&= 1 + \frac{\Gamma_\alpha^{-1}(q)}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}\left(\alpha\Gamma_\alpha^{-1}(q) - \sqrt{\alpha^2\Gamma_\alpha^{-1}(q)^2 + 4\left(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\right)\left(1+\Gamma_\alpha^{-1}(q)\right)}\right) \\
&= 1 + \frac{\Gamma_\alpha^{-1}(q)}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}\left(\alpha\Gamma_\alpha^{-1}(q) - \sqrt{\left(2+(2-\alpha)\Gamma_\alpha^{-1}(q)\right)^2}\right) \\
&= 1 - \frac{\Gamma_\alpha^{-1}(q)}{2q^{\alpha-1}\left(1+\Gamma_\alpha^{-1}(q)\right)}2\left(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\right) = \frac{1}{1+\Gamma_\alpha^{-1}(q)}.
\end{aligned}$$

$$\mu_{\alpha,q}(1) = 1 + \alpha\frac{\Gamma_\alpha^{-1}(q)(1-1)}{2q^{\alpha-1}} - \sqrt{\left(\alpha\frac{\Gamma_\alpha^{-1}(q)(1-1)}{2q^{\alpha-1}}\right)^2 + 2\frac{\Gamma_\alpha^{-1}(q)(1-1)}{2q^{\alpha-1}}} = 1.$$

Therefore $\mu_{\alpha,q}(p)$ in $\left[\frac{1}{1+\Gamma_\alpha^{-1}(q)}, 1\right]$ and $p \to \mu_{\alpha,q}(p)$ is an increasing function in $\left[\frac{1}{1+\Gamma_\alpha^{-1}(q)}, 1\right]$ and, for any p in $\left[\frac{1}{1+\Gamma_\alpha^{-1}(q)}, 1\right]$, its inverse is given by

$$\mu_{\alpha,q}^{-1}(p) := 1 - \frac{q^{\alpha-1}(1-p)^2}{\Gamma_\alpha^{-1}(q)(1-\alpha(1-p))}.$$

Next, we will show each point separately.

First point If $q \geq \underline{q}_\alpha$, we have

$$\begin{aligned}
\frac{R_{3,\alpha}(p,q)}{R_{1,\alpha}(p,q)} &= \frac{\Gamma_\alpha^{-1}(q)}{\tilde{v}_\alpha}\mu_{\alpha,q}(p)\frac{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,p\right)}{\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-\mu_{\alpha,q}(p)}{1-\mu_{\alpha,q}(p)}\right)} \\
&\overset{(a)}{=} \frac{\Gamma_\alpha^{-1}(q)}{\tilde{v}_\alpha}\mu_{\alpha,q}(p)\,\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-\frac{p-\mu_{\alpha,q}(p)}{1-\mu_{\alpha,q}(p)}}{1+(1-\alpha)\frac{p-\mu_{\alpha,q}(p)}{1-\mu_{\alpha,q}(p)}\Gamma_\alpha^{-1}(q)}\right),
\end{aligned}$$

where in (a), we used the identity $\frac{\Gamma_\alpha(x)}{\Gamma_\alpha(y)} = \Gamma_\alpha\left(\frac{x-y}{1+(1-\alpha)y}\right)$. Let us now focus on simplifying the term (A) inside $\Gamma_\alpha(\cdot)$. We have

$$\begin{aligned}
(A) &= \frac{\Gamma_\alpha^{-1}(q)\,\mu_{\alpha,q}(p)(1-p)}{1-\mu_{\alpha,q}(p)+\Gamma_\alpha^{-1}(q)(1-\alpha)\left(p-\mu_{\alpha,q}(p)\right)} \\
&\overset{(b)}{=} \frac{\Gamma_\alpha^{-1}(q)\,\mu_{\alpha,q}(p)\frac{q^{\alpha-1}\left(1-\mu_{\alpha,q}(p)\right)^2}{\Gamma_\alpha^{-1}(q)\left(1-\alpha\left(1-\mu_{\alpha,q}(p)\right)\right)}}{1-\mu_{\alpha,q}(p)+\Gamma_\alpha^{-1}(q)(1-\alpha)\left(1-\mu_{\alpha,q}(p)-\frac{q^{\alpha-1}\left(1-\mu_{\alpha,q}(p)\right)^2}{\Gamma_\alpha^{-1}(q)\left(1-\alpha\left(1-\mu_{\alpha,q}(p)\right)\right)}\right)} \\
&= \frac{q^{\alpha-1}\mu_{\alpha,q}(p)\left(1-\mu_{\alpha,q}(p)\right)}{1-\alpha\left(1-\mu_{\alpha,q}(p)\right)+(1-\alpha)\left(\Gamma_\alpha^{-1}(q)-\alpha\Gamma_\alpha^{-1}(q)\left(1-\mu_{\alpha,q}(p)\right)-q^{\alpha-1}\left(1-\mu_{\alpha,q}(p)\right)\right)} \\
&= \frac{q^{\alpha-1}\mu_{\alpha,q}(p)\left(1-\mu_{\alpha,q}(p)\right)}{1+(1-\alpha)\Gamma_\alpha^{-1}(q)-\left(1-\mu_{\alpha,q}(p)\right)\left(\alpha+\alpha(1-\alpha)\Gamma_\alpha^{-1}(q)-(1-\alpha)q^{\alpha-1}\right)} \\
&= \frac{q^{\alpha-1}\mu_{\alpha,q}(p)\left(1-\mu_{\alpha,q}(p)\right)}{q^{\alpha-1}-\left(1-\mu_{\alpha,q}(p)\right)q^{\alpha-1}} = 1-\mu_{\alpha,q}(p),
\end{aligned}$$

where in (b), we used that

$$p = \mu_{\alpha,q}^{-1}\left(\mu_{\alpha,q}\left(p\right)\right) = 1 - \frac{q^{\alpha-1}\left(1-\mu_{\alpha,q}\left(p\right)\right)^2}{\Gamma_\alpha^{-1}\left(q\right)\left(1-\alpha\left(1-\mu_{\alpha,q}\left(p\right)\right)\right)}.$$

Therefore we obtain that:

$$\frac{R_{3,\alpha}\left(p,q\right)}{R_{1,\alpha}\left(p,q\right)} = \frac{\Gamma_\alpha^{-1}\left(q\right)}{\tilde{v}_\alpha}\mu_{\alpha,q}\left(p\right)\Gamma_\alpha\left(1-\mu_{\alpha,q}\left(p\right)\right).$$

Since $p \to \mu_{\alpha,q}\left(p\right)$ is non-decreasing and $\Gamma_\alpha\left(\cdot\right)$ is non-increasing, then by composition, we have $p \mapsto R_{13,\alpha,q}\left(p\right) = \frac{R_{3,\alpha}(p,q)}{R_{1,\alpha}(p,q)}$ is non-decreasing in $\left[\underline{r}_\alpha(1,q),1\right]$ and $\frac{R_{3,\alpha}(p,q)}{R_{1,\alpha}(p,q)}$ in $\left[R_{13,\alpha,q}\left(\underline{r}_\alpha(1,q)\right),R_{13,\alpha,q}\left(1\right)\right]$, we have

$$R_{13,\alpha,q}\left(\underline{r}_\alpha(1,q)\right) = \frac{R_{3,\alpha}\left(\frac{1}{1+\Gamma_\alpha^{-1}(q)},q\right)}{R_{1,\alpha}\left(\frac{1}{1+\Gamma_\alpha^{-1}(q)},q\right)} = \frac{\frac{\Gamma_\alpha^{-1}(q)}{1+\Gamma_\alpha^{-1}(q)}\Gamma_\alpha\left(\frac{\Gamma_\alpha^{-1}(q)}{1+\Gamma_\alpha^{-1}(q)}\right)}{\tilde{v}_\alpha} \overset{(a)}{\leq} 1,$$

where in (a), we used the fact that the revenue function $x \to x\Gamma_\alpha\left(x\right)$ is maximized at $x = \frac{1}{\alpha}$ (with the convention that for $\alpha = 0$, $1/\alpha = \infty$) and the maximum value achieved is $\tilde{v}_\alpha$. Furthermore, we have

$$R_{13,\alpha,q}\left(1\right) = \frac{R_{3,\alpha}\left(1,q\right)}{R_{1,\alpha}\left(1,q\right)} = \frac{\Gamma_\alpha^{-1}\left(q\right)}{\tilde{v}_\alpha}.$$

Note that $R_{13,\alpha,q}\left(1\right) \geq 1$ iff $q \leq \Gamma_\alpha\left(\tilde{v}_\alpha\right)$. For q in $(\underline{r}_\alpha(1,q),\Gamma_\alpha\left(\tilde{v}_\alpha\right))$, we define $p_{13,\alpha,q} = R_{13,\alpha,q}^{-1}\left(1\right)$.

Therefore, we have that when q in $\left[\underline{q}_\alpha,\Gamma_\alpha\left(\tilde{v}_\alpha\right)\right)$, then $R_{3,\alpha}\left(p,q\right) \leq R_{1,\alpha}\left(p,q\right)$ if $p \leq p_{13,\alpha,q}$ and $R_{3,\alpha}\left(p,q\right) \geq R_{1,\alpha}\left(p,q\right)$ if $p \geq p_{13,\alpha,q}$. And if q in $\left[\Gamma_\alpha\left(\tilde{v}_\alpha\right),1\right]$ then $R_{1,\alpha}\left(p,q\right) \geq R_{3,\alpha}\left(p,q\right)$ for all p in $\left[\underline{r}_\alpha(1,q),1\right]$.

Second point We have

$$\begin{aligned}
\frac{R_{1,\alpha}(p,q)}{R_{2,\alpha}(p,q)} &= \frac{1}{\mu_{\alpha,q}(p)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\frac{p-\mu_{\alpha,q}(p)}{1-\mu_{\alpha,q}(p)}\right) = \frac{1}{\mu_{\alpha,q}(p)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\left(1-\frac{1-p}{1-\mu_{\alpha,q}(p)}\right)\right) \\
&\overset{(a)}{=} \frac{1}{\mu_{\alpha,q}(p)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\left(1-\frac{q^{\alpha-1}\left(1-\mu_{\alpha,q}(p)\right)}{\Gamma_\alpha^{-1}(q)\left(1-\alpha\left(1-\mu_{\alpha,q}(p)\right)\right)}\right)\right) \\
&= \frac{1}{\mu_{\alpha,q}(p)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)-\frac{q^{\alpha-1}}{\left(\frac{1}{1-\mu_{\alpha,q}(p)}-\alpha\right)}\right) \\
&= \frac{1}{\mu_{\alpha,q}(p)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)-\frac{q^{\alpha-1}}{\left(\frac{1}{1-\mu_{\alpha,q}(p)}-\alpha\right)}\right),
\end{aligned}$$

where in (a), we used:

$$p = \mu_{\alpha,q}^{-1}\left(\mu_{\alpha,q}(p)\right) = 1-\frac{q^{\alpha-1}\left(1-\mu_{\alpha,q}(p)\right)^2}{\Gamma_\alpha^{-1}(q)\left(1-\alpha\left(1-\mu_{\alpha,q}(p)\right)\right)}.$$

Therefore, by composition, we have $p \longrightarrow R_{12,\alpha,q}(p) = \frac{R_{1,\alpha}(p,q)}{R_{2,\alpha}(p,q)}$ is non-increasing in $\left[\underline{r}_\alpha(1,q),1\right]$ and $\frac{R_{1,\alpha}(p,q)}{R_{2,\alpha}(p,q)}$ in $\left[R_{12,\alpha,q}(1), R_{12,\alpha,q}\left(\underline{r}_\alpha(1,q)\right)\right]$, we have

$$\begin{aligned}
R_{12,\alpha,q}\left(\underline{r}_\alpha(1,q)\right) &= \frac{1}{\underline{r}_\alpha(1,q)}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)-\frac{q^{\alpha-1}}{\left(\frac{1}{1-\underline{r}_\alpha(1,q)}-\alpha\right)}\right) \\
&= \left(1+\Gamma_\alpha^{-1}(q)\right)\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)-\frac{q^{\alpha-1}}{\frac{1+(1-\alpha)\Gamma_\alpha^{-1}(q)}{\Gamma_\alpha^{-1}(q)}}\right) \\
&= \left(1+\Gamma_\alpha^{-1}(q)\right) \geq 1, \\
\lim_{p\to 1} R_{12,\alpha,q}(p) &= \lim_{\mu\to 1}\frac{1}{\mu}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)-\frac{q^{\alpha-1}}{\left(\frac{1}{1-\mu}-\alpha\right)}\right) = 0.
\end{aligned}$$

We define $p_{12,\alpha,q} = R_{12,\alpha,q}^{-1}(1)$. Therefore, $R_{1,\alpha}(p,q) \leq R_{2,\alpha}(p,q)$ if $p \leq p_{12,\alpha,q}$ and $R_{1,\alpha}(p,q) \geq R_{2,\alpha}(p,q)$ if $p \geq p_{12,\alpha,q}$.

Third point If q belongs to $[\underline{q}_{\alpha}, 1]$, we have

$$\begin{aligned}
R_{3,\alpha}(p,q) &= \frac{\Gamma_{\alpha}^{-1}(q)}{\tilde{v}_{\alpha}}\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right) \\
\frac{\partial R_{3,\alpha}(p,q)}{\partial p} &= \frac{\Gamma_{\alpha}^{-1}(q)}{\tilde{v}_{\alpha}}\left(\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right) - \Gamma_{\alpha}^{-1}(q)\,p\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right)^{2-\alpha}\right) \\
&= \frac{\Gamma_{\alpha}^{-1}(q)}{\tilde{v}_{\alpha}}\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right)\left(1 - \frac{\Gamma_{\alpha}^{-1}(q)\,p}{1+(1-\alpha)\,\Gamma_{\alpha}^{-1}(q)\,p}\right) \\
&= \begin{cases} \frac{\Gamma_{\alpha}^{-1}(q)^2}{\tilde{v}_{\alpha}}\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right)\left(\frac{\frac{1}{\Gamma_{\alpha}^{-1}(q)}}{1+(1-\alpha)\Gamma_{\alpha}^{-1}(q)p}\right) & \text{if } \alpha = 0, \\ \alpha\frac{\Gamma_{\alpha}^{-1}(q)^2}{\tilde{v}_{\alpha}}\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q)\,p\right)\left(\frac{\overline{r}_{\alpha}(1,q)-p}{1+(1-\alpha)\Gamma_{\alpha}^{-1}(q)p}\right) & \text{if } \alpha \in (0,1]. \end{cases}
\end{aligned}$$

Therefore, if q in $[\underline{q}_{\alpha}, 1]$, then $\overline{r}_{\alpha}(1,q) \geq 1$ and therefore $R_{3,\alpha}(\cdot, q)$ is non-decreasing in $\left[\underline{r}_{\alpha}(1,q), 1\right]$.

Fourth point

Case 1: Regular $\alpha = 0$ In this case, the first function is expressed as follows:

$$\begin{aligned}
R_{1,0}(p,q) &= \frac{p}{\mu_{0,q}(p)\left(1+\left(\frac{1}{q}-1\right)\frac{p-\mu_{0,q}(p)}{1-\mu_{0,q}(p)}\right)} = \frac{pq\left(\mu_{0,q}(p)-1\right)}{\mu_{0,q}(p)\left(p(q-1)-q+\mu_{0,q}(p)\right)} \\
\text{with } \mu_{0,q}(p) &= 1 - \frac{\sqrt{4\left(q^{-1}-1\right)(1-p)\,q^{-1}}}{2q^{-1}} = 1 - \sqrt{(1-p)(1-q)}.
\end{aligned}$$

Therefore

$$\begin{aligned}
R_{1,0}(p,q) &= \frac{p}{\left(1-\sqrt{(1-p)(1-q)}\right)\left(1+\left(\frac{1}{q}-1\right)\frac{p-1+\sqrt{(1-p)(1-q)}}{1-1+\sqrt{(1-p)(1-q)}}\right)} \\
&= \frac{p}{\left(1-\sqrt{(1-p)(1-q)}\right)\left(1+\frac{1-q}{q}\frac{p-1+\sqrt{(1-p)(1-q)}}{\sqrt{(1-p)(1-q)}}\right)} \\
&= \frac{p}{\left(1-\sqrt{(1-p)(1-q)}\right)\left(1+\frac{1-q}{q}\left(1-\frac{\sqrt{1-p}}{\sqrt{1-q}}\right)\right)} \\
&= \frac{pq}{\left(1-\sqrt{(1-p)(1-q)}\right)\left(q+1-q-\sqrt{(1-q)(1-p)}\right)} \\
&= \frac{pq}{\left(1-\sqrt{(1-p)(1-q)}\right)^2}.
\end{aligned}$$

We have, for all p in $[q,1]$:

$$\frac{\partial R_{1,0}}{\partial p}(p,q) = -\frac{q\sqrt{1-p}\left(\sqrt{1-q}-\sqrt{1-p}\right)}{(1-p)\left(1-\sqrt{(1-p)(1-q)}\right)^3} \leq 0 \quad \forall p \text{ in } [q,1],$$

and it is easy to see that function $p \to R_2(p,q)$ is non-decreasing.

Case 2: mhr case $\alpha = 1$ For $\alpha = 1$, the first function is expressed as follows:

$$\begin{aligned}
R_{1,1}(p,q) &= \frac{p}{\mu_{1,q}(p)} e^{-\log\left(q^{-1}\right)\frac{p-\mu_{1,q}(p)}{1-\mu_{1,q}(p)}} \\
\text{with } \mu_{1,q}(p) &= 1 - \frac{\sqrt{\log\left(q^{-1}\right)(1-p))^2 + 4\log\left(q^{-1}\right)(1-p)} - \log\left(q^{-1}\right)(1-p)}{2} \\
\text{and } p &= \mu_{1,q}^{-1}\left(\mu_{1,q}(p)\right) = 1 - \frac{\left(1-\mu_{1,q}(p)\right)^2}{\log\left(q^{-1}\right)\mu_{1,q}(p)}.
\end{aligned}$$

We therefore have

$$\begin{aligned}
R_{1,1}(p,q) &= \frac{1-\frac{\left(1-\mu_{1,q}(p)\right)^2}{\log\left(q^{-1}\right)\mu_{1,q}(p)}}{\mu_{1,q}(p)} e^{-\log\left(q^{-1}\right)\frac{1-\mu_{1,q}(p)-\frac{\left(1-\mu_{1,q}(p)\right)^2}{\log\left(q^{-1}\right)\mu_{1,q}(p)}}{1-\mu_{1,q}(p)}} \\
&= \left(\frac{1}{\mu_{1,q}(p)} - \frac{1}{\log\left(q^{-1}\right)}\left(\frac{1}{\mu_{1,q}(p)}-1\right)^2\right) e^{-\log\left(q^{-1}\right)+\frac{1}{\mu_{1,q}(p)}-1} \\
&= \frac{q}{e\log\left(q^{-1}\right)}\left(\frac{\log\left(q^{-1}\right)}{\mu_{1,q}(p)} - \frac{1}{\mu_{1,q}(p)^2} + \frac{2}{\mu_{1,q}(p)} - 1\right) e^{\frac{1}{\mu_{1,q}(p)}} \\
&= -\frac{q}{e\log\left(q^{-1}\right)}\left(\frac{1}{\mu_{1,q}(p)^2} - \frac{\log\left(q^{-1}\right)+2}{\mu_{1,q}(p)} + 1\right) e^{\frac{1}{\mu_{1,q}(p)}} =: \tilde{R}_1\left(\frac{1}{\mu_{1,q}(p)}\right),
\end{aligned}$$

with

$$\tilde{R}_1(x) = -\frac{q}{e\log\left(q^{-1}\right)}\left(x^2 - \left(2+\log\left(q^{-1}\right)\right)x + 1\right)e^x \quad \text{for } x \text{ in } \left[1, 1+\log\left(q^{-1}\right)\right].$$

On another hand, we have

$$\begin{aligned}
\frac{d\tilde{R}_1(x)}{dx} &= -\frac{q}{e\log\left(q^{-1}\right)}\left(2x-2-\log\left(q^{-1}\right)+x^2-\left(2+\log\left(q^{-1}\right)\right)x+1\right)e^x \\
&= -\frac{q}{e\log\left(q^{-1}\right)}\left(x^2-\log\left(q^{-1}\right)x-\left(1+\log\left(q^{-1}\right)\right)\right)e^x \\
&= -\frac{q}{e\log\left(q^{-1}\right)}(x+1)\left(x-\left(1+\log\left(q^{-1}\right)\right)\right)e^x \geq 0 \quad \forall x \text{ in } \left[1, 1+\log\left(q^{-1}\right)\right].
\end{aligned}$$

For all p in $[\frac{1}{1+\log\left(q^{-1}\right)}, 1]$, we have $\mu_{1,q}(p)$ in $[\frac{1}{1+\log\left(q^{-1}\right)}, 1]$, therefore $\frac{1}{\mu_{1,q}(p)}$ in $[1, 1+\log\left(q^{-1}\right)]$.

Therefore, by composition, $p \to R_{1,1}(p,q)$ is non increasing. The function $p \to R_{2,1}(p,q) = p$ is non-decreasing. □

Proof of Lemma 4.D-2. **Case 1: Regular case $\alpha = 0$.** In this case, for q in $(\underline{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})] \overset{\alpha=0}{=} \left(0, \frac{1}{2}\right]$, $p_{13,0,q}$ is a solution to the following equation

$$\begin{aligned}
&\frac{\Gamma_{\alpha}^{-1}(q)}{\tilde{v}_{\alpha}} \mu_{\alpha,q}(p) \Gamma_{\alpha}\left(1 - \mu_{\alpha,q}(p)\right) = 1 \\
\text{iff} \quad &\left(\frac{1}{q} - 1\right) \frac{1 - \sqrt{(1-p)(1-q)}}{1 + \sqrt{(1-p)(1-q)}} = 1 \\
\text{iff} \quad &\frac{1-2q}{1-q} = \frac{\sqrt{(1-p)(1-q)}}{1-q} \\
\text{iff} \quad &p = p_{13,0,q} = 1 - \frac{(1-2q)^2}{1-q}.
\end{aligned}$$

and $p_{12,0,q}$ is solution to the following equation

$$\begin{aligned}
&\frac{1}{\mu_{\alpha,q}(p)} \Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q) - \frac{q^{\alpha-1}}{\left(\frac{1}{1-\mu_{\alpha,q}(p)} - \alpha\right)}\right) = 1 \\
\text{iff} \quad &\frac{q}{\left(1 - \sqrt{(1-p)(1-q)}\right)^2} = 1 \\
\text{iff} \quad &\sqrt{q} = 1 - \sqrt{(1-p)(1-q)} \\
\text{iff} \quad &\sqrt{(1-p)} = \frac{1 - \sqrt{q}}{\sqrt{1-q}} \\
\text{iff} \quad &p = p_{12,0,q} = 1 - \frac{\left(1 - \sqrt{q}\right)^2}{1-q}.
\end{aligned}$$

Therefore

$$\begin{aligned}
p_{13,0,q} \leq p_{12,0,q} \quad \text{iff} \quad &1 - \frac{(1-2q)^2}{1-q} \leq 1 - \frac{\left(1-\sqrt{q}\right)^2}{1-q} \\
\text{iff} \quad &(1-2q)^2 \geq \left(1 - \sqrt{q}\right)^2 \\
\text{iff} \quad &2q \leq \sqrt{q} \\
\text{iff} \quad &q \leq \frac{1}{4} := \hat{q}_0.
\end{aligned}$$

Furthermore, note that $\hat{q}_0 \in [\underline{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})]$, since $\underline{q}_{\alpha} \overset{\alpha=0}{=} 0$ and $\Gamma_{\alpha}(\tilde{v}_{\alpha}) \overset{\alpha=0}{=} \frac{1}{2}$.

Case 2: mhr case $\alpha = 1$**.** In this case, for q in $(\underline{q}_{\alpha}, \Gamma_{\alpha}(\tilde{v}_{\alpha})] \overset{\alpha=1}{=} \left(e^{-1}, e^{-e^{-1}}\right]$, $p_{13,1,q}$ is solution to the following equation

$$\begin{aligned}
&\frac{\Gamma_{\alpha}^{-1}(q)}{\tilde{v}_{\alpha}}\mu_{\alpha,q}(p)\,\Gamma_{\alpha}\left(1-\mu_{\alpha,q}(p)\right) \overset{\alpha=1}{=} \log\left(q^{-1}\right) e\mu_{1,q}(p)\, e^{\mu_{1,q}(p)-1} = 1 \\
\text{iff}\quad &\mu_{1,q}(p)\, e^{\mu_{1,q}(p)-1} = \frac{1}{\log\left(q^{-1}\right)} \\
\text{iff}\quad &p = p_{13,1,q} = \mu_{1,q}^{-1}\left(W\left(\frac{1}{\log\left(q^{-1}\right)}\right)\right).
\end{aligned}$$

And $p_{12,1,q}$ is solution to the following equation

$$\begin{aligned}
&\frac{1}{\mu_{\alpha,q}(p)}\Gamma_{\alpha}\left(\Gamma_{\alpha}^{-1}(q) - \frac{q^{\alpha-1}}{\left(\frac{1}{1-\mu_{\alpha,q}(p)}-\alpha\right)}\right) \overset{\alpha=1}{=} \frac{1}{\mu_{1,q}(p)} e^{\log(q)+\frac{1}{\frac{1}{1-\mu_{1,q}(p)}-1}} = 1 \\
\text{iff}\quad &\frac{1}{\mu_{1,q}(p)} e^{\frac{1}{\mu_{1,q}(p)}}\frac{q}{e} = 1 \\
\text{iff}\quad &\frac{1}{\mu_{1,q}(p)} e^{\frac{1}{\mu_{1,q}(p)}} = \frac{e}{q} \\
\text{iff}\quad &p = p_{12,1,q} = \mu_{1,q}^{-1}\left(\frac{1}{W\left(\frac{e}{q}\right)}\right).
\end{aligned}$$

Therefore

$$\begin{aligned}
p_{13,1,q} \le p_{12,1,q} \quad \text{iff}\quad &\mu_{1,q}^{-1}\left(W\left(\frac{1}{\log\left(q^{-1}\right)}\right)\right) \le \mu_{1,q}^{-1}\left(\frac{1}{W\left(\frac{e}{q}\right)}\right) \\
\text{iff}\quad &W\left(\frac{1}{\log\left(q^{-1}\right)}\right) \le \frac{1}{W\left(\frac{e}{q}\right)} \quad \text{as } \mu_{1,q}^{-1}(.) \text{ is increasing} \\
\text{iff}\quad &W\left(\frac{1}{\log\left(q^{-1}\right)}\right) W\left(\frac{e}{q}\right) \le 1.
\end{aligned}$$

We now study $q \mapsto g(q) := W\left(\frac{1}{\log\left(q^{-1}\right)}\right) W\left(\frac{e}{q}\right)$ in $\left(e^{-1}, e^{-e^{-1}}\right]$, we have

$$\frac{dg(q)}{dq} = -\frac{W\left(\frac{e}{q}\right) W\left(-\frac{1}{\log(q)}\right)\left(W\left(\frac{e}{q}\right) + \log(q) W\left(-\frac{1}{\log(q)}\right) + \log(q) + 1\right)}{q \log(q)\left(W\left(\frac{e}{q}\right) + 1\right)\left(W\left(-\frac{1}{\log(q)}\right) + 1\right)}.$$

We next analyze the sign of the derivative.

$$\begin{aligned} sign\left(\frac{dg(q)}{dq}\right) &= sign\left(W\left(\frac{e}{q}\right) + \log(q) W\left(-\frac{1}{\log(q)}\right) + \log(q) + 1\right) = sign(h(q)) \\ \text{with } h(q) &:= W\left(\frac{e}{q}\right) + \log(q) W\left(-\frac{1}{\log(q)}\right) + \log(q) + 1 \\ \frac{dh(q)}{dq} &= \frac{\left(W\left(\frac{e}{q}\right) + 1\right) W\left(-\frac{1}{\log(q)}\right)^2 + W\left(-\frac{1}{\log(q)}\right) + 1}{q\left(W\left(\frac{e}{q}\right) + 1\right)\left(W\left(-\frac{1}{\log(q)}\right) + 1\right)} \geq 0 \quad \forall q \text{ in } \left[e^{-1}, e^{-e^{-1}}\right]. \end{aligned}$$

Therefore $q \mapsto h(q)$ is non-decreasing and we have $h\left(e^{-1}\right) = W\left(e^2\right) - W(1) > 0$. Therefore g is increasing in $\left(e^{-1}, e^{-e^{-1}}\right]$. Furthermore, we have $g\left(e^{-1}\right) = W(1) W\left(e^2\right) < 1$ and $g\left(e^{-e^{-1}}\right) = W\left(e^{1+1/e}\right) > 1$. Therefore there exists a unique $\hat{q}_1$ solution in $\left(e^{-1}, e^{-e^{-1}}\right]$ to the equation $W\left(\frac{1}{\log\left(q^{-1}\right)}\right) W\left(\frac{e}{q}\right) = 1$.

Therefore, we have, for q in $\left(e^{-1}, e^{-e^{-1}}\right]$

$$p_{13,1,q} \leq p_{12,1,q} \quad \text{iff} \quad g(q) \leq 1 \quad \text{iff} \quad q \leq \hat{q}_1.$$

□

4.E Proofs and auxiliary results for Section 4.5

Proofs and auxiliary results for Section 4.5.1

Proof of Proposition 4.2. Fix α in $[0,1]$, Ψ in $\mathcal{P}$, q in $(0,1)$, $N > 1$, and a finite sequence of increasing reals $\mathbb{A} = \{a_i\}_{i=1}^N$ such that $0 < a_1 \leq w \leq a_N$. The proof uses two building blocks

associated with uniformly bounding the losses stemming from truncating a mechanism and the losses stemming from local transfers of mass in a mechanism.

Define the two mechanisms Ψ_{a_N} in $\mathcal{P}$ and $\Psi_{\mathbb{A}}$ in $\mathcal{P}_{\mathbb{A}}$ as follows

$$\Psi_{a_N}(v) = \begin{cases} \Psi(v) & \text{if } v \in [0, a_N), \\ 1 & \text{if } v \geq a_N. \end{cases}$$

$$\Psi_{\mathbb{A}}(x) = \begin{cases} 0 & \text{if } x \in [0, a_1). \\ \Psi(a_{i+1}) & \text{if } x \in [a_i, a_{i+1}), \quad \text{for } 1 \leq i \leq N-1, \\ 1 & \text{if } x \in [a_N, \infty). \end{cases}$$

Ψ_{a_N} is a truncated version of Ψ at a_N and $\Psi_{\mathbb{A}}$ is a discretized and truncated version of Ψ.

Let F in $\mathcal{F}_\alpha(w, q)$. Next we analyze $R(\Psi, F) - R(\Psi_{\mathbb{A}}, F)$ by decomposing it as follows.

$$R(\Psi, F) - R(\Psi_{\mathbb{A}}, F) = R(\Psi, F) - R(\Psi_{a_N}, F) + R(\Psi_{a_N}, F) - R(\Psi_{\mathbb{A}}, F). \qquad \text{(4.E-1)}$$

To uniformly bound the maximal losses stemming from truncation $R(\Psi, F) - R(\Psi_{a_N}, F)$, we establish the following result, whose proof is deferred to Section 4.E.

Lemma 4.E-1 (Truncation)**.** *Fix a mechanism Ψ in $\mathcal{P}$, $b \geq w$, q in $(0, 1)$ and let*

$$\Psi_b(v) = \begin{cases} \Psi(v) & \text{if } v \in [0, b), \\ 1 & \text{if } v \geq b. \end{cases}$$

Then for any distribution F in $\mathcal{F}_\alpha(w, q)$,

$$R(\Psi_b, F) \geq R(\Psi, F) - \frac{1}{q\left(1 + (q^{-1} - 1)b/w\right)} \mathbf{1}\{b \leq \overline{r}_\alpha(w, q)\}.$$

In particular, the result upper bounds the maximal performance losses that can stem from truncating a pricing mechanism at b.

To uniformly bound the the impact of discretization $R(\Psi_{a_N}, F) - R(\Psi_{\mathbb{A}}, F)$, we first establish a result (whose proof is deferred to Section 4.E) that bounds the performance losses stemming from transferring mass locally in a mechanism.

Lemma 4.E-2 (Local transfer of mass)**.** *Fix a mechanism Ψ in $\mathcal{P}$, $0 < \epsilon < v$, and let*

$$\Psi_{\epsilon,v}(x) \quad = \quad \begin{cases} \Psi(x) & \text{if } x \in [0, v-\epsilon), \\ \Psi(v) & \text{if } x \in [v-\epsilon, v), \\ \Psi(x) & \text{if } x \geq v. \end{cases}$$

Then, for any distribution F in $\mathcal{F}_\alpha(w, q)$

$$R(\Psi_{\epsilon,v}, F) \geq R(\Psi, F) - \frac{\epsilon}{v}\left(\Psi(v) - \Psi(v-\epsilon)\right).$$

Applying Lemma 4.E-2 on $(v, \epsilon) = (a_i, a_i - a_{i-1})$, $N-1$ times consecutively for $2 \leq i \leq N$ on the mechanism Ψ_{a_N}, we obtain

$$\begin{aligned} R(\Psi_{a_N}, F) - R(\Psi_{\mathbb{A}}, F) \quad &\leq \quad \sum_{i=2}^{N} \frac{a_i - a_{i-1}}{a_i}(\Psi(a_i) - \Psi(a_{i-1})) \\ &\overset{(a)}{\leq} \quad \frac{\Delta(\mathbb{A})}{a_1}\sum_{i=2}^{N}(\Psi(a_i) - \Psi(a_{i-1})) = \frac{\Delta(\mathbb{A})}{a_1}(\Psi(a_N) - \Psi(a_1)) \leq \frac{\Delta(\mathbb{A})}{a_1}. \end{aligned}$$

where (a) follows from $a_i - a_{i-1} \leq \sup_i(a_i - a_{i-1}) = \sigma(\mathbb{A})$ and $a_i \geq a_1 > 0$. Using Lemma 4.E-1, we have

$$R(\Psi_b, F) \geq R(\Psi, F) - \frac{1}{q\left(1 + (q^{-1} - 1)a_N/w\right)}\mathbf{1}\{a_N \leq \overline{r}_\alpha(w, q)\}.$$

Returning to the decomposition in (4.E-1), we have established

$$R(\Psi, F) - R(\Psi_{\mathbb{A}}, F) \quad \leq \quad \frac{\Delta(\mathbb{A})}{a_1} + \frac{1}{q(1 + \left(q^{-1} - 1\right)a_N/w)}\mathbf{1}\{a_N \leq \overline{r}_\alpha(w, q)\}.$$

Noting that the inequality above applies for any F in $\mathcal{F}_\alpha(w,q)$ and that the mechanism $\Psi_{\mathbb{A}}$ does not depend on F, the result follows. □

Proof of Theorem 4.3. This result is a special case of Theorem 4.4.

Proofs of auxiliary results

Proof of Lemma 4.E-1. Let $Rev(q) = qF^{-1}(1-q)$ denote the revenue curve of associated with F in the quantity space. Let r_F denote the optimal oracle price, q_F the corresponding quantity, and recall, from Lemma 4.C-5 that $r_F \leq \overline{r}_\alpha(w,q)$. By definition, we have

$$R(\Psi_b, F) \quad = \quad R(\Psi, F) + \int_b^{+\infty} \frac{Rev(q_b) - Rev(q_x)}{\text{opt}(F)} d\Psi(x).$$

Case 1. Suppose first $b > \overline{r}_\alpha(w,q)$. In this case, then $r_F \leq b$. Given that F is regular, the revenue curve is monotone for $q \leq q_b$, and we have $Rev(q_b) - Rev(q_x)$ for $x \geq b$. We then have

$$R(\Psi_b, F) \geq R(\Psi, F).$$

Case 2. Suppose now that $b \leq \overline{r}_\alpha(w,q)$. In this case, we divide the analysis into two subcases.

Case a). Suppose first that $r_F \leq b$. We have for any $x \geq b \geq r$, by monotonicity of the revenue curve, $Rev(q_b) - Rev(q_x) \geq 0$, and therefore

$$R(\Psi_b, F) \geq R(\Psi, F).$$

Case b). $r_F > b$ We have:

$$\begin{aligned} R(\Psi_b, F) &\geq R(\Psi, F) + \int_b^{+\infty} \frac{Rev(q_b) - Rev(q_x)}{\text{opt}(F)} d\Psi(x) \\ &\geq R(\Psi, F) + \int_b^{+\infty} \left(\frac{Rev(q_b)}{\text{opt}(F)} - 1 \right) d\Psi(x) \\ &\geq R(\Psi, F) + \left(\frac{Rev(q_b)}{\text{opt}(F)} - 1 \right). \end{aligned} \tag{4.E-2}$$

Recall that by assumption $b \geq w$ and hence $q_b \leq q$. Using concavity of the revenue curve in the quantity space (which follows from regularity of F), we have

$$Rev(q_b) \quad \geq \quad Rev(q_F) + \frac{Rev(q) - Rev(q_F)}{q - q_F}(q_b - q_F).$$

This implies that

$$\frac{Rev(q_b)}{Rev(q_F)} \quad \geq \quad 1 + \left(\frac{Rev(q)}{Rev(q_F)} - 1\right)\frac{q_b - q_F}{q - q_F} \quad \geq \quad \frac{q - q_b}{q - q_F} + \frac{Rev(q)}{Rev(q_F)}\frac{q_b - q_F}{q - q_F} \quad \geq \quad \frac{q - q_b}{q - q_F}.$$

Noting that F is regular and using Lemma 4.1, we have

$$q_b \leq \Gamma_0\left(\Gamma_0^{-1}(q)\,\frac{b}{w}\right) = \frac{1}{1 + (q^{-1} - 1)b/w}.$$

Therefore

$$\frac{Rev(q_b)}{\mathrm{opt}(F)} \quad \geq \quad \frac{q}{q - q_F}\left(1 - \frac{1}{q\left(1 + \left(q^{-1} - 1\right)b/w\right)}\right) \geq 1 - \frac{1}{q\left(1 + \left(q^{-1} - 1\right)b/w\right)}.$$

Returning to (4.E-2), we deduce

$$R(\Psi_b, F) \geq R(\Psi, F) - \frac{1}{q(1 + (q^{-1} - 1)b/w)}.$$

Combining both cases, the result follows. □

Proof of Lemma 4.E-2. Let r_F denote the optimal oracle price, q_F the corresponding quantity. We have

$$R(\Psi_{\epsilon,v}, F) \quad = \quad R(\Psi, F) + \int_{v-\epsilon}^{v} \frac{Rev(q_{v-\epsilon}) - Rev(q_x)}{\mathrm{opt}(F)} d\Psi(x).$$

Case 1. Suppose $r_F \leq v - \epsilon$. In this case, using the regularity of F and the unimodality of the revenue curve, we have for any $x \geq v - \epsilon \geq r_F$, $Rev(q_{v-\epsilon}) - Rev(q_x) \geq 0$, and

$$R(\Psi_{\epsilon,v}, F) \geq R(\Psi, F).$$

Case 2. Suppose now $v - \epsilon < r_F \leq v$. In this case, we have

$$\begin{aligned} R(\Psi_{\epsilon,v}, F) &= R(\Psi, F) + \int_{v-\epsilon}^{v} \frac{Rev(q_{v-\epsilon}) - Rev(q_x)}{opt(F)} d\Psi(x) \\ &\geq R(\Psi, F) + \int_{v-\epsilon}^{v} \left(\frac{Rev(q_{v-\epsilon})}{\text{opt}(F)} - 1 \right) d\Psi(x) \\ &= R(\Psi, F) + \left(\frac{Rev(q_{v-\epsilon})}{\text{opt}(F)} - 1 \right) (\Psi(v) - \Psi(v - \epsilon)). \end{aligned}$$

In this case, we have $\text{opt}(F) = r_F q_F \leq v q_{v-\epsilon}$. Therefore

$$\begin{aligned} R(\Psi_{\epsilon,v}, F) &\geq R(\Psi, F) + \left(\frac{v - \epsilon}{v} - 1 \right) (\Psi(v) - \Psi(v - \epsilon)) \\ &\geq R(\Psi, F) - \frac{\epsilon}{v} (\Psi(v) - \Psi(v - \epsilon)). \end{aligned}$$

Case 3. Suppose now $v < r_F$. In this case, for any $v - \epsilon \leq x \leq v < r_F$, by monotonicity of the revenue curve, $Rev(q_{v-\epsilon}) - Rev(q_x) \leq 0$, and furthermore, $Rev(q_x) \leq Rev(q_v) \leq \text{opt}(F)$. In turn,

we have

$$\begin{aligned}
R(\Psi_{\epsilon,v}, F) &= R(\Psi, F) + \int_{v-\epsilon}^{v} \frac{Rev(q_{v-\epsilon}) - Rev(q_x)}{\text{opt}(F)} d\Psi(x) \\
&\geq R(\Psi, F) + \int_{v-\epsilon}^{v} \left(\frac{Rev(q_{v-\epsilon})}{Rev(q_v)} - \frac{Rev(q_x)}{Rev(q_v)} \right) d\Psi(x) \\
&\geq R(\Psi, F) + \int_{v-\epsilon}^{v} \left(\frac{Rev(q_{v-\epsilon})}{Rev(q_v)} - 1 \right) d\Psi(x) \\
&= R(\Psi, F) + \left(\frac{Rev(q_{v-\epsilon})}{Rev(q_v)} - 1 \right) (\Psi(v) - \Psi(v-\epsilon)) \\
&= R(\Psi, F) + \left((1 - \frac{\epsilon}{v}) \frac{q_{v-\epsilon}}{q_v} - 1 \right) (\Psi(v) - \Psi(v-\epsilon)) \\
&\geq R(\Psi, F) + \left((1 - \frac{\epsilon}{v}) - 1 \right) (\Psi(v) - \Psi(v-\epsilon)) \\
&= R(\Psi, F) - \frac{\epsilon}{v} (\Psi(v) - \Psi(v-\epsilon)).
\end{aligned}$$

Combining the three cases yields the result. □

Proofs for Section 4.5.2

Proof of Proposition 4.3. The proof is divided into two steps. In the first step, we will show the lower bound by analyzing the performance of a specific mechanism. Then in a second step, we will derive the upper through the analysis of a family of hard cases when q is close to 0.

Step 1: Lower bound Let us define the following measure:

$$d\Psi(u) = \begin{cases} 0 & \text{if } u < wq, \\ \frac{1}{u \log(\frac{1}{q})} & \text{if } u \text{ in } [qw, w) \\ 0 & \text{if } u \geq w. \end{cases}$$

We have that $\Psi(u)$ is a distribution since

$$\int_0^{\infty} d\Psi(u) = \frac{1}{\log(\frac{1}{q})} \int_{wq}^{w} \frac{1}{u} du = 1,$$

Using Theorem 4.1 and the fact that $\mathcal{F}_\alpha(w,q) \subseteq \mathcal{F}_0(w,q)$ for any $\alpha \in [0,1]$, we have

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) &\geq \inf_{F\in\mathcal{F}_0(w,q)} R(\Psi,F) \\
&= \min\Bigg\{ \inf_{x\in[\underline{r}_0(w,q),w)} \frac{1}{\mathrm{opt}(F_0(\cdot|x,(w,q)))} \int_0^\infty u\overline{F}_0(u|x,(w,q))d\Psi(u), \\
&\qquad \inf_{x\in[w,\bar{r}_0(w,q)]} \frac{1}{\mathrm{opt}(F_0(\cdot|x,(w,q)))} \int_0^\infty u\overline{F}_0(u|x,(w,q)))d\Psi(u) \Bigg\} \\
&= \min\Bigg\{ \inf_{x\in[wq,w)} \frac{1}{x}\left[\int_0^x u d\Psi(u) + \int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right], \\
&\qquad \inf_{x\in[w,\infty)} \frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))} \int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u) \Bigg\}.
\end{aligned}$$

We will analyze each term separately depending whether x in $[wq,w)$ or x in $[w,\infty)$.

Case 1: x **in** $[wq,w)$ We have

$$\begin{aligned}
&\frac{1}{x}\left[\int_0^x u d\Psi(u) + \int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right] \\
&= \frac{1}{x\log(\frac{1}{q})}\left[\int_{wq}^x u du + \int_x^w \overline{G}_{0,w}(u|(x,1),(w,q))du\right] \\
&= \frac{1}{x\log(\frac{1}{q})}\left[x - wq + \int_x^w \frac{1}{1+(1/q-1)\frac{u-x}{w-x}}du\right], \\
&= \frac{1}{x\log(\frac{1}{q})}\left(x - wq + \left[(w-x)\frac{\log(1+(1/q-1)\frac{u-x}{w-x})}{(1/q-1)}\right]_{u=x}^{u=w}\right), \\
&= \frac{1}{x\log(\frac{1}{q})}\left(x - wq + \log(\frac{1}{q})\frac{w-x}{(1/q-1)}\right),
\end{aligned}$$

Hence we get that

$$\begin{aligned}
&\frac{1}{x}\left[\int_0^x u d\Psi(u) + \int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right] \\
= \quad & \frac{1}{\log(\frac{1}{q})}\left(1 + \frac{wq}{x}(\frac{\log(q)}{q-1} - 1) - \frac{\log(\frac{1}{q})}{(1/q-1)}\right) \\
\overset{(a)}{\geq} \quad & \frac{1}{\log(\frac{1}{q})}\left(1 + \frac{wq}{w}(\frac{\log(q)}{q-1} - 1) - \frac{\log(\frac{1}{q})}{(1/q-1)}\right) = \frac{1-q}{\log(\frac{1}{q})},
\end{aligned}$$

where (a) is due to the fact that $\log(q) \leq q - 1 \leq 0$, and $x \leq w$.

Hence we conclude that

$$\frac{1}{x}\left[\int_0^x u d\Psi(u) + \int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right] \geq \frac{1-q}{\log(\frac{1}{q})}. \tag{4.E-3}$$

Case 2: x **in** $[w, \infty)$ Let us now analyze the second term, we have

$$\begin{aligned}
& \frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u) \\
= \quad & \frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\left[\int_0^w u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u) + \int_w^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)\right] \\
= \quad & \frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))\log(\frac{1}{q})}\int_{wq}^w \frac{1}{1+(\frac{1}{q}-1)\frac{u}{w}}du = \frac{\left[w\log(1+(\frac{1}{q}-1)\frac{u}{w})\right]_{u=wq}^{u=w}}{x\overline{G}_{0,x}(x|(0,1),(w,q))(\frac{1}{q}-1)\log(\frac{1}{q})} \\
= \quad & \frac{w}{(\frac{1}{q}-1)\log(\frac{1}{q})}(\log(\frac{1}{q}) - \log(2-q))\frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))} \\
= \quad & \frac{w}{(\frac{1}{q}-1)}(1 - \frac{\log(2-q)}{\log(\frac{1}{q})})\frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}.
\end{aligned}$$

The revenue function $x \to x\overline{G}_{0,x}(x|(0,1),(w,q))$ is non-decreasing in $[0, +\infty)$, therefore

$$x\overline{G}_{0,x}(x|(0,1),(w,q)) \leq \lim_{x\to+\infty} x\overline{G}_{0,x}(x|(0,1),(w,q)) = \frac{w}{(\frac{1}{q}-1)}.$$

Hence

$$\frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u) \quad \geq \quad (1-\frac{\log(2-q)}{\log(\frac{1}{q})}), \quad \text{(4.E-4)}$$

By combining (4.E-3) and (4.E-3), we get that

$$\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) \geq \min\left(\frac{1-q}{\log(\frac{1}{q})}, 1-\frac{\log(2-q)}{\log(\frac{1}{q})}\right).$$

For q in $[0, 1-\frac{1}{\sqrt{2}}]$, we have

$$\frac{1-q}{\log(\frac{1}{q})} \geq \frac{\frac{1}{\sqrt{2}}}{\log(\frac{1}{q})} \geq \frac{\log(2)}{\log(\frac{1}{q})}$$

and

$$1-\frac{\log(2-q)}{\log(\frac{1}{q})} = \frac{\log(\frac{1}{q(2-q)})}{\log(\frac{1}{q})} = \frac{\log(\frac{1}{1-(1-q)^2})}{\log(\frac{1}{q})} \geq \frac{\log(\frac{1}{1-(\frac{1}{\sqrt{2}})^2})}{\log(\frac{1}{q})} = \frac{\log(2)}{\log(\frac{1}{q})}.$$

Hence we we get that

$$\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) \geq \frac{\log(2)}{\log(\frac{1}{q})}.$$

This conclude the lower bound.

Step 2: Upper bound Let q in $(0,1)$ and K in $\mathbb{N}^*$. Define $\varepsilon = q^{\frac{1}{K}}$ in $(q,1)$ and $a_k = w\varepsilon^k$ in $[qw, w)$ for $k = 1\cdots K$. Consider the family of distributions $F_0(\cdot|a_k,(w,q))$ in $\mathcal{F}_0(w,q)$.

Using Yao's principle [126], we have

$$\begin{aligned}\sup_{\Psi\in\mathcal{P}}\inf_{F\in\mathcal{F}_0(w,q)} R(\Psi,F) &\le \sup_{p\ge 0}\frac{1}{K}\sum_{i=1}^{K}\frac{p\bar{F}_0(p|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))}\\ &= \frac{1}{K}\max_{1\le k\le K}\sup_{p\in[a_{k+1},a_k)}\sum_{i=1}^{K}\frac{p\bar{F}_0(p|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))}. \qquad \text{(4.E-5)}\end{aligned}$$

Now let's analyze the sup on each interval $[a_{k+1}, a_k)$. For all $1 \le k \le K$, the revenue curve associated with $\bar{F}_0(\cdot|a_i,(w,q))$ is monotone non-increasing on $[a_i, w)$ as the optimal reserve price of $F_0(\cdot|a_i,(w,q))$ is a_i.

Furthermore, the revenue curve is convex in $[0, w)$ as we have:

$$\left(p\bar{F}_0(p|a_i,(w,q))\right)' = \begin{cases} 1 & \text{if } p \le a_i \\ \frac{w(1-\frac{a_i}{wq})}{(w-a_i)(1+(\frac{1}{q}-1)\frac{p-a_i}{w-a_i})^2} & \text{if } p < w \\ 0 & \text{if } p \ge w. \end{cases}$$

Therefore the derivative of the revenue function $p\bar{F}_0(p|a_i,(w,q))$ is non-decreasing because $a_i \ge wq$ and $p \mapsto \frac{1}{(w-a_i)(1+(\frac{1}{q}-1)\frac{p-a_i}{w-a_i})^2}$ is non-increasing. Hence the function

$$p \mapsto \sum_{i=1}^{K}\frac{p\bar{F}_0(p|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))}$$

is convex on $[a_{k+1}, a_k)$. Thus, the sup on an interval must be attained at one of the extreme points of the interval.

Therefore by (4.E-5), we have that

$$\sup_{\Psi\in\mathcal{P}}\inf_{F\in\mathcal{F}_0(w,q)} R(\Psi,F) \le \sup_{p\ge 0} G(p) = \frac{1}{K}\max_{1\le k\le K}\sum_{i=1}^{K}\frac{a_k\bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))}. \qquad \text{(4.E-6)}$$

Now let us analyze the elementary term, $\frac{a_k\bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))}$ for any i, k. Note that $\mathrm{opt}(F_0(\cdot|a_i,(w,q)) = a_i$ by Lemma 4.C-4.

There are two cases of interest either $i \leq k$ or $i > k$, let us analyze each case separately.

Case 1, $i \leq k$**:** We have $\bar{F}_0(a_k|a_i,(w,q)) = 1$, thus

$$\frac{a_k \bar{F}_{0,a_i,w,q}(a_k)}{\mathrm{opt}(F_{0,a_i,w,q})} = \frac{a_k}{a_i} = \varepsilon^{k-i},$$

which implies that

$$\sum_{i=1}^{k} \frac{a_k \bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))} = \sum_{i=1}^{k} \varepsilon^{k-i} = \sum_{i=0}^{k-1} \varepsilon^{i} = \frac{1-\varepsilon^k}{1-\varepsilon} \leq \frac{1}{1-\varepsilon}.$$

Hence we conclude that

$$\sum_{i=1}^{k} \frac{a_k \bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))} \leq \frac{1}{1-\varepsilon}. \tag{4.E-7}$$

Case 2, $i \geq k$**:** We have that

$$\begin{aligned}
\frac{a_k \bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))} = \frac{a_k}{a_i}\frac{1}{1+(\frac{1}{q}-1)\frac{a_k-a_i}{w-a_i}} &= \varepsilon^{k-i}\frac{1-\varepsilon^i}{1-\varepsilon^i+\frac{1}{q}(\varepsilon^k-\varepsilon^i)-\varepsilon^k+\varepsilon^i} \\
&= \varepsilon^{k-i}\frac{1-\varepsilon^i}{1-\varepsilon^k+\frac{1}{q}(\varepsilon^k-\varepsilon^i)} \\
&= \frac{\varepsilon^{-i}-1}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon^{i-k})} \\
&\leq \frac{\varepsilon^{-i}-1}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon)},
\end{aligned}$$

where in the last inequality we used the fact that $\varepsilon \leq 1$.

Hence we conclude that

$$\begin{aligned}\sum_{i=k+1}^{K} \frac{a_k \bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))} &\leq \sum_{i=k+1}^{K} \frac{\varepsilon^{-i}-1}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon)} \\ &\overset{(a)}{\leq} \frac{1}{\frac{1}{\varepsilon}-1} \frac{\frac{1}{\varepsilon^{K+1}}-\frac{1}{\varepsilon^{k+1}}}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon)} \\ &\leq \frac{1}{1-\varepsilon} \frac{\frac{1}{\varepsilon^{K}}-\frac{1}{\varepsilon^{k}}}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon)} \\ &\overset{(b)}{\leq} \frac{1}{1-\varepsilon} \frac{\frac{1}{q}}{(\varepsilon^{-k}-1)+\frac{1}{q}(1-\varepsilon)},\end{aligned}$$

where in (a) we used $\varepsilon^{-i}-1 \leq \varepsilon^{-i}$ and (b) we used the fact that $\varepsilon \geq 0$ and that $\varepsilon^K = q$. From the last inequality we conclude that

$$\sum_{i=k+1}^{K} \frac{a_k \bar{F}_0(a_k|a_i,(w,q))}{\mathrm{opt}(F_0(\cdot|a_i,(w,q)))} \leq \frac{1}{1-\varepsilon} \frac{1}{q(\varepsilon^{-k}-1)+(1-\varepsilon)} \leq \frac{1}{(1-\varepsilon)^2}. \tag{4.E-8}$$

By combining the last two cases, in particular (4.E-6), (4.E-7) and (4.E-8), we get that

$$\sup_{\Psi \in \mathcal{P}} \inf_{F \in \mathcal{F}_0(w,q)} R(\Psi, F) \leq \frac{1}{K}\left(\frac{1}{1-\varepsilon} + \frac{1}{(1-\varepsilon)^2}\right).$$

By choosing $K = \log(1/q)$, thus $\varepsilon = e^{-1}$, we get:

$$\sup_{\Psi \in \mathcal{P}} \inf_{F \in \mathcal{F}_0(w,q)} R(\Psi, F) \leq \frac{c_2}{\log(1/q)} \text{ with } c_2 = \frac{1}{1-e^{-1}}\left(1 + \frac{1}{1-e^{-1}}\right)$$

This concludes the proof. □

Proof of Proposition 4.4. This proof is divided into two steps. In the first step, we will show the lower bound by analyzing the performance of a specific mechanism. Then in a second step, we will derive the upper through the analysis of a family of hard cases when q is close to 1.

Throughout the proof we will assume that $q \geq 3/4$ since we are interested in the limit when when q is close to 1.

Step 1: Lower bound Let us define the following measure parameterized by $a, b \geq 0$:

$$d\Psi(u) = \begin{cases} a & \text{if } u = w \\ b\frac{(u\overline{G}_{0,u}(u|(0,1),(w,q)))'}{u\overline{G}_{0,u}(u|(0,1),(w,q))} & \text{if } u > w. \end{cases}$$

Note that $d\Psi(u) \geq 0$ since the revenue function $u \to u\overline{G}_{0,u}(u|(0,1),(w,q))d\Psi(u)$ is increasing in $[w, \infty)$. Let us determine the condition on the parameters a and b so that Ψ is a distribution. For that we need the following

$$\int_0^\infty d\Psi(u) = 1,$$

which implies that

$$a + b\log\left(\frac{\lim_{u\to\infty} u\overline{G}_{0,u}(u|(0,1),(w,q))}{w\overline{G}_{0,w}(w|(0,1),(w,q))}\right) = 1.$$

Since $\overline{G}_{0,w}(w|(0,1),(w,q))d\Psi(u) = q$ and $\lim_{u\to\infty} u\overline{G}_{0,u}(u|(0,1),(w,q)) = \frac{w}{\frac{1}{q}-1}$, we get that

$$a + b\log\left(\frac{1}{1-q}\right) = 1.$$

Hence the relation between a and b is as follows

$$b = \frac{1-a}{\log\left(\frac{1}{1-q}\right)} \quad \text{and } a \text{ in } [0,1].$$

Using Theorem 4.1, we have

$$\begin{aligned}\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) &\geq \inf_{F\in\mathcal{F}_0(w,q)} R(\Psi,F)\\ &= \min\Bigg\{\inf_{x\in[wq,w)}\frac{1}{x}\left[\int_0^x u d\Psi(u)+\int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right],\\ &\qquad \inf_{x\in[w,\infty)}\frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)\Bigg\}\end{aligned} \tag{4.E-9}$$

We will analyze each term separately depending if x in $[wq,w)$ or x in $[w,\infty)$.

Case 1: x **in** $[wq,w)$ We have

$$\frac{1}{x}\left[\int_0^x u d\Psi(u)+\int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right]\overset{(a)}{=}\frac{aq}{x}\geq aq,$$

where the last inequality is due to the fact that $x\leq w$ and (a) is due to $d\Psi(u)=0$ for $u<w$, $d\Psi(w)=a$ and $\overline{G}_{0,w}(w|(x,1),(w,q))=q$.

Hence we conclude that

$$\inf_{x\in[wq,w)}\frac{1}{x}\left[\int_0^x u d\Psi(u)+\int_x^w u\overline{G}_{0,w}(u|(x,1),(w,q))d\Psi(u)\right]\geq aq. \tag{4.E-10}$$

Case 2: x **in** $[w,\infty)$ Let us now analyze the second term, we have

$$\begin{aligned}&\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)\\ =\;&\int_0^w u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)+\int_w^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)\\ \overset{(a)}{=}\;&aq+b\int_w^x u\overline{G}_{0,x}(u|(0,1),(w,q))\frac{(u\overline{G}_{0,u}(u|(0,1),(w,q)))'}{u\overline{G}_{0,u}(u|(0,1),(w,q))}du\\ =\;&awq+b\left[u\overline{G}_{0,u}(u|(0,1),(w,q))\right]_w^x\\ \geq\;&awq+b\left(x\overline{G}_{0,x}(x|(0,1),(w,q))-wq\right),\end{aligned}$$

(a) is due to $d\Psi(u) = 0$ for $u < w$, $d\Psi(w) = a$ and $w\overline{G}_{0,w}(w|(0,1),(w,q)) = wq$. Hence we conclude that

$$\begin{aligned}\frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u) &\geq wq\frac{a-b}{x\overline{G}_{0,x}(x|(0,1),(w,q))}+b\\ &\geq (a-b)(1-q)+b,\end{aligned}$$

where the last inequality we used the fact that $x \to x\overline{G}_{0,x}(x|(0,1),(w,q))$ is non-decreasing in $[w,\infty)$ and that $\lim_{x\to\infty} x\overline{G}_{0,x}(x|(0,1),(w,q)) = \frac{wq}{1-q}$.

Thus we conclude that

$$\begin{aligned}&\inf_{x \text{ in } [w,+\infty)} \frac{1}{x\overline{G}_{0,x}(x|(0,1),(w,q))}\int_0^x u\overline{G}_{0,x}(u|(0,1),(w,q))d\Psi(u)\\ \geq\ & (a-b)(1-q)+b. \end{aligned} \tag{4.E-11}$$

By combining (4.E-9), (4.E-10) and (4.E-11) we get that

$$\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) \geq \min\{aq,(a-b)(1-q)+b\}.$$

Now let us set

$$a = \frac{q}{2(q-\frac{1}{2})\log\left(\frac{1}{1-q}\right)+q}.$$

Note that a in $[0,1]$ as q in $[3/4,1]$, this also leads to the fact that

$$\begin{aligned}(a-b)(1-q)+b = aq = \frac{q^2}{2(q-\frac{1}{2})\log\left(\frac{1}{1-q}\right)+q} &\overset{(a)}{\geq} \frac{9}{16}\frac{1}{2(1-\frac{1}{2})\log\left(\frac{1}{1-q}\right)+1}\\ &= \frac{9}{16(\log\left(\frac{1}{1-q}\right)+1)}\\ &\overset{(b)}{\geq} \frac{9}{32}\frac{1}{\log\left(\frac{1}{1-q}\right)},\end{aligned}$$

where inequality (a) stems from the fact q in $[3/4, 1]$, and in (b) we have used $\log\left(\frac{1}{1-q}\right) \geq 1$.

Hence we we get that

$$\inf_{F \in \mathcal{F}_\alpha(w,q)} R(\Psi, F) \geq \frac{9}{32} \frac{1}{\log\left(\frac{1}{1-q}\right)}.$$

This conclude the lower bound.

Step 2: Upper bound To show the upper bound we will introduce a family of "hard" cases. We consider the family of distributions $(F_0(\cdot|r,(w,q))$ in $\mathcal{F}_0(w,q))_{r \geq w}$ and the following weight distribution:

$$d\lambda(r) = \begin{cases} 0 & \text{if } r < w \\ \frac{1}{\log\left(\frac{1}{1-q}\right)} \frac{(r\overline{G}_{0,r}(r|(0,1),(w,q)))'}{r\overline{G}_{0,r}(r|(0,1),(w,q))} & \text{if } r \geq w \end{cases}$$

One can verify that $\int_0^\infty d\lambda(r) = 1$ and that $d\lambda(r) \geq 0$. Let us define

$$G(p) = \int_0^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r).$$

Using Yao's principle [126], we have

$$\sup_{\Psi \in \mathcal{P}} \inf_{F \in \mathcal{F}_0(w,q)} R(\Psi, F) \leq \sup_{p \geq 0} G(p) = \sup_{p \geq 0} \int_0^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r). \quad \text{(4.E-12)}$$

Note that

$$\begin{aligned} \sup_{p \geq 0} \int_0^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r) &= \sup_{p \geq 0} \int_w^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r) \\ &= \sup_{p \geq w} \int_w^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r), \end{aligned}$$

where the last equality follows from the fact that $p \mapsto p\overline{F}_0(p|r,(w,q))$ is increasing on $[0,w]$ for any $r \geq w$.

Fix $p \geq w$ and let us analyze the integral term. We have

$$\begin{aligned}
&\int_w^\infty \frac{p\overline{F}_0(p|r,(w,q))}{\mathrm{opt}(F_0(\cdot|r,(w,q)))} d\lambda(r) \\
= &\int_p^\infty \frac{p\overline{F}_0(p|r,(w,q))}{r\overline{G}_{0,r}(r|(0,1),(w,q))} \frac{1}{\log\left(\frac{1}{1-q}\right)} \frac{(r\overline{G}_{0,r}(r|(0,1),(w,q)))'}{r\overline{G}_{0,r}(r|(0,1),(w,q))} dr \\
= &\frac{1}{\log\left(\frac{1}{1-q}\right)} p\overline{G}_{0,p}(p|(0,1),(w,q)) \int_p^\infty \frac{(r\overline{G}_{0,r}(r|(0,1),(w,q)))'}{(r\overline{G}_{0,r}(r|(0,1),(w,q)))^2} dr \\
= &\frac{1}{\log\left(\frac{1}{1-q}\right)} p\overline{G}_{0,p}(p|(0,1),(w,q)) \left(\frac{1}{p\overline{G}_{0,p}(p|(0,1),(w,q))} - \lim_{r\to\infty} \frac{1}{r\overline{G}_{0,r}(r|(0,1),(w,q))} \right) \\
= &\frac{1}{\log\left(\frac{1}{1-q}\right)} \left(1 - \left(\frac{1}{q} - 1\right) \frac{p\overline{G}_{0,p}(p|(0,1),(w,q))}{w} \right) \leq \frac{1}{\log\left(\frac{1}{1-q}\right)}.
\end{aligned}$$

By using the last inequality, together with (4.E-12), we obtain the result. □

4.F Proofs and auxiliary results for Section 4.6

Proof of Theorem 4.4. We aim to show that one can approximate the value of the maximin ratio via lower and upper bounds and we quantify the asymptotic error of this approximation as a function of the grid size $N > 0$.

We will do that in different steps:

- In a first step, we extend previous results to the interval uncertainty case:
  - We will first show in Proposition 4.F-1 that in the interval uncertainty case, we reduce the family of worst case distributions by generalizing Theorem 4.1.
  - Under such a reduction, we then show in Proposition 4.F-2 that we can still approximate the performance of any mechanism by its discrete version by generalizing Proposition 4.2.

- In a second step, we derive lower bounds on the maximin ratio in the form of linear programs.
- In a third step, we show that through an appropriate choice of the support of a discrete mechanism, one can approximate the maximin ratio arbitrarily closely through the lower bound.

Step 1. We first reduce the possible set of worst-cases to consider by extending Theorem 4.1. For that, let us define the following subset of distributions

$$\mathcal{S}_{\alpha,w,q_l,q_h} = \left\{F_\alpha(\cdot|r,(w,q_l)) : r \text{ in } \left[\underline{r}_\alpha(w,q_l), w\right)\right\} \cup \{F_\alpha(\cdot|r,(w,q_h)) : r \text{ in } [w, \overline{r}_\alpha(w,q_h)]\} \tag{4.F.1}$$

where we use the convention that whenever $\overline{r}_\alpha(w,q_h) < w$, $[w,\overline{r}_\alpha(w,q_h)] := \emptyset$. We have the following result, whose proof is deferred to Section 4.F.

Proposition 4.F-1. *For any q_l, q_h in $(0,1)^2$ such that $q_l \leq q_h$, and for any subset of mechanisms $\mathcal{P}' \subseteq \mathcal{P}$,*

$$\mathcal{R}(\mathcal{P}', \mathcal{F}_\alpha(w,[q_l,q_h])) = \mathcal{R}(\mathcal{P}', \mathcal{S}_{\alpha,w,q_l,q_h}).$$

In addition, the next proposition generalizes Proposition 4.2, and its proof is deferred to Section 4.F.

Proposition 4.F-2. *Let q_l, q_h in $(0,1)^2$ such that $q_l \leq q_h$. Fix a mechanism Ψ in $\mathcal{P}$, $N > 1$, and any finite sequence of increasing reals $\mathbb{A} = \{a_i\}_{i=0}^N$ such that $a_0 = \underline{r}_\alpha(w,q_l), a_N \geq w$. Then there exists $\Psi_\mathbb{A}$ in $\mathcal{P}_\mathbb{A}$ such that*

$$\inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi_\mathbb{A},F) \geq \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F) - \frac{\Delta(\mathbb{A})}{\underline{r}_\alpha(w,q_l)} - \frac{\mathbf{1}\{a_N < \overline{r}_\alpha(w,q_h)\}}{q_h(1+(q_h^{-1}-1)a_N/w)},$$

where $\Delta(\mathbb{A}) = \sup_i\{a_i - a_{i-1}\}$.

Step 2. Fix an arbitrary sequence of increasing reals $\mathbb{A} = \{a_i\}_{i=0}^{2N+1}$ such that $a_0 = \underline{r}_\alpha(w,q_l)$, $a_{N+1} = w$ and $a_{2N+1} \leq \overline{r}_\alpha(w,q_h)$. Set $a_{2N+2} := \overline{r}_\alpha(w,q_h)$. Note that $\overline{r}_\alpha(w,q_h) = \infty$ when

$\alpha = 0$. With some abuse of notation, we will use intervals that include $\overline{r}_\alpha(w, q_h)$. These should be interpreted as open when $\alpha = 0$.

We next develop a lower bound on the maximin ratio $\mathcal{R}(\mathcal{P}_\mathbb{A}, \mathcal{F}_\alpha(w, [q_l, q_h]))$ in the form of a linear program. Fix a mechanism Ψ in $\mathcal{P}_\mathbb{A}$ and denote by $p_0, ..., p_{2N+1}$ the corresponding probabilities. We set $p_{2N+2} := 0$. Using Proposition 4.F-1. Then we have:

$$
\begin{aligned}
&\inf_{F \in \mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \\
= \;& \min\Bigg\{ \inf_{x \in [\underline{r}_\alpha(w,q_l), w)} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_l))d\Psi(u), \\
& \inf_{x \in [w, \overline{r}_\alpha(w,q_h)]} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_h)))d\Psi(u) \Bigg\} \\
= \;& \min\Bigg\{ \min_{i=0,\cdots,N} \inf_{x \in [a_i, a_{i+1})} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_l))d\Psi(u), \\
& \min_{i=N+1,\cdots,2N+1} \inf_{x \in [a_i, a_{i+1}]} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_h)))d\Psi(u) \Bigg\}.
\end{aligned}
$$

Note that, for $x \in [\underline{r}_\alpha(w, q_l), w)$, $\overline{F}_\alpha(\cdot|x, (w, q_l))$ is non-decreasing in x and that the revenue function $u \mapsto u\overline{F}_\alpha(\cdot|x, (w, q_l))$ is increasing in u on $[0, x)$ and decreasing on (x, w). In addition, note that, for $x \in [w, \overline{r}_\alpha(w, q_h)]$, the revenue function $u \mapsto u\overline{F}_\alpha(u|x, (w, q_h))$ is non-decreasing on $[0, x]$. We let $\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-, (w, q_l))) = \lim_{x \to a_{i+1}^-} \mathrm{opt}(F_\alpha(\cdot|x, (w, q_l)))$ for any $i = 0, \cdots, N$. Hence, we have

$$
\begin{aligned}
\inf_{F \in \mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \;\geq\; & \min\Bigg\{ \min_{i=0,\cdots,N} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-, (w, q_l)))} \int_0^\infty u\overline{F}_\alpha(u|a_i, (w, q_l))d\Psi(u), \\
& \min_{i=N+1,\cdots,2N+1} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-, (w, q_h)))} \int_0^\infty u\overline{F}_\alpha(u|a_i, (w, q_h))d\Psi(u) \Bigg\} \\
= \;& \min\Bigg\{ \min_{i=0,\cdots,N} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}^-, (w, q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i, (w, q_l))p_j, \\
& \min_{i=N+1,\cdots,2N+1} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}, (w, q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i, (w, q_h))p_j \Bigg\}, \qquad \text{(4.F-2)}
\end{aligned}
$$

where the equality simply stems from the fact that Ψ in $\mathcal{P}_{\mathbb{A}}$. The problem of maximizing over mechanisms in $\mathcal{P}_{\mathbb{A}}$ is clearly lower bounded by the problem of maximizing the RHS above over $p_0, ..., p_{2N+1}$. The latter problem admits exactly $\mathcal{LP}$-int as its epigraph formulation, and hence we have

$$\mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w, [q_l, q_h])) \quad \geq \quad \underline{\mathcal{L}}_{\alpha, q_l, q_h, \mathbb{A}}.$$

Step 3. We next establish that with a proper choice of sequence $\mathbb{A}$, $\underline{\mathcal{L}}_{\alpha, q_l, q_h, \mathbb{A}}$ may be arbitrarily close to the maximin ratio $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, [q_l, q_h]))$. To do so, we will first develop an upper bound on $\mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w, [q_l, q_h]))$. Then, we will construct a particular sequence $\mathbb{A}$ and establish for this sequence, the gap between $\mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w, [q_l, q_h]))$ and $\underline{\mathcal{L}}_{\alpha, q_l, q_h, \mathbb{A}}$ is small and that the gap between $\mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w, [q_l, q_h]))$ and $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w, [q_l, q_h]))$ is also small. This will yield the result.

Suppose that $a_0 > 0$. Following the same reasoning as in step 2 above, we may also obtain an upper bound on $\inf_{F \in \mathcal{F}_\alpha(w, [q_l, q_h])} R(\Psi, F)$. Indeed, we have

$$\begin{aligned}
\inf_{F \in \mathcal{F}_\alpha(w, [q_l, q_h])} R(\Psi, F) \quad \leq \quad & \min \Bigg\{ \min_{i=0, \cdots, N} \frac{1}{\text{opt}(F_\alpha(\cdot | a_i, (w, q_l)))} \int_0^\infty u \overline{F}_\alpha(u | a_{i+1}^-, (w, q_l)) d\Psi(u), \\
& \min_{i=N+1, \cdots, 2N+1} \frac{1}{\text{opt}(F_\alpha(\cdot | a_i, (w, q_h)))} \int_0^\infty u \overline{F}_\alpha(u | a_{i+1}^-, (w, q_h)) d\Psi(u) \Bigg\} \\
= \quad & \min \Bigg\{ \min_{i=0, \cdots, N} \frac{1}{\text{opt}(F_\alpha(\cdot | a_i, (w, q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j | a_{i+1}^-, (w, q_l)) p_j, \\
& \min_{i=N+1, \cdots, 2N+1} \frac{1}{\text{opt}(F_\alpha(\cdot | a_i, (w, q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j | a_{i+1}^-, (w, q_h)) p_j \Bigg\}.
\end{aligned}$$

With $u\overline{F}_\alpha(u | a_{i+1}^-, (w, q_l)) = \lim_{x \to a_{i+1}^-} u\overline{F}_\alpha(u | x, (w, q_l))$ for any $u \geq 0$ and $i = 0, \cdots, 2N+1$. The problem of maximizing over mechanisms in $\mathcal{P}_{\mathbb{A}}$ is clearly upper bounded by the problem of maximizing the RHS above over $p_0, ..., p_{2N+1}$. The epigraph formulation of the latter problem can be written as

$$\begin{aligned}
\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;=\; & \max_{\mathbf{p},c} \; c && (\mathcal{LP}\text{-int-up})\\
& s.t. \;\; \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_i,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_l))p_j \geq c \quad i=0,\ldots N,\\
& \frac{1}{\mathrm{opt}(F_\alpha(\cdot|a_i,(w,q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_h))p_j \geq c \quad i=N+1,\ldots 2N+1,\\
& \sum_{j=0}^{2N+1} p_j \leq 1, \quad p_i \geq 0 \quad i=0,\ldots 2N+1.
\end{aligned}$$

Therefore, we have

$$\mathcal{R}(\mathcal{P}_{\mathbb{A}},\mathcal{F}_\alpha(w,[q_l,q_h])) \;\leq\; \overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}.$$

Hence, we have established the following.

$$\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \leq \mathcal{R}(\mathcal{P}_{\mathbb{A}},\mathcal{F}_\alpha(w,[q_l,q_h])) \leq \overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}.$$

We next quantify the gap $\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}$ as a function the discretization grid size N for a particular sequence. For $N>1$, $b=\overline{r}_\alpha(w,q_h)$ if $\alpha\in(0,1]$, $b>w$ if $\alpha=0$ and η in $(0,\underline{r}_\alpha(w,q_l))$, consider the following finite sequence of prices $\mathbb{A}=\{a_i\}_{i=0}^{2N+1}$ in $[\underline{r}_\alpha(w,q_l),\min\{b,\overline{r}_\alpha(w,q_h)\}]$:

$$a_i \;=\; \begin{cases} \underline{r}_\alpha(w,q_l)+\frac{i}{N}\left((w-\eta)-\underline{r}_\alpha(w,q_l)\right) & \text{if } 0\leq i\leq N,\\ w+\frac{i-(N+1)}{N}\left(\min\{b,\overline{r}_\alpha(w,q_h)\}-w\right) & \text{if } N+1\leq i\leq 2N+1. \end{cases}$$

When fixing the probability weights $\mathbf{p}$, let $\underline{c}(\mathbf{p})$ denote the maximum value achievable (as a function of c) in the inner problem in ($\mathcal{LP}$-int). In particular, it can be expressed as the minimum in (4.F-2).

Let $\mathbf{p}$ correspond be a probability weight vector corresponding to an optimal solution to the upper bound linear program ($\mathcal{LP}$-int-up). We have

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;\leq\; \overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{c}(\mathbf{p}).$$

We next analyze upper bound the gap $\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{c}(\mathbf{p})$ as a function of the constraints that lead to the minimum value when solving $\underline{c}(\mathbf{p})$.

Case 1: If $\underline{c}(\mathbf{p}) = \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i,(w,q_l))p_j$ for some $0 \leq i \leq N-1$. Then we have

$$\begin{aligned}
& \overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{c}(\mathbf{p}) \\
= \;& \overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i,(w,q_l))p_j \\
\leq \;& \frac{1}{\text{opt}(F_\alpha(\cdot|a_i,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_l))p_j \\
& - \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i,(w,q_l))p_j \\
= \;& \left(\frac{1}{\text{opt}(F_\alpha(\cdot|a_i,(w,q_l)))} - \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))}\right) \sum_{j=0}^{2N} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_l))p_j \\
& + \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \sum_{j=0}^{2N+1} a_j \left[\overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_l)) - \overline{F}_\alpha(a_j|a_i,(w,q_l))\right] p_j \\
= \;& \left(\frac{a_{i+1}-a_i}{a_i}\right) \sum_{j=0}^{2N+1} \frac{a_j \overline{F}_\alpha(a_j|a_{i+1},(w,q_l))}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_l)))} p_j \\
& + \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_l)))} \sum_{j=i+1}^{N+1} a_j \left[\overline{F}_\alpha(a_j|a_{i+1},(w,q_l)) - \overline{F}_\alpha(a_j|a_i,(w,q_l))\right] p_j,
\end{aligned}$$

where in the last equality, we have used that $a_{i+1} = \text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))$ for $0 \leq i \leq N-1$ (cf. Lemma 4.C-2), $\overline{F}_\alpha(\cdot|a_{i+1}^-,(w,q_l)) = \overline{F}_\alpha(\cdot|a_{i+1},(w,q_l))$ for $0 \leq i \leq N-1$ and the fact that $\overline{F}_\alpha(\cdot|a_{i+1},(w,q_l)) = \overline{F}_\alpha(\cdot|a_i,(w,q_l))$ on $[0,a_i]$ and on $(w,+\infty)$. We analyze the two terms on the

RHS above separately.

$$\left(\frac{a_{i+1}-a_i}{a_i}\right)\sum_{j=0}^{2N+1}\frac{a_j\overline{F}_\alpha(a_j|a_{i+1},(w,q_l))}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_l)))}p_j \;\leq\; \left(\frac{a_{i+1}-a_i}{a_i}\right)\sum_{j=0}^{2N+1}p_j \;\leq\; \frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right),$$

where the first inequality follows from the definition of opt, and the second from the fact that $\mathbf{p}$ belongs to the simplex, from definition and from lower bounding a_i by $a_0 = \underline{r}_\alpha(w,q_l)$.

Now, let for $j = i+1,...,N$, $g_j(x) = \overline{F}_\alpha(a_j|x,(w,q_l))$. Note that $g_j(\cdot)$ is differentiable in $[a_i,a_j]$ with derivative bounded as follows

$$\begin{aligned} g_j'(x) &= \left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_l)\,\frac{a_j-x}{w-x}\right)\right)' \\ &= \Gamma_\alpha^{-1}(q_l)\,\frac{(w-a_j)}{(w-x)^2}\left(\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_l)\,\frac{a_j-x}{w-x}\right)\right)^{2-\alpha} \leq \frac{\Gamma_\alpha^{-1}(q_l)}{w-x} \leq \frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}. \end{aligned}$$

We deduce that

$$\begin{aligned} &\frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_l)))}\sum_{j=i+1}^{N+1}a_j\left[\overline{F}_\alpha(a_j|a_{i+1},(w,q_l))-\overline{F}_\alpha(a_j|a_i,(w,q_l))\right]a_jp_j \\ \leq\;& \frac{1}{a_{i+1}}\sum_{j=i+1}^{N+1}\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}(a_{i+1}-a_i)a_jp_j \\ \leq\;& \frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}. \end{aligned}$$

Hence, we have, in this case

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}-\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;\leq\; \frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)\left[1+w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}\right].$$

Case 2: Suppose $\underline{c}(\mathbf{p}) = \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))}\sum_{j=0}^{2N+1}a_j\overline{F}_\alpha(a_j|a_i,(w,q_l))p_j$ for $i=N$. In this case, we have

$$\underline{c}(\mathbf{p}) \;=\; \frac{1}{w}\left(\sum_{j=0}^{N}a_jp_j+wq_lp_{N+1}\right),$$

and

$$\begin{aligned}
\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} &\leq \frac{1}{w-\eta}\left(\sum_{j=0}^{N} a_j p_j + w q_l p_{N+1}\right) - \frac{1}{w}\left(\sum_{j=0}^{N} a_j p_j + w q_l p_{N+1}\right) \\
&\leq \left(\frac{1}{w-\eta} - \frac{1}{w}\right)\left(\sum_{j=0}^{N} a_j p_j + w q_l p_{N+1}\right) \\
&\leq \frac{\eta}{w-\eta} w \sum_{j=0}^{N+1} p_j \\
&\leq \frac{\eta w}{w-\eta}.
\end{aligned}$$

Case 3: Suppose $\underline{c}(\mathbf{p}) = \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i,(w,q_h))p_j$ for some $i = N+1,...,2N$.

$$\begin{aligned}
&\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \\
\leq\ & \frac{1}{\text{opt}(F_\alpha(\cdot|a_i,(w,q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_h))p_j \\
& - \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_h)))} \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_i,(w,q_h))p_j \\
=\ & \left(\frac{1}{\text{opt}(F_\alpha(\cdot|a_i,(w,q_h)))} - \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))}\right) \sum_{j=0}^{2N+1} a_j \overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_h))p_j \\
& + \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))} \sum_{j=0}^{2N+1} a_j \left[\overline{F}_\alpha(a_j|a_{i+1}^-,(w,q_h)) - \overline{F}_\alpha(a_j|a_i,(w,q_h))\right] p_j \\
=\ & \frac{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h))) - \text{opt}(F_\alpha(\cdot|a_i,(w,q_h)))}{\text{opt}(F_\alpha(\cdot|a_i,(w,q_h)))} \sum_{j=0}^{2N+1} \frac{a_j \overline{F}_\alpha(a_j|a_{i+1},(w,q_h))}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))} p_j,
\end{aligned}$$

where in the last equality, we have used that $\overline{F}_\alpha(\cdot|a_{i+1}^-, (w, q_h)) = \overline{F}_\alpha(\cdot|a_i, (w, q_l))$ on $\{a_0, \ldots, a_{i+1}\}$. We analyze the above term on the RHS.

$$\begin{aligned}
&\frac{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}, (w, q_h))) - \mathrm{opt}(F_\alpha(\cdot|a_i, (w, q_h)))}{\mathrm{opt}(F_\alpha(\cdot|a_i, (w, q_h)))} \sum_{j=0}^{2N+1} \frac{a_j \overline{F}_\alpha(a_j|a_{i+1}, (w, q_h))}{\mathrm{opt}(F_\alpha(\cdot|a_{i+1}, (w, q_h)))} p_j \\
&\leq \left(\frac{a_{i+1}\overline{G}_{\alpha,a_{i+1}}(a_{i+1}|(0,1), (w, q_h)) - a_i \overline{G}_{\alpha,a_i}(a_i|(0,1), (w, q_h))}{a_i \overline{G}_{\alpha,a_i}(a_i|(0,1), (w, q_h))} \right) \sum_{j=0}^{2N+1} p_j \\
&\leq \left(\frac{a_{i+1}\overline{G}_{\alpha,a_{i+1}}(a_{i+1}|(0,1), (w, q_h)) - a_i \overline{G}_{\alpha,a_i}(a_i|(0,1), (w, q_h))}{a_i \overline{G}_{\alpha,a_i}(a_i|(0,1), (w, q_h))} \right),
\end{aligned}$$

where the first inequality follows from the definition of opt and $\mathrm{opt}(F_\alpha(\cdot|a_{i+1}, (w, q_h)))$, and the second from the fact that $\mathbf{p}$ belongs to the simplex.

Now, let $g_{\alpha,q_h}(x) = x\overline{G}_{\alpha,x}(x|(0,1), (w, q_h))$. Note that $g_{\alpha,q_h}(\cdot)$ is differentiable in $[w, \overline{r}_\alpha(w, q_h))$ with derivative bounded as follows

$$\begin{aligned}
g'_{\alpha,q_h}(x) &= \left(x\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\right)' \\
&= \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right) - \frac{\Gamma_\alpha^{-1}(q_h)\,x}{w}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)^{2-\alpha} \\
&= \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\left(1 - \frac{\Gamma_\alpha^{-1}(q_h)\,x}{w}\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)^{1-\alpha}\right) \\
&= \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\left(1 - \frac{\Gamma_\alpha^{-1}(q_h)\,x}{w\left(1 + (1-\alpha)\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)}\right) \\
&= \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\left(1 - \frac{\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}{1 + (1-\alpha)\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}\right) \\
&= \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\frac{1 - \alpha\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}{1 + (1-\alpha)\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}.
\end{aligned}$$

Therefore, since $x \leq \overline{r}_\alpha(w, q_h) := \frac{w}{\alpha\Gamma_\alpha^{-1}(q_h)}$, we have that

$$|g'_{\alpha,q_h}(x)| = \left|\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\frac{1 - \alpha\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}{1 + (1-\alpha)\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}\right| = \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q_h)\frac{x}{w}\right)\frac{1 - \alpha\Gamma_\alpha^{-1}(q_h)\frac{x}{w}}{1 + (1-\alpha)\Gamma_\alpha^{-1}(q_h)\frac{x}{w}} \leq 1.$$

We deduce that

$$\begin{aligned}\left(\frac{a_{i+1}\overline{G}_{\alpha,a_{i+1}}(a_{i+1}|(0,1),(w,q_h)) - a_i\overline{G}_{\alpha,a_i}(a_i|(0,1),(w,q_h))}{a_i\overline{G}_{\alpha,a_i}(a_i|(0,1),(w,q_h))}\right) &\leq \frac{a_{i+1}-a_i}{g_{\alpha,q_h}(a_i)}\\ &\leq \frac{\min\{b,\overline{r}_\alpha(w,q_h)\}-w}{Ng_{\alpha,q_h}(w)}\\ &= \frac{\min\{b,\overline{r}_\alpha(w,q_h)\}-w}{Nwq}.\end{aligned}$$

Hence, we have, in this case

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \leq \frac{\min\{b,\overline{r}_\alpha(w,q_h)\}-w}{Nwq}.$$

Case 4: Suppose $\underline{c}(\mathbf{p}) = \frac{1}{\text{opt}(F_\alpha(\cdot|a_{i+1},(w,q_h)))}\sum_{j=0}^{2N+1} a_j\overline{F}_\alpha(a_j|a_i,(w,q_h))p_j$ for $i = 2N+1$.

$$\begin{aligned}&\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}\\ \leq\ & \frac{1}{\text{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))}\sum_{j=0}^{2N+1} a_j\overline{F}_\alpha(a_j|a^-_{2N+1},(w,q_h))p_j\\ &-\frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))}\sum_{j=0}^{2N+1} a_j\overline{F}_\alpha(a_j|a_{2N},(w,q_h))p_j\\ =\ & \left(\frac{1}{\text{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))} - \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))}\right)\sum_{j=0}^{2N+1} a_j\overline{F}_\alpha(a_j|a^-_{2N+2},(w,q_h))p_j\\ &+\frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))}\sum_{j=0}^{2N+1} a_j\left[\overline{F}_\alpha(a_j|a^-_{2N+2},(w,q_h)) - \overline{F}_\alpha(a_j|a_{2N+1},(w,q_h))\right]p_j\\ =\ & \frac{\text{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h))) - \text{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))}{\text{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))}\sum_{j=0}^{2N+1}\frac{a_j\overline{F}_\alpha(a_j|a^-_{2N+2},(w,q_h))}{\text{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))}p_j,\end{aligned}$$

where in the last equality, we have used the fact that $\overline{F}_\alpha(\cdot|a^-_{2N+2},(w,q_h)) = \overline{F}_\alpha(\cdot|a_{2N+1},(w,q_l))$ on $\{a_0,\cdots,a_{2N+1}\}$.

We analyze the above term on the RHS in two separate cases $\alpha \in (0,1]$ and $\alpha = 0$.

In the case where $\alpha \in (0,1]$, we have

$$\begin{aligned}
&\frac{\mathrm{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h))) - \mathrm{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))}{\mathrm{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))} \sum_{j=0}^{2N+1} \frac{a_j \overline{F}_\alpha(a_j|a^-_{2N+2},(w,q_h))}{\mathrm{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))} p_j \\
\leq\ & \left(\frac{\overline{r}_\alpha(w,q_h)\overline{G}_{\alpha,\overline{r}_\alpha(w,q_h)}(\overline{r}_\alpha(w,q_h)|(0,1),(w,q_h)) - a_{2N+1}\overline{G}_{\alpha,a_{2N+1}}(a_{2N+1}|(0,1),(w,q_h))}{a_{2N+1}\overline{G}_{\alpha,a_{2N+1}}(a_{2N+1}|(0,1),(w,q_h))} \right) \sum_{j=0}^{2N+1} p_j \\
\leq\ & \left(\frac{\overline{r}_\alpha(w,q_h)\overline{G}_{\alpha,\overline{r}_\alpha(w,q_h)}(\overline{r}_\alpha(w,q_h)|(0,1),(w,q_h)) - a_{2N+1}\overline{G}_{\alpha,a_{2N+1}}(a_{2N+1}|(0,1),(w,q_h))}{a_{2N+1}\overline{G}_{\alpha,a_i}(a_i|(0,1),(w,q_h))} \right) \\
\leq\ & \frac{g_{\alpha,q_h}(\overline{r}_\alpha(w,q_h)) - g_{\alpha,q_h}(a_{2N+1})}{g_{\alpha,q_h}(a_{2N+1})} \leq \frac{\overline{r}_\alpha(w,q_h) - a_{2N+1}}{g_{\alpha,q_h}(w)} = \frac{\overline{r}_\alpha(w,q_h) - w}{Nwq},
\end{aligned}$$

where the first inequality follows from the definition of opt, the second from the fact that ı belongs to the simplex, and the fourth from the fact that the derivative $g'_{\alpha,q_h}(\cdot)$ is bounded (established in the previous case).

In the case where $\alpha = 0$, we have

$$\begin{aligned}
&\frac{\mathrm{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h))) - \mathrm{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))}{\mathrm{opt}(F_\alpha(\cdot|a_{2N+1},(w,q_h)))} \sum_{j=0}^{2N+1} \frac{a_j \overline{F}_\alpha(a_j|a^-_{2N+2},(w,q_h))}{\mathrm{opt}(F_\alpha(\cdot|a^-_{2N+2},(w,q_h)))} p_j \\
\leq\ & \frac{\lim_{x\to\infty}\mathrm{opt}(F_0(\cdot|x,(w,q_h))) - \mathrm{opt}(F_0(\cdot|b,(w,q_h)))}{\mathrm{opt}(F_0(\cdot|b,(w,q_h)))} \sum_{j=0}^{2N+1} p_j \\
\leq\ & \frac{\frac{1}{\frac{1}{q_h}-1} - \frac{b}{1+\left(\frac{1}{q_h}-1\right)b}}{\frac{b}{1+\left(\frac{1}{q_h}-1\right)b}} = \frac{1+\left(\frac{1}{q_h}-1\right)b}{\left(\frac{1}{q_h}-1\right)b} - 1 = \frac{1}{\left(\frac{1}{q_h}-1\right)b},
\end{aligned}$$

where the first inequality follows from the definition of opt, the second from the fact that **p** belongs to the simplex, and the definition of $\mathrm{opt}(F_\alpha(\cdot|a^-_{2N+1},(w,q_h)))$. Hence, we have, in this case

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \leq \frac{q_h}{(1-q_h)b}.$$

We are now in a position to combine all cases and conclude.

If $\alpha \in (0,1]$, we have established that

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;\leq\; \max\left\{\frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)\left[1+w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}\right], \frac{\eta w}{w-\eta}, \frac{\overline{r}_\alpha(w,q_h)-w}{Nwq}\right\}.$$

Recall that Proposition 4.F-2 implies that

$$\mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w,[q_l,q_h])) \;\leq\; \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h])) \;\leq\; \mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w,[q_l,q_h])) + \frac{\Delta(\mathbb{A})}{\underline{r}_\alpha(w,q_l)}.$$

Noting that $\Delta(\mathbb{A}) = \max\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N}, \eta, \frac{\overline{r}_\alpha(w,q_h)-w}{N}\}$, we have

$$\begin{aligned}
\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} &\leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h])) \\
&\leq \mathcal{R}(\mathcal{P}_{\mathbb{A}}, \mathcal{F}_\alpha(w,[q_l,q_h])) + \frac{\Delta(\mathbb{A})}{\underline{r}_\alpha(w,q_l)} \\
&\leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \left(\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}\right) + \max\left\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N\underline{r}_\alpha(w,q_l)}, \eta, \frac{\overline{r}_\alpha(w,q_h)-w}{N\underline{r}_\alpha(w,q_l)}\right\} \\
&\leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \max\left\{\frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)\left[1+w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}\right], \frac{\eta w}{w-\eta}, \frac{\overline{r}_\alpha(w,q_h)-w}{Nwq}\right\} \\
&\quad + \max\left\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N\underline{r}_\alpha(w,q_l)}, \eta, \frac{\overline{r}_\alpha(w,q_h)-w}{N\underline{r}_\alpha(w,q_l)}\right\}.
\end{aligned}$$

By choosing $\eta = \frac{w}{\sqrt{N}}$, we obtain

$$\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h])) \leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \mathcal{O}(\frac{1}{\sqrt{N}}).$$

Suppose now $\alpha = 0$. In this case.

$$\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \;\leq\; \max\left\{\frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)\left[1+w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}\right], \frac{\eta w}{w-\eta}, \frac{b-w}{Nwq}, \frac{q_h}{(1-q_h)b}\right\}.$$

Using again Proposition 4.F-2 and the fact that $\Delta(\mathbb{A}) = \max\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N}, \eta, \frac{b-w}{N}\}$, we have

$$\begin{aligned}
\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} &\leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h])) \\
&\leq \mathcal{R}(\mathcal{P}_\mathbb{A}, \mathcal{F}_\alpha(w,[q_l,q_h])) + \frac{\Delta(\mathbb{A})}{\underline{r}_\alpha(w,q_l)} \\
&\leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \left(\overline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} - \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}}\right) + \max\left\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N\underline{r}_\alpha(w,q_l)}, \eta, \frac{b-w}{N\underline{r}_\alpha(w,q_l)}\right\} \\
&\leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \max\left\{\frac{1}{N}\left(\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{\underline{r}_\alpha(w,q_l)}\right)\left[1 + w\frac{\Gamma_\alpha^{-1}(q_l)}{w-\eta}\right], \frac{\eta w}{w-\eta}, \frac{b-w}{Nwq}, \frac{q_h}{(1-q_h)b}\right\} \\
&\quad + \max\left\{\frac{w-\eta-\underline{r}_\alpha(w,q_l)}{N\underline{r}_\alpha(w,q_l)}, \eta, \frac{b-w}{N\underline{r}_\alpha(w,q_l)}\right\}.
\end{aligned}$$

By choosing $\eta = \frac{w}{\sqrt{N}}$ and $b = w\sqrt{N}$, we obatin

$$\underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} \leq \mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h])) \leq \underline{\mathcal{L}}_{\alpha,q_l,q_h,\mathbb{A}} + \mathcal{O}(\frac{1}{\sqrt{N}}).$$

This concludes the proof. □

Proofs of auxiliary results

Proof of Proposition 4.F-1. First we show that

$$\inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F) = \inf_{q\in[q_l,q_h]} \inf_{F \text{ in } \mathcal{F}_\alpha(w,q)} R(\Psi,F)$$

Let $q \in [q_l, q_h]$ and $F \in \mathcal{F}_\alpha(w,q)$, we have:

$$\begin{aligned}
R(\Psi,F) &\geq \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F) \\
\implies \inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) &\geq \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F) \\
\implies \inf_{q\in[q_l,q_h]} \inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) &\geq \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F).
\end{aligned}$$

Let $\epsilon > 0$ and $F_\epsilon \in \mathcal{F}_\alpha(w, [q_l, q_h])$ such that:

$$\begin{aligned}
& \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \geq R(\Psi, F_\epsilon) - \epsilon \\
\Longrightarrow\ & \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \geq \inf_{F\in\mathcal{F}_\alpha(w,\{\overline{F}_\epsilon(w)\})} R(\Psi, F) - \epsilon \\
\Longrightarrow\ & \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \geq \inf_{q\in[q_l,q_h]} \inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi, F) - \epsilon,
\end{aligned}$$

by taking $\epsilon \to 0$, we obtain the desired result.

Let $F \in \mathcal{F}_\alpha(w, [q_l, q_h])$ and $q = \overline{F}(w) \in [q_l, q_h]$, by Theorem 4.1, we have:

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi, F) &= \min\Bigg\{ \inf_{x\in[\underline{r}_\alpha(w,q),w)} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q))d\Psi(u), \\
&\qquad \inf_{x\in[w,\overline{r}_\alpha(w,q)]} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q)))d\Psi(u) \Bigg\} \\
&= \min\Bigg\{ \inf_{x\in[\underline{r}_\alpha(w,q),w)} \frac{1}{x}\left[\int_{[0,x)} ud\Psi(u) + \int_{[x,w]} u\overline{G}_{\alpha,w}(u|(x,1),(w,q))d\Psi(u)\right], \\
&\qquad \inf_{x\in[w,\overline{r}_\alpha(w,q)]} \frac{1}{x\overline{G}_{\alpha,x}(x|(0,1),(w,q))} \int_{[0,x]} u\overline{G}_{\alpha,x}(u|(0,1),(w,q))d\Psi(u) \Bigg\}.
\end{aligned}$$

Using the non-decreasing monotonicity of the functions $q \to \underline{r}_\alpha(w,q) = \frac{w}{\Gamma_\alpha^{-1}(q)+1}$, $q \to \overline{r}_\alpha(w,q) = \frac{w}{\alpha\Gamma_\alpha^{-1}(q)}$, we have:

$$\begin{aligned}
\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi, F) &\geq \min\Bigg\{ \inf_{x\in[\underline{r}_\alpha(w,q_l),w)} \frac{1}{x}\left[\int_{[0,x)} ud\Psi(u) + \int_{[x,w]} u\overline{G}_{\alpha,w}(u|(x,1),(w,q))d\Psi(u)\right], \\
&\qquad \inf_{x\in[w,\overline{r}_\alpha(w,q_h)]} \frac{1}{x\overline{G}_{\alpha,x}(x|(0,1),(w,q))} \int_{[0,x]} u\overline{G}_{\alpha,x}(u|(0,1),(w,q))d\Psi(u) \Bigg\}.
\end{aligned}$$

We have, for fixed (u, x) such that $x \in [\underline{r}_\alpha(w, q_l), w), u \in [x, w)$, the following function is clearly non-decreasing

$$q \to \overline{G}_{\alpha,w}(u|(x,1),(w,q)) = \Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{u-x}{w-x}\right).$$

We have, for fixed (u,x) such that $x \in [w, \overline{r}_\alpha(w,q_h)], u \in [w,x]$, the following function is non-increasing

$$q \to \frac{u\overline{G}_{\alpha,x}(u|(0,1),(w,q))}{x\overline{G}_{\alpha,x}(x|(0,1),(w,q))} = \frac{u\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{u}{w}\right)}{x\Gamma_\alpha\left(\Gamma_\alpha^{-1}(q)\,\frac{x}{w}\right)} = \begin{cases} \frac{u}{x}\left(\frac{u}{x} + \frac{x-u}{x(1+(1-\alpha)\Gamma_\alpha^{-1}(q)\frac{x}{w})}\right)^{\frac{1}{\alpha-1}} & \text{if } \alpha \in [0,1) \\ \frac{u}{x} q^{\frac{u-x}{w}} & \text{if } \alpha = 1. \end{cases}$$

Using the monotonicity of the above functions we get :

$$\begin{aligned} \inf_{F \text{ in } \mathcal{F}_\alpha(w,q)} R(\Psi,F) \;\geq\; \min\Bigg\{ & \inf_{x\in[\underline{r}_\alpha(w,q_l),w)} \frac{1}{x}\left[\int_{[0,x)} u d\Psi(u) + \int_{[x,w]} u\overline{G}_{\alpha,w}(u|(x,1),(w,q_l))d\Psi(u)\right], \\ & \inf_{x\in[w,\overline{r}_\alpha(w,q_h)]} \frac{1}{x\overline{G}_{\alpha,x}(x|(0,1),(w,q))} \int_{[0,x]} u\overline{G}_{\alpha,x}(u|(0,1),(w,q_h))d\Psi(u)\Bigg\}. \end{aligned}$$

Since the right hand-side does not depend on q, we take the minimum on q

$$\begin{aligned} \inf_{F\in\mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi,F) \;\geq\; & \min\Bigg\{ \inf_{x\in[\underline{r}_\alpha(w,q_l),w)} \frac{1}{x}\left[\int_{[0,x)]} u d\Psi(u) + \int_{[x,w]} u\overline{G}_{\alpha,w}(u|(x,1),(w,q_l))d\Psi(u)\right], \\ & \inf_{x\in[w,\overline{r}_\alpha(w,q_h)]} \frac{1}{x\overline{G}_{\alpha,x}(x|(0,1),(w,q))} \int_{[0,x]} u\overline{G}_{\alpha,x}(u|(0,1),(w,q_h))d\Psi(u)\Bigg\} \\ = \;& \min\Bigg\{ \inf_{x\in[\underline{r}_\alpha(w,q_l),w)} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_l))d\Psi(u), \\ & \inf_{x\in[w,\overline{r}_\alpha(w,q_h)]} \frac{1}{\mathrm{opt}(F_\alpha(\cdot|x,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_h)))d\Psi(u)\Bigg\}. \end{aligned}$$

This concludes the proof. □

Proof of Proposition 4.F-2. Fix q in $[q_l,q_h]$. Then, using Proposition 4.2, there exists $\Psi_\mathbb{A}$ in $\mathcal{P}_\mathbb{A}$ such that

$$\inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi_\mathbb{A},F) \geq \inf_{F\in\mathcal{F}_\alpha(w,q)} R(\Psi,F) - \frac{\Delta(\mathbb{A})}{a_1} - \frac{1}{q(1+(q^{-1}-1)a_N/w)}\mathbf{1}\{a_N < \overline{r}_\alpha(w,q)\},$$

where $\Delta(\mathbb{A}) = \sup_i\{a_i - a_{i-1}\}$. We have the following function

$$q \to -\frac{1}{q(1+(q^{-1}-1)a_N/w)} = -\frac{1}{\frac{a_N}{w}+q(1-\frac{a_N}{w})},$$

is non-increasing because $a_N > w$ therefore

$$-\frac{1}{q(1+(q^{-1}-1)a_N/w)} \geq -\frac{1}{q_h(1+(q_h^{-1}-1)a_N/w)},$$

moreover we have $q \to \overline{r}_\alpha(w,q) = \frac{w}{\alpha\Gamma_\alpha^{-1}(q)}$ is non-decreasing, therefore

$$\mathbf{1}\{a_N < \overline{r}_\alpha(w,q)\} \leq \mathbf{1}\{a_N < \overline{r}_\alpha(w,q_h)\}.$$

Hence, since $-\frac{1}{q(1+(q^{-1}-1)a_N/w)} < 0$, we have

$$-\frac{\mathbf{1}\{a_N < \overline{r}_\alpha(w,q)\}}{q(1+(q^{-1}-1)a_N/w)} \geq -\frac{\mathbf{1}\{a_N < \overline{r}_\alpha(w,q_h)\}}{q_h(1+(q_h^{-1}-1)a_N/w)}.$$

Using these lower bounds, we obtain

$$\inf_{F \text{ in } \mathcal{F}_\alpha(w,q)} R(\Psi_{\mathbb{A}},F) \geq \inf_{F \text{ in } \mathcal{F}_\alpha(w,q)} R(\Psi,F) - \frac{\Delta(\mathbb{A})}{\underline{r}_\alpha(w,q_l)} - \frac{\mathbf{1}\{a_N < \overline{r}_\alpha(w,q_h)\}}{q_h(1+(q_h^{-1}-1)a_N/w)},$$

taking the infinimum over q from both sides concludes the proof. □

4.G Upper bound linear program and implementation parameters

In this section, we show that one can obtain an upper bound on the maxmin ratio $\mathcal{R}(\mathcal{P}, \mathcal{F}_\alpha(w,[q_l,q_h]))$ by solving a linear program. Fix an arbitrary sequence of increasing reals $\mathbb{A} = \{a_i\}_{i=0}^{2N}$ such that $a_0 = 0, a_1 = \underline{r}_\alpha(w,q_l)$, $a_{N+1} = w$ and $a_{2N} \leq \overline{r}_\alpha(w,q_h)$. Set $a_{2N+1} := \overline{r}_\alpha(w,q_h)$.

Fix a mechanism Ψ in $\mathcal{P}$ and denote by $p_{j+1} = \int_{I_j} d\Psi(u)$ where we define the intervals $(I_j)_{j=0,\cdots,2N}$ as follows:

$$I_j = \begin{cases} [a_j, a_{j+1}) & \text{if } 0 \leq j < 2N, \\ [a_{2N}, a_{2N+1}] & \text{if } j = 2N. \end{cases}$$

Using Proposition 4.F-1, we have

$$\begin{aligned} & \inf_{F \in \mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \\ = \; & \min\Bigg\{ \inf_{x \in [\underline{r}_\alpha(w,q_l), w)} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|x,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_l))d\Psi(u), \\ & \inf_{x \in [w, \overline{r}_\alpha(w,q_h)]} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|x,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_h)))d\Psi(u) \Bigg\} \\ = \; & \min\Bigg\{ \min_{i=1,\cdots,N} \inf_{x \in [a_i, a_{i+1})} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|x,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_l))d\Psi(u), \\ & \min_{i=N+1,\cdots,2N} \inf_{x \in [a_i, a_{i+1}]} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|x,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|x,(w,q_h)))d\Psi(u) \Bigg\}. \end{aligned}$$

Following the same reasoning as in the proof of Theorem 4.4, we may also obtain an upper bound on $\inf_{F \in \mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F)$. Indeed, we have

$$\begin{aligned} \inf_{F \in \mathcal{F}_\alpha(w,[q_l,q_h])} R(\Psi, F) \; \leq \; & \min\Bigg\{ \min_{i=1,\cdots,N} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \int_0^\infty u\overline{F}_\alpha(u|a_{i+1}^-,(w,q_l))d\Psi(u), \\ & \min_{i=N+1,\cdots,2N} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_h)))} \int_0^\infty u\overline{F}_\alpha(u|a_{i+1}^-,(w,q_h))d\Psi(u) \Bigg\} \\ = \; & \min\Bigg\{ \min_{i=1,\cdots,N} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_l)))} \sum_{j=0}^{2N} \int_{I_j} u\overline{F}_\alpha(u|a_{i+1}^-,(w,q_l))d\Psi(u), \\ & \min_{i=N+1,\cdots,2N} \frac{1}{\operatorname{opt}(F_\alpha(\cdot|a_{i+1}^-,(w,q_h)))} \sum_{j=0}^{2N} \int_{I_j} u\overline{F}_\alpha(u|a_{i+1}^-,(w,q_h))d\Psi(u) \Bigg\}. \end{aligned}$$

For $x \in [\underline{r}_\alpha(w, q_l), w)$, the revenue function $u \mapsto u\overline{F}_\alpha(\cdot|x, (w, q_l))$ is increasing in u on $[0, x)$ and decreasing on (x, w). In addition, note that, for $x \in [w, \overline{r}_\alpha(w, q_h)]$, the revenue function $u \mapsto u\overline{F}_\alpha(u|x, (w, q_h))$ is non-decreasing on $[0, x]$ and $u\overline{F}_\alpha(u|x, (w, q_h)) = 0$ for $u > x$. Hence, we have

$$
\begin{aligned}
&\inf_{F \in \mathcal{F}_\alpha(w, [q_l, q_h])} R(\Psi, F) \\
\leq\ & \min\Bigg\{ \min_{i=1,\cdots,N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_l)))} \sum_{j=0}^{2N} \int_{I_j} u\overline{F}_\alpha(u|a^-_{i+1}, (w, q_l)) d\Psi(u), \\
& \min_{i=N+1,\cdots,2N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_h)))} \sum_{j=0}^{2N} \int_{I_j} u\overline{F}_\alpha(u|a^-_{i+1}, (w, q_h)) d\Psi(u) \Bigg\} \\
\leq\ & \min\Bigg\{ \min_{i=1,\cdots,N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_l)))} \Bigg[\sum_{j=0}^{i} a_{j+1}\overline{F}_\alpha(a_{j+1}|a^-_{i+1}, (w, q_l)) \int_{I_j} d\Psi(u) + \\
& \sum_{j=i+1}^{2N} a_j \overline{F}_\alpha(a_j|a^-_{i+1}, (w, q_l)) \int_{I_j} d\Psi(u) \Bigg], \\
& \min_{i=N+1,\cdots,2N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_h)))} \sum_{j=0}^{2N} a_{j+1}\overline{F}_\alpha(a_{j+1}|a^-_{i+1}, (w, q_h)) \int_{I_j} d\Psi(u) \Bigg\} \\
=\ & \min\Bigg\{ \min_{i=1,\cdots,N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_l)))} \Bigg[\sum_{j=1}^{i+1} a_j \overline{F}_\alpha(a_j|a^-_{i+1}, (w, q_l)) p_j + \\
& \sum_{j=i+2}^{2N+1} a_{j-1}\overline{F}_\alpha(a_{j-1}|a^-_{i+1}, (w, q_l)) p_j \Bigg], \\
& \min_{i=N+1,\cdots,2N} \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1}, (w, q_h)))} \sum_{j=1}^{2N+1} a_j \overline{F}_\alpha(a_j|a^-_{i+1}, (w, q_h)) p_j \Bigg\}.
\end{aligned}
$$

The problem of maximizing over mechanisms in $\mathcal{P}$ is clearly upper bounded by the problem of maximizing the RHS above over $p_1, ..., p_{2N+1}$. The epigraph formulation of the latter problem can

be written as

$$\begin{aligned}
\overline{\mathcal{L}_{U\alpha,q_l,q_h,\mathbb{A}}} \;=\; & \max_{\mathbf{p},c} \; c \qquad (4.G\text{-}1)\\
& s.t. \;\; \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1},(w,q_l)))}\left[\sum_{j=1}^{i+1} a_j \overline{F}_\alpha(a_j|a^-_{i+1},(w,q_l))p_j + \sum_{j=i+2}^{2N+1} a_{j-1}\overline{F}_\alpha(a_{j-1}|a^-_{i+1},(w,q_l))p_j\right] \geq \\
& \quad i = 1,...N,\\
& \quad \frac{1}{\text{opt}(F_\alpha(\cdot|a^-_{i+1},(w,q_h)))}\sum_{j=1}^{2N+1} a_j \overline{F}_\alpha(a_j|a^-_{i+1},(w,q_h))p_j \geq c \quad i = N+1,...2N,\\
& \quad \sum_{j=1}^{2N+1} p_j \leq 1, \quad p_i \geq 0 \quad i = 1,...2N+1.
\end{aligned}$$

Therefore we obtain that:

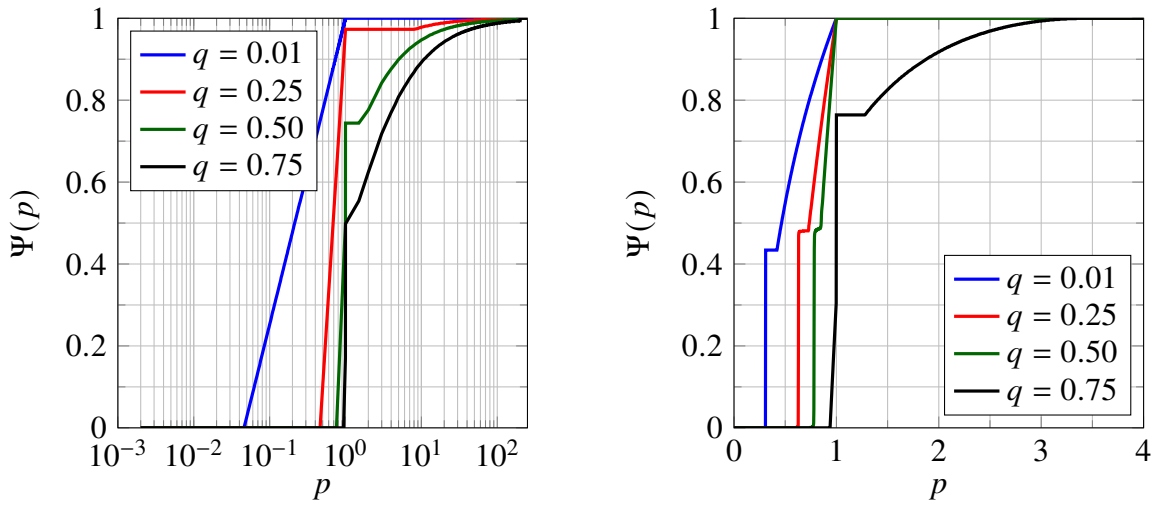
$$\mathcal{R}(\mathcal{P},\mathcal{F}_\alpha(w,[q_l,q_h])) \leq \overline{\mathcal{L}_{U\alpha,q_l,q_h,\mathbb{A}}}.$$

Implementation parameters: For all reported values in the main text, we use the following sequence in the Linear Programs

$$a_i \;=\; \begin{cases} \underline{r}_\alpha(1,q_l) + \frac{i}{N}\left(1-\eta-\underline{r}_\alpha(1,q_l)\right) & \text{if } 0 \leq i \leq N.\\ w + \frac{i-N-1}{N}\left(\min(b,\overline{r}_\alpha(1,q_h)) - 1\right) & \text{if } N+1 \leq i \leq 2N+1, \end{cases}$$

with $N = 2500, \eta = 10^{-5}, b = 250$.

4.H Additional Illustrations of near optimal mechanisms for Section 4.5



regular distributions **mhr distributions**

Figure 4.9: Illustration of near optimal mechanisms: The figure depicts near optimal pricing distributions for $w = 1$, q in $\{0.01, 0.25, 0.5, 0.75\}$. The left panel corresponds to regular distributions (plotted using a log scale) and the right panel to mhr distributions (on a regular scale).

References

[1] A. Bahamou, D. Goldfarb, and Y. Ren, *A mini-block fisher method for deep neural networks*, 2022.

[2] A. Bahamou and D. Goldfarb, "A dynamic sampling adaptive-sgd method for machine learning," *CoRR*, vol. abs/1912.13357, 2019. arXiv: 1912.13357.

[3] A. Allouah, A. Bahamou, and O. Besbes, "Optimal pricing with a single point," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, ser. EC '21, Budapest, Hungary: Association for Computing Machinery, 2021, p. 50, ISBN: 9781450385541.

[4] D. Goldfarb, Y. Ren, and A. Bahamou, "Practical quasi-newton methods for training deep neural networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 2386–2396.

[5] Y. Ren and D. Goldfarb, "Kronecker-factored quasi-newton methods for convolutional neural networks," *CoRR*, vol. abs/2102.06737, 2021. arXiv: 2102.06737.

[6] A. Allouah, A. Bahamou, and O. Besbes, "Pricing with samples," *Oper. Res.*, vol. 70, pp. 1088–1104, 2022.

[7] M. Ghasemizade and M. Schirmer, "Subsurface flow contribution in the hydrological cycle: Lessons learned and challenges ahead-a review," *Environmental Earth Sciences*, vol. 69, May 2013.

[8] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[9] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[10] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, 2012.

[11] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[12] R. M. Schmidt, F. Schneider, and P. Hennig, *Descending through a crowded valley - benchmarking deep learning optimizers*, 2021. arXiv: 2007.01547 [cs.LG].

[13] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, Dec. 1964.

[14] Y. Nesterov, "Introductory lectures on convex programming volume i: Basic course," *Lecture notes*, vol. 3, no. 4, p. 5, 1998.

[15] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.

[16] R. Fletcher, "A new approach to variable metric algorithms," *The computer journal*, vol. 13, no. 3, pp. 317–322, 1970.

[17] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Mathematics of computation*, vol. 24, no. 109, pp. 23–26, 1970.

[18] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of computation*, vol. 24, no. 111, pp. 647–656, 1970.

[19] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[20] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Classics in Applied Mathematics). Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1970, ISBN: 9780898719468.

[21] P. Xu, F. Roosta, and M. W. Mahoney, "Newton-type methods for non-convex optimization under inexact hessian information," *Mathematical Programming*, pp. 1–36, 2019.

[22] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-newton method for large-scale optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1008–1031, 2016.

[23] R. Gower, D. Goldfarb, and P. Richtárik, "Stochastic block bfgs: Squeezing more curvature out of data," in *International Conference on Machine Learning*, 2016, pp. 1869–1878.

[24] M. Wang, E. X. Fang, and B. Liu, "Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions," *Mathematical Programming*, vol. 161, no. 1-2, pp. 419–449, 2017.

[25] S.-I. Amari, H. Park, and K. Fukumizu, "Adaptive method of realizing natural gradient learning for multilayer perceptrons," *Neural computation*, vol. 12, no. 6, pp. 1399–1409, 2000.

[26] J. Martens, "Deep learning via hessian-free optimization.," in *ICML*, vol. 27, 2010, pp. 735–742.

[27] O. Vinyals and D. Povey, "Krylov subspace descent for deep learning," in *Artificial Intelligence and Statistics*, 2012, pp. 1261–1268.

[28] H. He, S. Zhao, Z. Tang, J. C. Ho, Y. Saad, and Y. Xi, "An efficient nonlinear acceleration method that exploits symmetry of the hessian," *arXiv preprint arXiv:2210.12573*, 2022.

[29] D. Scieur, E. Oyallon, A. d'Aspremont, and F. Bach, "Nonlinear acceleration of cnns," *arXiv preprint arXiv:1806.00370*, 2018.

[30] D. Scieur, L. Liu, T. Pumir, and N. Boumal, "Generalization of quasi-newton methods: Application to robust symmetric multisecant updates," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, A. Banerjee and K. Fukumizu, Eds., ser. Proceedings of Machine Learning Research, vol. 130, PMLR, 2021, pp. 550–558.

[31] Y. Ren and D. Goldfarb, "Efficient subsampled gauss-newton and natural gradient methods for training neural networks," *arXiv preprint arXiv:1906.02353*, 2019.

[32] N. Roux, P.-a. Manzagol, and Y. Bengio, "Topmoumoute online natural gradient algorithm," in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, Curran Associates, Inc., 2008.

[33] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *International conference on machine learning*, PMLR, 2015, pp. 2408–2417.

[34] R. Grosse and J. Martens, "A kronecker-factored approximate fisher matrix for convolution layers," in *International Conference on Machine Learning*, PMLR, 2016, pp. 573–582.

[35] T. Heskes, "On "natural" learning and pruning in multilayered perceptrons," *Neural Computation*, vol. 12, Jan. 2000.

[36] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[37] T. George, C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent, "Fast approximate natural gradient descent in a kronecker-factored eigenbasis," *arXiv preprint arXiv:1806.03884*, 2018.

[38] D. Goldfarb, Y. Ren, and A. Bahamou, "Practical quasi-newton methods for training deep neural networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 2386–2396.

[39] A. Botev, H. Ritter, and D. Barber, "Practical gauss-newton optimisation for deep learning," in *International Conference on Machine Learning*, PMLR, 2017, pp. 557–565.

[40] V. Gupta, T. Koren, and Y. Singer, "Shampoo: Preconditioned stochastic tensor optimization," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 1842–1850.

[41] Y. Ren and D. Goldfarb, "Tensor normal training for deep learning models," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[42] D. Bergemann and K. Schlag, "Pricing without priors," *Journal of the European Economic Association*, vol. 6, pp. 560–569, Apr. 2008.

[43] S. S. Eren and C. Maglaras, "Monopoly pricing with limited demand information," *Journal of Revenue and Pricing Management*, vol. 9, no. 1, pp. 23–48, 2010.

[44] M. C. Cohen, G. Perakis, and R. S. Pindyck, "A simple rule for pricing with limited knowledge of demand," *Management Science*, vol. 67, no. 3, pp. 1608–1621, 2021.

[45] W. Ma and D. Simchi-Levi, "Constructing demand curves from a single observation of bundle sales," in *Web and Internet Economics: 18th International Conference, WINE 2022, Troy, NY, USA, December 12–15, 2022, Proceedings*, Troy, NY, USA: Springer-Verlag, 2022, 150–166, ISBN: 978-3-031-22831-5.

[46] O. Besbes, A. N. Elmachtoub, and Y. Sun, "Pricing analytics for rotable spare parts," *INFORMS Journal on Applied Analytics*, vol. 50, no. 5, pp. 313–324, 2020. eprint: https://doi.org/10.1287/inte.2020.1033.

[47] Z. Huang, Y. Mansour, and T. Roughgarden, "Making the most of your samples," *SIAM Journal on Computing*, vol. 47, pp. 651–674, Jan. 2018.

[48] H. Fu, N. Immorlica, B. Lucier, and P. Strack, "Randomization beats second price as a prior-independent auction," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, ser. EC '15, Portland, Oregon, USA: Association for Computing Machinery, 2015, p. 323.

[49] M. Babaioff, Y. A. Gonczarowski, Y. Mansour, and S. Moran, "Are two (samples) really better than one?" In *Proceedings of the 2018 ACM Conference on Economics and Computation*, ser. EC '18, Ithaca, NY, USA: Association for Computing Machinery, 2018, p. 175, ISBN: 9781450358293.

[50] C. Daskalakis and M. Zampetakis, "More revenue from two samples via factor revealing sdps," in *Proceedings of the 21st ACM Conference on Economics and Computation*, ser. EC '20, Virtual Event, Hungary: Association for Computing Machinery, 2020, 257–272.

[51] A. Allouah, A. Bahamou, and O. Besbes, "Pricing with samples," *Operations Research*, vol. 70, no. 2, pp. 1088–1104, 2022.

[52] R. Kleinberg and T. Leighton, "The value of knowing a demand curve: Bounds on regret for online posted-price auctions," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '03, USA: IEEE Computer Society, 2003, p. 594, ISBN: 0769520405.

[53] O. Besbes and A. Zeevi, "Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms," *Operations Research*, vol. 57, pp. 1407–1420, Dec. 2009.

[54] J. Broder and P. Rusmevichientong, "Dynamic pricing under a general parametric choice model," *Operations Research*, vol. 60, pp. 965–980, Aug. 2012.

[55] N. B. Keskin and A. Zeevi, "Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies," *Operations Research*, vol. 62, pp. 1142–1167, Oct. 2014.

[56] J. Bu, D. Simchi-Levi, and Y. Xu, "Online pricing with offline data: Phase transition and inverse square law," *Management Science*, vol. 68, no. 12, pp. 8568–8588, 2022. eprint: https://doi.org/10.1287/mnsc.2022.4322.

[57] J. Martens, "New insights and perspectives on the natural gradient method," *Journal of Machine Learning Research*, vol. 21, no. 146, pp. 1–76, 2020.

[58] F. Kunstner, P. Hennig, and L. Balles, "Limitations of the empirical fisher approximation for natural gradient descent," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[59] V. Thomas, F. Pedregosa, B. van Merriënboer, P.-A. Manzagol, Y. Bengio, and N. L. Roux, "On the interplay between noise and curvature and its effect on optimization and generalization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., ser. Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 3503–3513.

[60] R. Collobert, "Large scale machine learning," Université de Paris VI, Tech. Rep., 2004.

[61] S. ichi Amari, R. Karakida, and M. Oizumi, *Fisher information and natural gradient learning of random deep networks*, 2018. arXiv: 1808.07172 [cs.LG].

[62] Y. Ollivier, *Riemannian metrics for neural networks i: Feedforward networks*, 2015. arXiv: 1303.0818 [cs.NE].

[63] R. Anil, V. Gupta, T. Koren, K. Regan, and Y. Singer, "Scalable second order optimization for deep learning," *arXiv preprint arXiv:2002.09018*, 2021. arXiv: 2002.09018 [cs.LG].

[64] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[65] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[66] G. Zhang, J. Martens, and R. Grosse, "Fast convergence of natural gradient descent for overparameterized neural networks," *arXiv preprint arXiv:1905.10961*, 2019.

[67] Z. Yao, A. Gholami, S. Shen, K. Keutzer, and M. W. Mahoney, "Adahessian: An adaptive second order optimizer for machine learning," *AAAI (Accepted)*, 2021.

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[69] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[70] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[71] Y. Ren and D. Goldfarb, "Kronecker-factored quasi-Newton methods for convolutional neural networks," *arXiv preprint arXiv:2102.06737*, 2021.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[73] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[75] G. Zhang, C. Wang, B. Xu, and R. Grosse, "Three mechanisms of weight decay regularization," in *International Conference on Learning Representations*, 2019.

[76] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," *arXiv preprint arXiv:1910.05446*, 2019.

[77] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[78] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[79] J. Chen, T. Ma, and C. Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," *arXiv preprint arXiv:1801.10247*, 2018.

[80] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[81] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[82] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl, *Measuring the effects of data parallelism on neural network training*, 2019. arXiv: 1811.03600 [cs.LG].

[83] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[84] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to SGD," *CoRR*, vol. abs/1712.07628, 2017. arXiv: 1712.07628.

[85] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, Dec. 1964.

[86] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[87] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[88] T. Tieleman and G. Hinton, *Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural Networks for Machine Learning, 2012.

[89] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *CoRR*, vol. abs/1212.5701, 2012.

[90] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[91] M. Mahsereci and P. Hennig, "Probabilistic line searches for stochastic optimization," *CoRR*, vol. abs/1502.02846, 2015. arXiv: 1502.02846.

[92] S. Vaswani, A. Mishkin, I. H. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien, "Painless stochastic gradient: Interpolation, line-search, and convergence rates," *CoRR*, vol. abs/1905.09997, 2019. arXiv: 1905.09997.

[93] C. Paquette and K. Scheinberg, "A stochastic line search method with expected complexity analysis," *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 349–376, 2020. eprint: https://doi.org/10.1137/18M1216250.

[94] N. Loizou, S. Vaswani, I. Hadj Laradji, and S. Lacoste-Julien, "Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, A. Banerjee and K. Fukumizu, Eds., ser. Proceedings of Machine Learning Research, vol. 130, PMLR, 2021, pp. 1306–1314.

[95] A. Orvieto, S. Lacoste-Julien, and N. Loizou, *Dynamics of sgd with stochastic polyak stepsizes: Truly adaptive variants and convergence to exact solution*, 2022.

[96] S. Li, W. J. Swartworth, M. Takáč, D. Needell, and R. M. Gower, *Sp2: A second order stochastic polyak method*, 2022.

[97] R. M. Gower, A. Defazio, and M. G. Rabbat, "Stochastic polyak stepsize with a moving target," *CoRR*, vol. abs/2106.11851, 2021. arXiv: 2106.11851.

[98] F. Schaipp, R. M. Gower, and M. Ulbrich, *A stochastic proximal polyak step size*, 2023.

[99] R. M. Gower, M. Blondel, N. Gazagnadou, and F. Pedregosa, *Cutting some slack for sgd with adaptive polyak stepsizes*, 2022.

[100] A. Botev, H. Ritter, and D. Barber, "Practical gauss-newton optimisation for deep learning," *ICML 2017*, vol. abs/1706.03662, 2017. arXiv: 1706.03662 [stat.ML].

[101] T. Sun and Q. Tran-Dinh, "Generalized self-concordant functions: A recipe for newton-type methods," *Mathematical Programming*, vol. 178, no. 1-2, pp. 145–213, 2019.

[102] Y. Nesterov and A. S. Nemirovsky, "Interior-point polynomial methods in convex programming," 1994.

[103] Y. Zhang and X. Lin, "Disco: Distributed optimization for self-concordant empirical loss," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and

D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, 2015, pp. 362–370.

[104] F. R. Bach, "Self-concordant analysis for logistic regression," *ArXiv*, vol. abs/0910.4627, 2009.

[105] Q. Tran-Dinh, Y.-H. Li, and V. Cevher, "Composite convex minimization involving self-concordant-like cost functions," in *MCO*, 2015.

[106] W. Gao and D. Goldfarb, "Quasi-newton methods: Superlinear convergence without line searches for self-concordant functions," *Optimization Methods and Software*, vol. 34, no. 1, pp. 194–217, 2019. eprint: https://doi.org/10.1080/10556788.2018.1510927.

[107] B. Ghorbani, S. Krishnan, and Y. Xiao, "An investigation into neural net optimization via hessian eigenvalue density," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, Long Beach, California, USA: PMLR, 2019, pp. 2232–2241.

[108] L. M. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč, "Sgd and hogwild! convergence without the bounded gradients assumption," *ICML 2018*, 2018. arXiv: 1802.03801 [math.OC].

[109] R. Leblond, F. Pedregosa, and S. Lacoste-Julien, "Improved asynchronous parallel optimization analysis for stochastic incremental methods," *Journal of Machine Learning Research*, vol. 19, no. 81, pp. 1–68, 2018.

[110] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[111] B. A. Pearlmutter, "Fast exact multiplication by the hessian," *Neural Comput.*, vol. 6, no. 1, pp. 147–160, Jan. 1994.

[112] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. arXiv: 1608.06993.

[113] M. Bagnoli and T. Bergstrom, "Log-concave probability and its applications," *Economic theory*, vol. 26, no. 2, pp. 445–469, 2005.

[114] C. Ewerhart, "Regular type distributions in mechanism design and ρ-concavity," *Economic Theory*, vol. 53, no. 3, pp. 591–603, 2013.

[115] P. Azar, C. Daskalakis, S. Micali, and S. M. Weinberg, "Optimal and efficient parametric auctions," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete*

Algorithms, ser. SODA '13, New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2013, 596–604, ISBN: 9781611972511.

[116] R. Caldentey, Y. Liu, and I. Lobel, "Intertemporal pricing under minimax regret," *Operations Research*, vol. 65, no. 1, 104–129, Feb. 2017.

[117] N. Chehrazi and T. A. Weber, "Monotone approximation of decision problems," *Operations Research*, vol. 58, no. 4-part-2, pp. 1158–1177, 2010. eprint: https://doi.org/10.1287/opre.1100.0814.

[118] N. Chen, A. A. Cire, M. Hu, and S. Lagzi, "Model-free assortment pricing with transaction data," *Management Science*, vol. 0, no. 0, null, 0. eprint: https://doi.org/10.1287/mnsc.2022.4651.

[119] P. D. Azar and S. Micali, "Parametric digital auctions," in *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ser. ITCS '13, Berkeley, California, USA: Association for Computing Machinery, 2013, 231–232, ISBN: 9781450318594.

[120] H. Fu, C. Liaw, and S. Randhawa, "The vickrey auction with a single duplicate bidder approximates the optimal revenue," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, ser. EC '19, Phoenix, AZ, USA: Association for Computing Machinery, 2019, 419–420, ISBN: 9781450367929.

[121] R. Cole and T. Roughgarden, "The sample complexity of revenue maximization," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC '14, New York, New York: Association for Computing Machinery, 2014, 243–252, ISBN: 9781450327107.

[122] M. Derakhshan, N. Golrezaei, and R. Paes Leme, "Linear program-based approximation for personalized reserve prices," *Management Science*, vol. 68, no. 3, pp. 1849–1864, 2022. eprint: https://doi.org/10.1287/mnsc.2020.3897.

[123] A. Allouah and O. Besbes, "Sample-based optimal pricing," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, ser. EC '19, Phoenix, AZ, USA: Association for Computing Machinery, 2019, p. 391.

[124] J. Riley and R. Zeckhauser, "Optimal selling strategies: When to haggle, when to hold firm," *Quarterly Journal of Economics*, vol. 98, no. 2, pp. 267–289, 1983.

[125] N. Schweizer and N. Szech, "The Quantitative View of Myerson Regularity," CESifo, CESifo Working Paper Series 5712, 2016.

[126] A. C. Yao, "Probabilistic computations: Toward a unified measure of complexity," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, 1977, pp. 222–227.