Algorithms and Learning for Fair Portfolio Design

Emily Diana, Travis Dick, Hadi Elzayn, Michael Kearns, Aaron Roth,
Zachary Schutzman, Saeed Sharifi-Malvajerdi, Juba Ziani

University of Pennsylvania

June 15, 2020

Abstract

We consider a variation on the classical finance problem of optimal portfolio design. In our setting, a large population of consumers is drawn from some distribution over *risk tolerances*, and each consumer must be assigned to a portfolio of lower risk than her tolerance. The consumers may also belong to underlying groups (for instance, of demographic properties or wealth), and the goal is to design a small number of portfolios that are fair across groups in a particular and natural technical sense.

Our main results are algorithms for optimal and near-optimal portfolio design for both social welfare and fairness objectives, both with and without assumptions on the underlying group structure. We describe an efficient algorithm based on an internal two-player zero-sum game that learns near-optimal fair portfolios *ex ante* and show experimentally that it can be used to obtain a small set of fair portfolios *ex post* as well. For the special but natural case in which group structure coincides with risk tolerances (which models the reality that wealthy consumers generally tolerate greater risk), we give an efficient and optimal fair algorithm. We also provide generalization guarantees for the underlying risk distribution that has no dependence on the number of portfolios and illustrate the theory with simulation results.

Contents

1 Introduction

In this work we consider algorithmic and learning problems in a model for the fair design of financial products. We imagine a large population of individual retail investors or *consumers*, each of whom has her own tolerance for investment risk in the form of a limit on the variance of returns. It is well known in quantitative finance that for any set of financial assets, the optimal expected returns on investment are increasing in risk.

A large retail investment firm (such as Vanguard or Fidelity) wishes to design portfolios to serve these consumers under the common practice of assigning consumers only to portfolios with lower risks than their tolerances. The firm would like to design and offer only a *small* number of such products — much smaller than the number of consumers — since the execution and maintenance of portfolios is costly and ongoing. The overarching goal is to design the products to minimize consumer *regret* — the loss of returns due to being assigned to lower-risk portfolios compared to the bespoke portfolios saturating tolerances — both with and without fairness considerations.

We consider a notion of group fairness adapted from the literature on fair division. Consumers belong to underlying groups that may be defined by standard demographic features such as race, gender or age, or they may be defined by the risk tolerances themselves, as higher risk appetite is generally correlated with higher wealth. We study *minmax group fairness*, in which the goal is to minimize the maximum regret across groups — i.e. to optimize for the *least well-off* group. Compared to the approach of constraining regret to be equal across groups (which is not even always feasible in our setting), minmax optimal solutions have the property that they Pareto-dominate regret-equalizing solutions: every group has regret that can only be smaller than it would be if regret were constrained to be equalized across groups.

Related Work. Our work generally falls within the literature on fairness in machine learning, which is too broad to survey in detail here; see [3] for a recent overview. We are perhaps closer in spirit to research in fair division or allocation problems [1, 2, 13], in which a limited resource must be distributed across a collection of players so as to maximize the utility of the *least* well-off; here, the resource in question is the small number of portfolios to design. However, we are not aware of any technical connections between our work and this line of research. Within the fairness in machine learning literature, our work is closest to *fair facility location* problems [9, 10], which attempt to choose a small number of "centers" to serve a large and diverse population and "fair allocation" problems that arise in the context of predictive policing [4–6].

There seems to have been relatively little explicit consideration of fairness issues in quantitative finance generally and optimal portfolio design specifically. An exception is [8], in which the interest is in fairly amortizing transaction costs of a single portfolio across investors rather than designing multiple portfolios to meet a fairness criterion.

Our Results and Techniques. In Section 3, we provide a dynamic program to find p products that optimize the average regret of a single population. In Section 4, we divide the population into different groups and develop techniques to guarantee minmax group fairness: in Section 4.1, we show a separation between deterministic solutions and randomized solutions (i.e. distributions over sets of p products) for minmax group fairness; in Section 4.2, we leverage techniques for learning in games to find a distribution over products that optimizes for minmax group fairness; in Section 4.3, we focus on deterministic solutions and extend our dynamic programming approach to efficiently optimize for the minmax objective when the number of groups is constant; in Section 4.4, we study the natural special case in which groups are defined by disjoint intervals on the real line and give algorithms that are efficient even for large numbers of groups. In Section 5, we show that when consumers' risk tolerances are drawn i.i.d. from an unknown distribution, empirical risk bounds from a sample of consumers generalize to the underlying distribution, and we prove tight sample complexity bounds. Finally, in Section 6, we provide experiments to complement our theoretical results.

2 Model and Preliminaries

We aim to create products (portfolios) consisting of weighted collections of assets with differing means and standard deviations (risks). Each consumer is associated with a real number $\tau \in \mathbb{R}_{\geq 0}$, which is the *risk tolerance* of the consumer — an upper bound on the standard deviation of returns. We assume that consumers' risk tolerances are bounded.

2.1 Bespoke Problem

We adopt the standard Markowitz framework [11] for portfolio design. Given a set of m assets with mean $\mu \in \mathbb{R}^m_+$ and covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$, and a consumer risk tolerance τ, the *bespoke* portfolio achieves the maximum expected return that the consumer can realize by assigning weights $\mathbf{a}$ over the assets (where the weight of an asset represents the fraction of the portfolio allocated to said asset) subject to the constraint that the overall risk – quantified as the standard deviation of the mixture $\mathbf{a}$ over assets – does not exceed her tolerance τ. Finding a consumer's bespoke portfolio can be written down as an optimization problem. We formalize the bespoke problem in Equation (1) below, and call the solution $r(\tau)$.

$$r(\tau) = \max_{\mathbf{a} \in \mathbb{R}^m} \left\{ \mathbf{a}^\top \mu \,|\, \mathbf{a}^\top \Sigma\, \mathbf{a} \leq \tau^2, \mathbb{1}^\top \mathbf{a} = 1 \right\} \tag{1}$$

Here we note that optimal portfolios are summarized by function $r(\tau)$, which is non-decreasing. Since consumer tolerances are bounded, we let B denote the maximum value of $r(\tau)$ across all consumers.

2.2 A Regret Notion

Suppose there are n consumers to whom we want to offer products. Our goal is to design $p \ll n$ products that minimize a notion of *regret* for a given set of consumers. A product has a risk (standard deviation) which we will denote by c. We assume throughout that in addition to the selected p products, there is always a risk-free product (say cash) available that has zero return; we will denote this risk-free product by $c_0 \equiv 0$ throughout the paper ($r(c_0) = 0$). For a given consumer with risk threshold τ, the regret of the consumer with respect to a set of products $\mathbf{c} = (c_1, c_2, \ldots, c_p) \in \mathbb{R}^p_{\geq 0}$ is the difference between the return of her bespoke product and the maximum return of any product with risk that is *less than or equal to* her risk threshold. To formalize this, the regret of products $\mathbf{c}$ for a consumer with risk threshold τ is defined as $\mathrm{R}_\tau(\mathbf{c}) = r(\tau) - \max_{c_j \leq \tau} r(c_j)$. Note since $c_0 = 0$ always exists, the $\max_{c_j \leq \tau} r(c_j)$ term is well defined. Now for a given set of consumers $S = \{\tau_i\}_{i=1}^n$, the regret of products $\mathbf{c}$ on S is simply defined as the *average regret* of $\mathbf{c}$ on S:

$$\mathrm{R}_S(\mathbf{c}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathrm{R}_{\tau_i}(\mathbf{c}) \tag{2}$$

When S includes the entire population of consumers, we call $\mathrm{R}_S(\mathbf{c})$ the *population regret*. The following notion for the *weighted regret* of $\mathbf{c}$ on S, given a vector $\mathbf{w} = (w_1, \ldots, w_n)$ of weights for each consumer, will be useful in Sections 3 and 4:

$$\mathrm{R}_S(\mathbf{c}, \mathbf{w}) \triangleq \sum_{i=1}^{n} w_i\, \mathrm{R}_{\tau_i}(\mathbf{c}) \tag{3}$$

Absent any fairness concern, our goal is to design efficient algorithms to minimize $\mathrm{R}_S(\mathbf{c})$ for a given set of consumers S and target number of products p: $\min_{\mathbf{c}} \mathrm{R}_S(\mathbf{c})$. This will be the subject of Section 3. We can always find an optimal set of products as a subset of the n consumer risk thresholds $S = \{\tau_i\}_i$, because if any product c_j is not in S, we can replace it by $\min\{\tau_i \mid \tau_i \geq c_j\}$ without decreasing the return for any consumer[1]. We let $C_p(S)$ represent the set of all subsets of size p for a given set of consumers

[1] We consider a more general regret framework in Appendix B which allows consumers to be assigned to products with risk higher than their tolerance, and in this case it is not necessarily optimal to always place products on the consumer risk thresholds.

S: $C_p(S) = \{\mathbf{c} = (c_1, c_2, \ldots, c_p) \subseteq S\}$. We therefore can reduce our regret minimization problem to the following problem:

$$\mathcal{R}(S, p) \triangleq \min_{\mathbf{c} \in C_p(S)} \mathrm{R}_S(\mathbf{c}) \tag{4}$$

Similarly, we can reduce the weighted regret minimization problem to the following problem:

$$\mathcal{R}(S, \mathbf{w}, p) \triangleq \min_{\mathbf{c} \in C_p(S)} \mathrm{R}_S(\mathbf{c}, \mathbf{w}). \tag{5}$$

2.3 Group Fairness: *ex post* and *ex ante*

Now suppose consumers are partitioned into g groups: $S = \{G_k\}_{k=1}^{g}$, e.g. based on their attributes like race or risk levels. Each G_k consists of the consumers of group k represented by their risk thresholds. We will often abuse notation and write $i \in G_k$ to denote that consumer i has threshold $\tau_i \in G_k$. Given this group structure, minimizing the regret of the whole population absent any constraint might lead to some groups incurring much higher regret than others. With fairness concerns in mind, we turn to the design of efficient algorithms to minimize the maximum regret over groups (we call this maximum "group regret"):

$$\mathcal{R}_{\text{fair}}(S, p) \triangleq \min_{\mathbf{c} \in C_p(S)} \left\{ \max_{1 \le k \le g} \mathrm{R}_{G_k}(\mathbf{c}) \right\} \tag{6}$$

The set of products $\mathbf{c}$ that solves the above minmax problem[2] will be said to satisfy *ex post* minmax fairness (for brevity, we call this "fairness" throughout). One can relax problem (6) by allowing the designer to *randomize* over sets of p products and output a *distribution* over $C_p(S)$ (as opposed to one deterministic set of products) that minimizes the maximum *expected* regret of groups:

$$\widehat{\mathcal{R}}_{\text{fair}}(S, p) \triangleq \min_{\mathcal{C} \in \Delta(C_p(S))} \left\{ \max_{1 \le k \le g} \mathbb{E}_{\mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] \right\} \tag{7}$$

where $\Delta(A)$ represents the set of probability distributions over the set A, for any A. The distribution $\mathcal{C}$ that solves the above minmax problem will be said to satisfy *ex ante* minmax fairness — meaning fairness is satisfied in expectation *before* realizing any set of products drawn from the distribution $\mathcal{C}$ — but there is no fairness guarantee on the realized draw from $\mathcal{C}$. Such a notion of fairness is useful in settings in which the designer has to make repeated decisions over time and has the flexibility to offer different sets of products in different time steps. In Section 4, we provide algorithms that solve both problems cast in Equations (6) and (7). We note that while there is a simple integer linear program (ILP) that solves Equations (6) and (7), such an ILP is often intractable to solve. We use it in our experiments on small instances to evaluate the quality of our efficient algorithms.

3 Regret Minimization Absent Fairness

In this section we provide an efficient dynamic programming algorithm for finding the set of p products that minimizes the (weighted) regret for a collection of consumers. This dynamic program will be used as a subroutine in our algorithms for finding optimal products for minmax fairness.

Let $S = \{\tau_i\}_{i=1}^{n}$ be a collection of consumer risk thresholds and $\mathbf{w} = (w_i)_{i=1}^{n}$ be their weights, such that $\tau_1 \le \cdots \le \tau_n$ (w.l.o.g). The key idea is as follows: suppose that consumer index z defines the riskiest product in an optimal set of p products. Then all consumers $z, \ldots, n$ will be assigned to that product and the consumers $1, \ldots, z-1$ will not be. Therefore, if we knew the highest risk product in an optimal solution, we would be left with a smaller sub-problem in which the goal is to optimally choose $p-1$ products for the first $z-1$ consumers. Our dynamic programming algorithm finds the optimal p' products for the first n' consumers for all values of $n' \le n$ and $p' \le p$.

[2] In Appendix C, we show that optimizing for population regret can lead to arbitrarily bad group regret in relative terms, and vice-versa.

More formally for any $n' \leq n$, let $S[n'] = \{\tau_i\}_{i=1}^{n'}$ and $\mathbf{w}[n']$ denote the n' lowest risk consumers and their weights. For any $n' \leq n$ and $p' \leq p$, let $T(n', p') = \mathcal{R}(S[n'], \mathbf{w}[n'], p')$ be the optimal weighted regret achievable in the sub-problem using p' products for the first n' weighted consumers. We make use of the following recurrence relations:

Lemma 1. *The function T defined above satisfies the following properties:*

1. *For any $1 \leq n' \leq n$, we have $T(n', 0) = \sum_{i=1}^{n'} w_i r(\tau_i)$.*
2. *For any $1 \leq n' \leq n$ and $0 \leq p' \leq p$, we have*

$$T(n', p') = \min_{z \in \{p', \ldots, n'\}} \left(T(z-1, p'-1) + \sum_{i=z}^{n'} w_i \big(r(\tau_i) - r(\tau_z)\big) \right).$$

Proof. See Appendix G.1. □

The running time of our dynamic programming algorithm, which uses the above recurrence relations to solve all sub-problems, is summarized below.

Theorem 1. *There exists an algorithm that, given a collection of consumers $S = \{\tau_i\}_{i=1}^{n}$ with weights $\mathbf{w} = (w_i)_{i=1}^{n}$ and a target number of products p, outputs a collection of products $\mathbf{c} \in C_p(S)$ with minimal weighted regret: $\mathrm{R}_S(\mathbf{c}, \mathbf{w}) = \mathcal{R}(S, \mathbf{w}, p)$. This algorithm runs in time $O(n^2 p)$.*

Proof. The algorithm computes a table containing the values $T(n', p')$ for all values of $n' \leq n$ and $p' \leq p$ using the above recurrence relations. The first column, when $p' = 0$, is computed using property 1 from Lemma 1, while the remaining columns are filled using property 2. By keeping track of the value of the index z achieving the minimum in each application of property 2, we can also reconstruct the optimal products for each sub-problem.

To bound the running time, observe that the sums appearing in both properties can be computed in $O(1)$ time, after pre-computing all partial sums of the form $\sum_{i=1}^{n'} w_i r(\tau_i)$ and $\sum_{i=1}^{n'} w_i$ for $n' \leq n$. Computing these partial sums takes $O(n)$ time. With this, property 1 can be evaluated in $O(1)$ time, and property 2 can be evaluated in $O(n)$ time (by looping over the values of z). In total, we can fill out all $O(np)$ table entries in $O(n^2 p)$ time. Reconstructing the optimal set of products takes $O(p)$ time. □

4 Regret Minimization with Group Fairness

In this section, we study the problem of choosing p products when the consumers can be partitioned into g groups, and we want to optimize minmax fairness across groups, for both the *ex post* minmax fairness Program (6) and the *ex ante* minmax fairness Program (7).

We start the discussion of minmax fairness by showing a separation between the *ex post* objective in Program (6) and the *ex ante* objective in Program (7). More precisely, we show in subsection 4.1 that the objective value of Program (6) can be $\Omega(g)$ times higher than the objective value of Program (6).

In the remainder of the section, we provide algorithms to solve Programs (6) and (7). In subsection 4.2, we provide an algorithm that solves Program (7) to any desired additive approximation factor via no-regret dynamics. In subsection 4.3, we provide a dynamic program approach that finds an approximately optimal solution to Program (6) when the number of groups g is small. Finally, in subsection 4.4, we provide a dynamic program that solves Program (6) exactly in a special case of our problem in which the groups are given by disjoint intervals of consumer risk thresholds.

4.1 Separation Between Randomized and Deterministic Solutions

The following theorem shows a separation between the minmax (expected) regret achievable by deterministic versus randomized strategies (as per Programs (6) and (7)); in particular, the regret $\mathcal{R}_{\text{fair}}$ of the best deterministic strategy can be $\Omega(g)$ times worse than the regret $\widehat{\mathcal{R}}_{\text{fair}}$ of the best randomized strategy:

Theorem 2. *For any g and p, there exists an instance S consisting of g groups such that*

$$\frac{\widehat{\mathcal{R}}_{fair}(S,p)}{\mathcal{R}_{fair}(S,p)} \leq \frac{1}{p+1}\left\lceil \frac{p+1}{g} \right\rceil$$

Proof. The proof is provided in Appendix G.2. □

In the following theorem, we show that for any instance of our problem, by allowing a multiplicative factor g blow-up in the target number of products p, the optimal deterministic minmax value will be at least as good as the randomized minmax value with p products.

Theorem 3. *We have that for any instance S consisting of g groups, and any p,*

$$\mathcal{R}_{fair}(S,gp) \leq \widehat{\mathcal{R}}_{fair}(S,p)$$

Proof. Fix any instance $S = \{G_k\}_{k=1}^{g}$ and any p. Let $\mathbf{c}_k^* \triangleq \operatorname{argmin}_{\mathbf{c}\in C_p(S)} \mathrm{R}_{G_k}(\mathbf{c})$ which is the best set of p products for group G_k. We have that

$$\begin{aligned}
\mathcal{R}_{\text{fair}}(S,gp) &= \min_{\mathbf{c}\in C_{gp}(S)} \max_{1\leq k\leq g} \mathrm{R}_{G_k}(\mathbf{c}) \\
&\leq \max_{1\leq k\leq g} \mathrm{R}_{G_k}(\cup_k \mathbf{c}_k^*) \\
&\leq \max_{1\leq k\leq g} \mathrm{R}_{G_k}(\mathbf{c}_k^*) \\
&\leq \max_{1\leq k\leq g} \mathbb{E}_{\mathbf{c}\sim\mathcal{C}}\left[\mathrm{R}_{G_k}(\mathbf{c})\right]
\end{aligned}$$

where the last inequality follows from the definition of $\mathbf{c}_k^*$ and it holds for any distribution $\mathcal{C} \in \Delta(C_p(S))$. □

4.2 An Algorithm to Optimize for *ex ante* Fairness

In this section, we provide an algorithm to solve the *ex ante* Program (7). Remember that the optimization problem is given by

$$\widehat{\mathcal{R}}_{\text{fair}}(S,p) \triangleq \min_{\mathcal{C}\in\Delta(C_p(S))} \left\{ \max_{1\leq k\leq g} \mathbb{E}_{\mathbf{c}\sim\mathcal{C}}\left[\mathrm{R}_{G_k}(\mathbf{c})\right] \right\} \tag{7}$$

Algorithm (1) relies on the dynamics introduced by Freund and Schapire [7] to solve Program (7). The algorithm interprets this minmax optimization problem as a zero-sum game between the designer, who wants to pick products to minimize regret, and an adversary, whose goal is to pick the highest regret group. This game is played repeatedly, and agents update their strategies at every time step based on the history of play. In our setting, the adversary uses the multiplicative weights algorithm to assign weights to groups (as per Freund and Schapire [7]) and the designer best-responds using the dynamic program from Section 3 to solve Equation (8) to choose an optimal set of products, noting that

$$\mathbb{E}_{k\sim D(t)}\left[\mathrm{R}_{G_k}(\mathbf{c})\right] = \sum_{k\in[g]} D_k(t) \sum_{i\in G_k} \frac{R_{\tau_i}(\mathbf{c})}{|G_k|} = \sum_{i=1}^{n} R_{\tau_i}(\mathbf{c}) \sum_{k\in[g]} \frac{D_k(t)\mathbb{I}\{i\in G_k\}}{|G_k|} = \mathrm{R}_S(\mathbf{c}, \mathbf{w}(t))$$

where $w_i(t) \triangleq \sum_{k\in[g]} \frac{D_k(t)}{|G_k|}\mathbb{I}\{i\in G_k\}$ denotes the weight assigned to agent i, at time step t.

Theorem 4 shows that the time-average of the strategy of the designer in Algorithm 1 is an approximate solution to minmax problem (7).

Algorithm 1 2-Player Dynamics for the Ex Ante Minmax Problem

Input: p target number of products, consumers S partitioned in groups $G_1, \ldots, G_g$, and T.

Initialization: The no-regret player picks the uniform distribution $D(1) = \left(\frac{1}{g}, \ldots, \frac{1}{g}\right) \in \Delta([g])$,

for $t = 1, \ldots, T$ **do**

The best-response player chooses $\mathbf{c}(t) = (c_1(t), \ldots, c_p(t)) \in C_p(S)$ so as to solve

$$\mathbf{c}(t) = \operatorname*{argmin}_{\mathbf{c} \in C_p(S)} \mathbb{E}_{k \sim D(t)} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right]. \tag{8}$$

The no-regret player observes $u_k(t) = \mathrm{R}_{G_k}(\mathbf{c}(t))/B$ for all $k \in [g]$. The no-regret player sets $D(t+1)$ via multiplicative weight update with $\beta = \frac{1}{1+\sqrt{2\frac{\ln g}{T}}} \in (0,1)$, as follows:

$$D_k(t+1) = \frac{D_k(t)\beta^{u_k(t)}}{\sum_{h=1}^{g} D_h(t)\beta^{u_h(t)}} \ \forall k \in [g],$$

Output: $\mathcal{C}_T$: the uniform distribution over $\{\mathbf{c}(t)\}_{t=1}^{T}$.

Theorem 4. *Suppose that for all $i \in [n]$, $r(\tau_i) \leq B$. Then for all $T > 0$, Algorithm 1 runs in time $O(Tn^2p)$ and the output distribution $\mathcal{C}_T$ satisfies*

$$\max_{k \in [g]} \mathbb{E}_{\mathbf{c} \sim \mathcal{C}_T} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] \leq \widehat{\mathcal{R}}_{fair}(S, p) + B \left(\sqrt{\frac{2 \ln g}{T}} + \frac{\ln g}{T} \right). \tag{9}$$

Proof. Note that the action space of the designer $C_p(S)$ and the action set of the adversary $\{G_k\}_{k=1}^{g}$ are both finite, so our zero-sum game can be written in normal form. Further, $u_k(t) \in [0,1]$, noting that the return of each agent is in $[0, B]$ — so must be the average return of a group. Therefore, our minmax game fits the framework of Freund and Schapire [7], and we have that

$$\max_{k \in [g]} \mathbb{E}_{\mathbf{c} \sim \mathcal{C}_T} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] \leq \min_{\mathcal{C} \in \Delta(C_p(S))} \max_{\mathcal{G} \in \Delta([g])} \mathbb{E}_{k \sim \mathcal{G},\, \mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] + B \left(\sqrt{\frac{2 \ln g}{T}} + \frac{\ln g}{T} \right).$$

The approximation statement is obtained by noting that for any distribution $\mathcal{G}$ over groups,

$$\max_{\mathcal{G} \in \Delta([g])} \mathbb{E}_{k \sim \mathcal{G},\, \mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] = \max_{1 \leq k \leq g} \mathbb{E}_{\mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right].$$

With respect to running time, note that at every time step $t \in [T]$, the algorithm first solves

$$\mathbf{c}(t) = \operatorname*{argmin}_{\mathbf{c} \in C_p(S)} \mathbb{E}_{k \sim D(t)} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right].$$

We note that this can be done in time $O(n^2 p)$ as per Section 3, remembering that

$$\mathbb{E}_{k \sim D(t)} \left[\mathrm{R}_{G_k}(\mathbf{c}) \right] = \mathrm{R}_S(\mathbf{c}, \mathbf{w}(t))$$

i.e., Equation (8) is a weighted regret minimization problem. Then, the algorithm computes $u_k(t)$ for each group k, which can be done in time $O(n)$ (by making a single pass through each customer i and updating the corresponding u_k for $i \in G_k$). Finally, computing the g weights takes $O(g) \leq O(n)$ time. Therefore, each step takes time $O(n^2 p)$, and there are T such steps, which concludes the proof. □

Importantly, Algorithm 1 outputs a *distribution* over p-sets of products; Theorem 4 shows that this distribution $\mathcal{C}_T$ approximates the *ex ante* minmax regret $\widehat{\mathcal{R}}_{\text{fair}}$. $\mathcal{C}_T$ can be used to construct a *deterministic* set of product with good regret guarantees: while each p-set in the support of $\mathcal{C}_T$ may have high group regret, the union of all such p-sets must perform at least as well as $\mathcal{C}_T$ and therefore meet benchmarks $\widehat{\mathcal{R}}_{\text{fair}}$ and $\mathcal{R}_{\text{fair}}$. However, this union may lead to an undesirable blow-up in the number of deployed products. The experiments in Section 6 show how to avoid such a blow-up in practice.

4.3 An *ex post* Minmax Fair Strategy for Few Groups

In this section, we present a dynamic programming algorithm to find p products that approximately optimize the *ex post* maximum regret across groups, as per Program (6). Remember the optimization problem is given by

$$\mathcal{R}_{\text{fair}}(S,p) = \min_{\mathbf{c}\in C_p(S)} \left\{ \max_{1\le k\le g} \mathrm{R}_{G_k}(\mathbf{c}) \right\} \tag{6}$$

The algorithm aims to build a set containing all g-tuples of average regrets $(\mathrm{R}_{G_1},\dots,\mathrm{R}_{G_g})$ that are simultaneously achievable for groups $G_1,\dots,G_g$. However, doing so may be computationally infeasible, as there can be as many regret tuples as there are ways of choosing p products among n consumer thresholds, i.e. $\binom{n}{p}$ of them. Instead, we discretize the set of possible regret values for each group and build a set that only contains rounded regret tuples via recursion over the number of products p. The resulting dynamic program runs efficiently when the number of groups g is a small constant and can guarantee an arbitrarily good additive approximation to the minmax regret. We provide the main guarantee of our dynamic program below and defer the full dynamic program and all technical details to Appendix E.

Theorem 5. *Fix any $\varepsilon > 0$. There exists a dynamic programming algorithm that, given a collection of consumers $S = \{\tau_i\}_{i=1}^n$, groups $\{G_k\}_{k=1}^g$, and a target number of products p, finds a product vector $\mathbf{c} \in C_p(S)$ with $\max_{k\in[g]} \mathrm{R}_{G_k}(\mathbf{c}) \le \mathcal{R}_{fair}(S,p) + \varepsilon$ in time $O\left(n^2 p \left(\left\lceil \frac{Bp}{\varepsilon} \right\rceil + 1\right)^g\right)$.*

This running time is efficient when g is small, with a much better dependency in parameters p and n than the brute force approach that searches over all $\binom{n}{p}$ ways of picking p products.

4.4 The Special Case of Interval Groups

We now consider the case in which each of the g groups G_k is defined by an interval of risk thresholds. I.e., there are risk limits $b_1,\dots,b_{g+1}$ so that G_k contains all consumers with risk limit in $[b_k, b_{k+1})$. Equivalently, every consumer in G_k has risk limit strictly smaller than any member of G_{k+1}.

Our main algorithm in this section is for a decision version of the fair product selection problem, in which we are given interval groups $G_1,\dots,G_g$, a number of products p, a target regret κ, and the goal is to output a collection of p products such that every group has average regret at most κ if possible or else output Infeasible. We can convert any algorithm for the decision problem into one that approximately solves the minmax regret problem by performing binary search over the target regret κ to find the minimum feasible value. Finding an ε-suboptimal set of products requires $O(\log \frac{B}{\varepsilon})$ runs of the decision algorithm, where B is a bound on the minmax regret.

Our decision algorithm processes the groups in order of increasing risk limit, choosing products $\mathbf{c}^{(k)} \subset G_k$ when processing group G_k. Given the maximum risk product $x = \max(\cup_{h=1}^{k-1}\mathbf{c}^{(h)})$ chosen for groups $G_1,\dots,G_{k-1}$, we choose $\mathbf{c}^{(k)} \subset G_k$ to be a set of products of minimal size that achieves regret at most κ for group G_k (together with the product x). Among all smallest product sets satisfying this, we choose one with the product with the highest risk. We call such a set of products *efficient* for group G_k. We argue inductively that the union $\mathbf{c} = \bigcup_{k=1}^g \mathbf{c}^{(k)}$ is the smallest set of products that achieves regret at most κ for all groups. In particular, if $|\mathbf{c}| \le p$ then we have a solution to the decision problem, otherwise it is infeasible. We may also terminate the algorithm early if at any point we have already chosen more than p products.

Formally, for any set S of consumer risk limits, number of products p', default product $x \le \min(S)$, and target regret κ, define $\mathrm{SAT}(S,p',x,\kappa) = \{\mathbf{c} \subset S : |\mathbf{c}| \le p', \mathrm{R}_S(\mathbf{c}\cup\{x\}) \le \kappa\}$ to be the (possibly empty) collection of all sets of at most p' products selected from S for which the average regret of the consumers in S falls below κ using the products $\mathbf{c}$ together the default product x. For any collection of product sets A, we say that the product set $\mathbf{c} \in A$ is efficient in A whenever for all $\mathbf{d} \in A$ we have $|\mathbf{c}| \le |\mathbf{d}|$ and if $|\mathbf{c}| = |\mathbf{d}|$ then $\max(\mathbf{c}) \ge \max(\mathbf{d})$. That is, among all product sets in A, $\mathbf{c}$ has the fewest possible products and, among all such sets it has a maximal highest risk product. Each iteration of our algorithm chooses an efficient product set from $\mathrm{SAT}(G_k,p',x,\kappa)$, where p' is the number of products left to choose and x is the highest risk product chosen so far. Pseudocode is given in Algorithm 2.

Algorithm 2 Fair Product Decision Algorithm

Input: Interval groups $G_1, \ldots, G_g$, max products p, target regret κ

1. Let $\mathbf{c} \leftarrow \emptyset$
2. For $k = 1, \ldots, g$
   (a) If $\text{SAT}(G_k, p - |\mathbf{c}|, \max(\mathbf{c}), \kappa) = \emptyset$ output INFEASIBLE
   (b) Otherwise, let $\mathbf{c}^{(k)}$ be an efficient set of products in $\text{SAT}(G_k, p - |\mathbf{c}|, \max(\mathbf{c}), \kappa)$.
   (c) Let $\mathbf{c} \leftarrow \mathbf{c} \cup \mathbf{c}^{(k)}$.
3. Output $\mathbf{c}$.

Lemma 2. *For any interval groups $G_1, \ldots, G_k$, number of products p, and target regret κ, Algorithm 2 will output a set of at most p products for which every group has regret at most κ if one exists, otherwise it outputs* INFEASIBLE.

Proof. See Appendix G.3. □

It remains to provide an algorithm that finds an efficient set of products in the set $\text{SAT}(S, p', x, \kappa)$ if one exists. The following Lemma shows that we can use a slight modification of the dynamic programming algorithm from Theorem 1 to find such a set of products in $O(|S|^2 p')$ time if one exists, or output INFEASIBLE.

Lemma 3. *There exists an algorithm for finding an efficient set of products in $\text{SAT}(S, p', x, \kappa)$ if one exists and outputs* INFEASIBLE *otherwise. The running time of the algorithm is $O(|S|^2 p')$.*

Proof. A straightforward modification of the dynamic program described in Section 3 allows us to solve the problem of minimizing the regret of population S when using p' products, and the default option is given by a product with risk limit $x \leq \tau$ for all $\tau \in S$ (instead of c_0). The dynamic program runs in time $O(|S|^2 p')$, and tracks the optimal set of p'' products to serve the $z - 1$ lowest risk consumers in S while assuming the remaining consumers are offered product τ_z, for all $z \leq |S|$ and $p'' < p'$. Assuming $\text{SAT}(S, p', x, \kappa)$ is non-empty, one of these z's corresponds to the highest product in an efficient solution, and one of the values of p'' corresponds to the number of products used in said efficient solution. Therefore, the corresponding optimal choice of products is efficient: since it is optimal, the regret remains below κ. It then suffices to search over all values of p' and z after running the dynamic program, which takes an additional time at most $O(|S|p')$. □

Combined, the above results prove the following result:

Theorem 6. *There exists an algorithm that, given a collection of consumers divided into interval groups $S = G_k{}_{k=1}^{g}$ and a number of products p, outputs a set $\mathbf{c}$ of p products satisfying $\max_{k \in [g]} \text{R}_{G_k}(\mathbf{c}) \leq \mathcal{R}(S, p) + \varepsilon$ and runs in time $O(\log(\frac{B}{\varepsilon}) p \sum_{k=1}^{g} |G_k|^2)$, where B is a bound on the maximum regret of any group.*

Proof. Run binary search on the target regret κ using Algorithm 2 together with the dynamic program. Each run takes $O(p \sum_{k=1}^{g} |G_k|^2)$ time, and we need to do $O(\log \frac{B}{\varepsilon})$ runs. □

5 Generalization Guarantees

5.1 Generalization for Regret Minimization Absent Fairness

Suppose now that there is a distribution $\mathcal{D}$ over consumer risk thresholds. Our goal is to find a collection of p products that minimizes the *expected* (with respect to $\mathcal{D}$) regret of a consumer when we only have access to n risk limits sampled from $\mathcal{D}$. For any distribution $\mathcal{D}$ over consumer risk limits and any p, we define $\text{R}_{\mathcal{D}}(\mathbf{c}) = \mathbb{E}_{\tau \sim \mathcal{D}}[\text{R}_\tau(\mathbf{c})]$, which is the *distributional* counterpart of $\text{R}_S(\mathbf{c})$.

In Theorem 7, we provide a generalization guarantee that shows it is enough to optimize $\text{R}_S(\mathbf{c})$ when S is a sample of size $n \geq 2B^2 \varepsilon^{-2} \log(4/\delta)$ drawn *i.i.d.* from $\mathcal{D}$.

Theorem 7 (Generalization Absent Fairness)**.** *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be any non-decreasing function with bounded range B and $\mathcal{D}$ be any distribution over $\mathbb{R}_{\geq 0}$. For any $\varepsilon > 0$ and $\delta > 0$, and for any target number of products p, if $S = \{\tau_i\}_{i=1}^n$ is drawn i.i.d. from $\mathcal{D}$ provided that*

$$n \geq \frac{2B^2 \log(4/\delta)}{\varepsilon^2}$$

then with probability at least $1 - \delta$, we have

$$\sup_{\mathbf{c} \in \mathbb{R}^p_{\geq 0}} \left| \mathrm{R}_S(\mathbf{c}) - \mathrm{R}_{\mathcal{D}}(\mathbf{c}) \right| \leq \varepsilon.$$

Proof. See Appendix G.4. □

The number of samples needed to be able to approximately optimize $\mathrm{R}_{\mathcal{D}}(\mathbf{c})$ to an additive ε factor has a standard square dependency in $1/\varepsilon$ but is independent of the number of products p. This result may come as a surprise, especially given that standard measures of the sample complexity of our function class *do* depend on p. Indeed, we examine uniform convergence guarantees over the function class $\mathcal{F}_p = \{f_{\mathbf{c}}(\tau) \,|\, \mathbf{c} \in \mathbb{R}^p_{\geq 0}\}$ where $f_{\mathbf{c}}(\tau) = \max_{c_j \leq \tau} r(c_j)$[3] via *Pollard's pseudo-dimension* (PDim) [12] and show that the complexity of this class measured by Pollard's pseudo-dimension grows with the target number of products p: $\mathrm{PDim}(\mathcal{F}_p) \geq p$.

Definition 1 (Pollard's Pseudo-dimension)**.** *A class $\mathcal{F}$ of real-valued functions P-shatters a set of points $\tau_1, \ldots, \tau_n$ if there exist a set of "targets" $\gamma_1, \ldots, \gamma_n$ such that for every subset $T \subset [n]$ of point indices, there exists a function, say $f_T \in \mathcal{F}$ such that $f_T(\tau_i) \geq \gamma_i$ if and only if $i \in T$. In other words, all 2^n possible above/below patterns are achievable for the targets $\gamma_1, \ldots, \gamma_n$. The pseudo-dimension of $\mathcal{F}$, denoted by $\mathrm{PDim}(\mathcal{F})$, is the size of the largest set of points that it P-shatters.*

Lemma 4. *Let $r : \mathbb{R} \to \mathbb{R}$ be any function that is strictly increasing on some interval $[a, b] \subset \mathbb{R}$*[4]*. Then for any p, the corresponding class of functions $\mathcal{F}_p$ has $\mathrm{PDim}(\mathcal{F}_p) \geq p$.*

Proof. See Appendix G.4. □

5.2 Generalization for Fairness Across Several Groups

In the presence of g groups, we can think of $\mathcal{D}$ as being a mixture over g distributions, say $\{\mathcal{D}_k\}_{k=1}^g$, where $\mathcal{D}_k$ is the distribution of group G_k, for every k. We let the weight of this mixture on component $\mathcal{D}_k$ be π_k. Let $\pi_{\min} = \min_{1 \leq k \leq g} \pi_k$. In this framing, a sample $\tau \sim \mathcal{D}$ can be seen as first drawing $k \sim \pi$ and then $\tau \sim \mathcal{D}_k$. Note that we have sampling access to the distribution $\mathcal{D}$ and cannot directly sample from the components of the mixture: $\{\mathcal{D}_k\}_{k=1}^g$.

Theorem 8 (Generalization with Fairness)**.** *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be any non-decreasing function with bounded range B and $\mathcal{D}$ be any distribution over $\mathbb{R}_{\geq 0}$. For any $\varepsilon > 0$ and $\delta > 0$, and for any target number of products p, if $S = \{\tau_i\}_{i=1}^n$ consisting of g groups $\{G_k\}_{k=1}^g$ is drawn i.i.d. from $\mathcal{D}$ provided that*

$$n \geq \frac{2}{\pi_{min}} \left(\frac{4B^2 \log(8g/\delta)}{\varepsilon^2} + \log(2g/\delta) \right)$$

then with probability at least $1 - \delta$, we have

$$\sup_{\mathcal{C} \in \Delta\left(\mathbb{R}^p_{\geq 0}\right),\, k \in [g]} \left| \mathop{\mathbb{E}}_{\mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{G_k}(\mathbf{c})\right] - \mathop{\mathbb{E}}_{\mathbf{c} \sim \mathcal{C}} \left[\mathrm{R}_{\mathcal{D}_k}(\mathbf{c})\right] \right| \leq \varepsilon.$$

Proof. See Appendix G.4. □

[3] Note that uniform convergence over $\mathcal{F}_p$ is equivalent to uniform convergence over the class of $R_\tau(\mathbf{c})$ functions.

[4] While not always true, most natural instances of the choice of assets are such that $r(\tau)$ is strictly increasing on some interval $[a, b]$.

6 Experiments

In this section, we present experiments that aim to complement our theoretical results. The code for these experiments can be found at https://github.com/TravisBarryDick/FairConsumerFinance.

Data. The underlying data for our experiments consists of time series of daily closing returns for 50 publicly traded U.S. equities over a 15-year period beginning in 2005 and ending in 2020; the equities chosen were those with the highest liquidity during this period. From these time series, we extracted the average daily returns and covariance matrix, which we then annualized by the standard practice of multiplying returns and covariances by 252, the number of trading days in a calendar year. The mean annualized return across the 50 stocks is 0.13, and all but one are positive due to the long time period spanned by the data. The correlation between returns and risk (standard deviation of returns) is 0.29 and significant at $P = 0.04$. The annualized returns and covariances are then the basis for the computation of optimal portfolios given a specified risk limit τ as per Section 2.[5]

In Figure 1(a), we show a scatter plot of risk vs. returns for the 50 stocks. For sufficiently small risk values, the optimal portfolio has almost all of its weight in cash, since all of the equities have higher risk and insufficient independence. At intermediate values of risk, the optimal portfolio concentrates its weight on just the 7 stocks[6] highlighted in red in Figure 1(a). This figure also plots the optimal risk-return frontier, which generally lies to the northwest (lower risk and higher return) of the stocks themselves, due to the optimization's exploitation of independence. The black dot highlights the optimal return for risk tolerance 0.1, for which we show the optimal portfolio weights in Figure 1(b). Note that once the risk tolerance reaches that of the single stock with highest return (red dot lying on the optimal frontier, representing Netflix), the frontier becomes flat, since at that point the portfolio puts all its weight on this stock.

Algorithms and Benchmarks. We consider both population and group regret and a number of algorithms: the integer linear program (ILP) for optimizing group regret; an implementation of the no-regret dynamics (NR) for group regret described in Algorithm 1; a "sparsified" NR (described below); the dynamic program (DP) of Section 3 for optimizing population regret; and a greedy heuristic, which iteratively chooses the product that reduces population regret the most.[7] We will evaluate each of these algorithms on both types of regret (and provide further details in Appendix F).

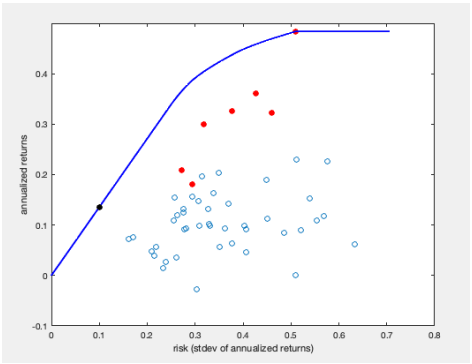
Note that Algorithm 1 outputs a *distribution* over sets of p products; a natural way of extracting a fixed set of products is to take the union of the support (which we refer to as algorithm NR below), but in principle this could lead to far more than p products. We thus sparsify this union with a heuristic that extracts only $p + s$ total products (here $s \geq 0$ is a number of additional "slack" products allowed) by iteratively removing the higher of the two products closest to each other until we have reduced to $p + s$ products. This heuristic is motivated by the empirical observation that the NR dynamics often produce clumps of products close together, and we will see that it performs well for small values of s.

Experimental Design and Results. Our first experiment compares the empirical performance of the algorithms we have discussed on both population and group regret. We focus on a setting with $p = 5$ desired products and 50 consumers drawn with uniform probability from 3 groups. Each group is defined by a Gaussian distribution of risk tolerances with (μ, σ) of $(0.02, 0.002)$, $(0.03, 0.003)$ and $(0.04, 0.004)$ (truncated at 0 if necessary; as per the theory, we add a cash option with zero risk and return). Thus the groups tend to be defined by risk levels, as in the interval groups case, but are noisy and therefore overlap in risk space. The algorithms we compare include the NR algorithm run for $T = 500$ steps; the NR sparsified to contain

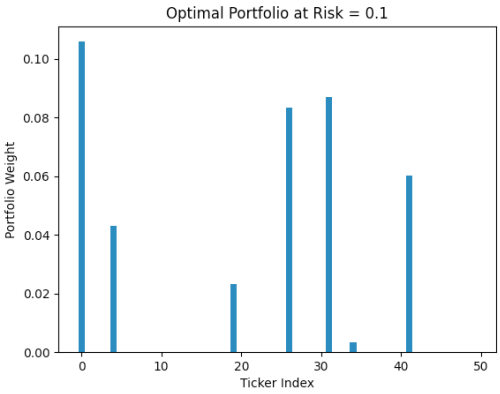
[5] For ease of understanding, here we consider a restriction of the Markowitz portfolio model of [11] and Equation (1) in which short sales are not allowed, i.e. the weight assigned to each asset must be non-negative.

[6] These 7 stocks are Apple, Amazon, Gilead Sciences, Monster Beverage Corporation, Netflix, NVIDIA, and Ross Stores.

[7] In Appendix D, we show that the average population return $f_S(\mathbf{c}) = \frac{1}{n}\sum_i \max_{c_j \leq \tau_i} r(c_j)$ is submodular, and thus the greedy algorithm, which has the advantage of $O(np)$ running time compared to the $O(n^2 p)$ of the DP, also enjoys the standard approximate submodular performance guarantees.



(a) Risk vs. Returns: Scatterplot and Optimal Frontier.

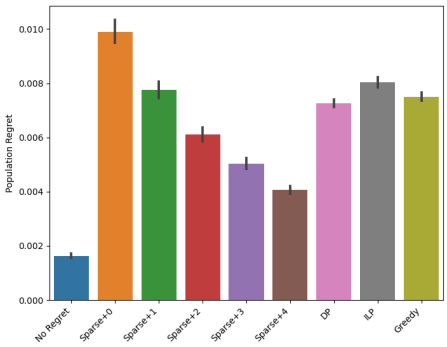


(b) Optimal Portfolio Weights at Risk = 0.1.

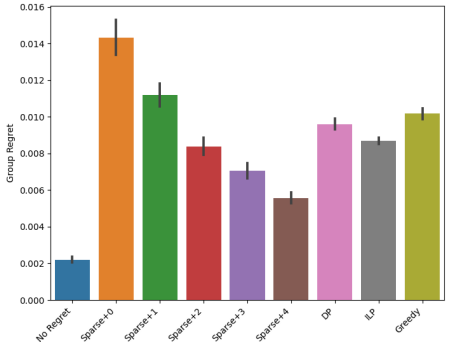
Figure 1: Asset Risks and Returns, Optimal Frontier and Portfolio Weights

from 0 to 4 extra products; the ILP; the DP, and greedy. We compute population and group regret and average results over 100 instances.

Figure 2 displays the results[8]. For both population and group regret, NR performs significantly better than ILP, DP, and greedy but also uses considerably larger numbers of products (between 8 and 19, with a median of 13). The sparsified NR using $s = 0$ additional products results in the highest regret but improves rapidly with slack allowed. By allowing $s = 2$ extra products, the sparsified NR achieves lower population regret than the DP with $p = 5$ products and lower group regret as the ILP with 5 products. While we made no attempt to optimize the integer program (besides using one of the faster solvers available), we found sparsified NR to be about two orders of magnitude faster than solving the ILP (0.3 vs. 14 seconds on average per instance).



(a) Average Population Regret



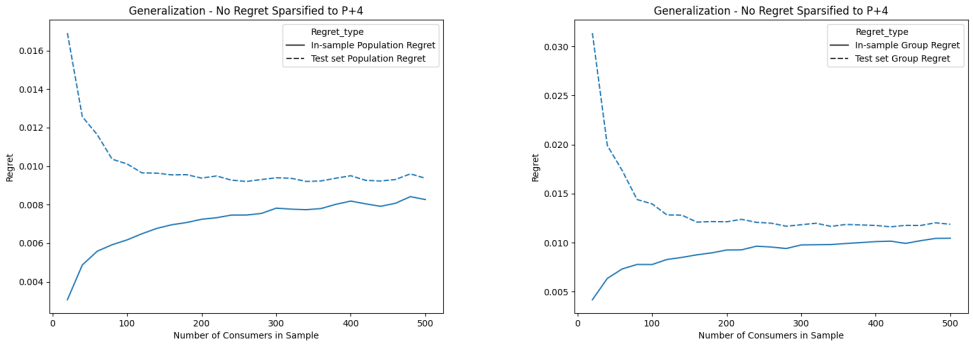
(b) Average Group Regret

Figure 2: Algorithm Performance

Our second experiment explores generalization. We fix $p = 5$ and use sparsified NR with a slack of $s = 4$ for a total of 9 products. We draw a test set of 5000 consumers from the same distribution described above. For sample sizes of $\{25, 50, ..., 500\}$ consumers, we obtain product sets using sparsified NR and calculate the incurred regret using these products on the test set. We repeat this process 100 times for each number of

[8]We provide additional design details and considerations in Appendix F.

consumers and average them. This is plotted in Figure 3; we observe that measured both by population regret as well as by group regret, the test regret decreases as sample size increases. The decay rate is roughly $1/\sqrt{n}$, as suggested by theory, but our theoretical bound is significantly worse due to sub-optimal constants[9]. Training regret increases with sample size because, for a fixed number of products, it is harder to satisfy a larger number of consumers.



(a) Population Regret

(b) Group Regret

Figure 3: Generalization for No-Regret with $p = 5$ Sparsified to $p + s = 9$ Products

References

[1] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 647–664, 2017.

[2] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[3] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020.

[4] Kate Donahue and Jon Kleinberg. Fairness and utilization in allocating resources with uncertain demand. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 658–668, 2020.

[5] Hadi Elzayn, Shahin Jabbari, Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, and Zachary Schutzman. Fair algorithms for learning in allocation problems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 170–179, 2019.

[6] Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. *arXiv preprint arXiv:1706.09847*, 2017.

[7] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332, 1996.

[8] Dan A Iancu and Nikolaos Trichakis. Fairness and efficiency in multiportfolio optimization. *Operations Research*, 62(6):1285–1301, 2014.

[9] The theoretical bound is roughly an order of magnitude higher than the experimental bound; we do not plot it as it makes the empirical errors difficult to see.

[9] Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location. *arXiv preprint arXiv:1908.09041*, 2019.

[10] Sepideh Mahabadi and Ali Vakilian. (individual) fairness for k-clustering. *arXiv preprint arXiv:2002.06742*, 2020.

[11] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x.

[12] David Pollard. *Convergence of stochastic processes*. New York: Springer-Verlag, 1984.

[13] Ariel D Procaccia and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 675–692, 2014.

Appendix

A Probabilistic Tools

Lemma 5 (Additive Chernoff-Hoeffding Bound)**.** *Let $X_1, \ldots, X_n$ be a sequence of i.i.d. random variables with $a \leq X_i \leq b$ and $\mathbb{E}[X_i] = \mu$ for all i. Let $B \triangleq b - a$. We have that for all $s > 0$,*

$$\Pr\left[\left|\frac{\sum_i X_i}{n} - \mu\right| \geq s\right] \leq 2\exp\left(\frac{-2ns^2}{B^2}\right)$$

Lemma 6 (Standard DKW-Inequality)**.** *Let $\mathcal{D}$ be any distribution over $\mathbb{R}$ and $X_1, \ldots, X_n$ be an i.i.d. sample drawn from $\mathcal{D}$. Define $N(t) \triangleq \Pr_{X\sim\mathcal{D}}(X \leq t)$ and $\hat{N}_n(t) \triangleq n^{-1}\sum_{i=1}^n \mathbb{I}\{X_i \leq t\}$ to be the cumulative density functions of $\mathcal{D}$ and the drawn sample, respectively. We have that for all $s > 0$,*

$$\Pr\left[\sup_{t\in\mathbb{R}}\left|\hat{N}_n(t) - N(t)\right| \geq s\right] \leq 2\exp\left(-2ns^2\right)$$

The following Lemma is just a sanity check to make sure that we can apply the DKW inequality to the *strict* version of a cumulative density function.

Lemma 7 (DKW-Inequality for Strict CDF)**.** *Let $\mathcal{D}$ be any distribution over $\mathbb{R}$ and $X_1, \ldots, X_n$ be an i.i.d. sample drawn from $\mathcal{D}$. Define $P(t) \triangleq \Pr_{X\sim\mathcal{D}}(X < t)$ and $\hat{P}_n(t) \triangleq n^{-1}\sum_{i=1}^n \mathbb{I}\{X_i < t\}$ to be the strict cumulative density functions of $\mathcal{D}$ and the drawn sample, respectively. We have that for all $s > 0$,*

$$\Pr\left[\sup_{t\in\mathbb{R}}\left|\hat{P}_n(t) - P(t)\right| \geq s\right] \leq 2\exp\left(-2ns^2\right)$$

Proof. The key idea is to apply the standard DKW inequality to the (non-strict) CDF of the random variable $-X_i$. Towards that end, define $N(t) \triangleq \Pr_{X\sim D}(-X \leq t)$ and $\hat{N}_n(t) \triangleq n^{-1}\sum_{i=1}^n \mathbb{I}\{-X_i \leq t\}$. Then we have the following connection to P and $\hat{P}_n$:

$$\begin{aligned}
\left|\hat{P}_n(t) - P(t)\right| &= \left|1 - \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{X_i \geq t\} - 1 + \Pr_{X\sim D}(X \geq t)\right| \\
&= \left|\frac{1}{n}\sum_{i=1}^n \mathbb{I}\{-X_i \leq -t\} - \Pr_{X\sim D}(-X \leq -t)\right| \\
&= \left|\hat{N}_n(-t) - N(-t)\right|
\end{aligned}$$

The proof is complete by the standard DKW inequality (see Lemma 6). □

Lemma 8 (Multiplicative Chernoff Bound)**.** *Let $X_1, \ldots, X_n$ be a sequence of i.i.d. random variables with $0 \leq X_i \leq 1$ and $\mathbb{E}[X_i] = \mu$ for all i. We have that for all $s > 0$,*

$$\Pr\left[\frac{\sum_i X_i}{n} \leq (1-s)\mu\right] \leq \exp\left(\frac{-n\mu s^2}{2}\right)$$

B A More General Regret Notion

In this section we propose a family of regret functions parametrized by a positive real number that captures the regret notion we have been using throughout the paper as a special case. We will show how minor tweaks allow us to extend the dynamic program for whole population regret minization of Section 3, as well as the no-regret dynamics for *ex ante* fair regret minimization of Section 4.2, to this more general notion of regret.

For a given consumer with risk threshold τ, and for any $\alpha \in (0, \infty]$, the regret of the consumer when assigned to a single product with risk threshold c is defined as follows:

$$\mathrm{R}_{\tau}^{\alpha}(c) = \begin{cases} r(\tau) - r(c) & c \leq \tau \\ \alpha\,(c - \tau) & c > \tau \end{cases} \tag{10}$$

When offering more than one product, say p products represented by $\mathbf{c} = (c_1, c_2, \ldots, c_p)$, the regret of a consumer with risk threshold τ is the best regret she can get using these products. Concretely,

$$\mathrm{R}_{\tau}^{\alpha}(\mathbf{c}) = \min_{1 \leq i \leq p} \mathrm{R}_{\tau}^{\alpha}(c_i) \tag{11}$$

We note that our previous regret notion is recovered by setting $\alpha = \infty$. When $\alpha \neq \infty$, a consumer may be assigned to a product with higher risk threshold, and the consumer's regret is then measured by the difference between her desired risk and the product's risk, scaled by a factor of α. We note that in practice, consumers may have different behavior as to whether they are willing to accept a higher risk product, i.e. different consumers may have different values of α; in this section, we use a unique value of α for all consumers for simplicity of exposition and note that our insights generalize to different consumers having different α's. The regret of a set of consumers $S = \{\tau_i\}_{i=1}^{n}$ is defined as the average regret of consumers, as before. In other words,

$$\mathrm{R}_{S}^{\alpha}(\mathbf{c}) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{R}_{\tau_i}^{\alpha}(\mathbf{c}) \tag{12}$$

We first show in Lemma 9 how we can find one single product to minimize the regret of a set of consumers represented by a set $S = \{\tau_i\}_{i=1}^{n}$. Note this general formulation of regret will allow choosing products that do not necessarily fall onto the consumer risk thresholds. In Lemma 9 we will show that given access to the derivative function $r'(\tau)$, the problem $\min_{c \in \mathbb{R}_+} \mathrm{R}_{S}^{\alpha}(c)$ can be reduced to an optimization problem that can be solved exactly in $O(n)$ time. Before that, we first observe the following property of function $r(\tau)$ (Equation (1)) which will be used in Lemma 9:

Claim 1. *r is a concave function.*

Proof of Claim 1. Let X be the vector of random variables representing m assets and recall μ is the mean of X and Σ is its covariance matrix. Fix $\tau_1, \tau_2 \geq 0$, and $\beta \in (0, 1)$. For $i \in \{1, 2\}$, let $\mathbf{w}_i$ be an optimal solution to the optimization problem for $r(\tau_i)$, i.e., $\mathbf{w}_i$ is such that $r(\tau_i) = \mathbf{w}_i^{\top} \mu$. We want to show that

$$r\left(\beta\tau_1 + (1-\beta)\,\tau_2\right) \geq \beta r(\tau_1) + (1-\beta) r(\tau_2) = \left(\beta \mathbf{w}_1 + (1-\beta)\,\mathbf{w}_2\right)^{\top} \mu \tag{13}$$

So all we need to show is that $\mathbf{w} \triangleq \beta \mathbf{w}_1 + (1-\beta)\mathbf{w}_2$ is feasible in the corresponding optimization problem for $r\left(\beta\tau_1 + (1-\beta)\,\tau_2\right)$. Then, by definition, Equation (13) holds. We have that

$$\mathbb{1}^{\top} \mathbf{w} = \beta(\mathbb{1}^{\top} \mathbf{w}_1) + (1-\beta)(\mathbb{1}^{\top} \mathbf{w}_2) = 1$$

and

$$\begin{aligned}
\mathbf{w}^{\top} \Sigma\, \mathbf{w} &= \beta^2 \mathbf{w}_1^{\top} \Sigma\, \mathbf{w}_1 + (1-\beta)^2 \mathbf{w}_2^{\top} \Sigma\, \mathbf{w}_2 + 2\beta(1-\beta) \mathbf{w}_1^{\top} \Sigma\, \mathbf{w}_2 \\
&\leq \beta^2 \tau_1^2 + (1-\beta)^2 \tau_2^2 + 2\beta(1-\beta) \cdot Cov\left(\mathbf{w}_1^{\top} X, \mathbf{w}_2^{\top} X\right) \\
&\leq \beta^2 \tau_1^2 + (1-\beta)^2 \tau_2^2 + 2\beta(1-\beta) \cdot \sqrt{Var(\mathbf{w}_1^{\top} X) Var(\mathbf{w}_2^{\top} X)} \\
&= \beta^2 \tau_1^2 + (1-\beta)^2 \tau_2^2 + 2\beta(1-\beta) \cdot \sqrt{(\mathbf{w}_1^{\top} \Sigma\, \mathbf{w}_1)(\mathbf{w}_2^{\top} \Sigma\, \mathbf{w}_2)} \\
&\leq \beta^2 \tau_1^2 + (1-\beta)^2 \tau_2^2 + 2\beta(1-\beta) \tau_1 \tau_2 \\
&= \left(\beta\tau_1 + (1-\beta)\tau_2\right)^2
\end{aligned}$$

where the second inequality follows from Cauchy-Schwarz inequality. Note also that $Cov\left(\mathbf{w}_1^{\top} X, \mathbf{w}_2^{\top} X\right) = \mathbf{w}_1^{\top} \Sigma\, \mathbf{w}_2$ and $Var(\mathbf{w}_i^{\top} X) = \mathbf{w}_i^{\top} \Sigma\, \mathbf{w}_i$, for $i \in \{1, 2\}$. □

Lemma 9. *Let $S = \{\tau_i\}_{i=1}^n$ where $\tau_1 \le \tau_2 \le \ldots \le \tau_n$. We have that $\mathrm{R}_S^\alpha(c)$ is a convex function and*

$$\min_{c\in\mathbb{R}_{\ge 0}} \mathrm{R}_S^\alpha(c) = \min\left\{\min_{1\le i\le n} \mathrm{R}_S^\alpha(\tau_i), \min_{1\le i\le n-1} \mathrm{R}_S^\alpha(c_i)\cdot \mathbb{I}_\infty\{\tau_i < c_i < \tau_{i+1}\}\right\} \tag{14}$$

where for every $1 \le i \le n-1$, $c_i = (r')^{-1}\left(\frac{\alpha i}{n-i}\right)$ ($c_i = \infty$ if $(r')^{-1}\left(\frac{\alpha i}{n-i}\right)$ does not exist) and

$$\mathbb{I}_\infty\{\tau_i < c_i < \tau_{i+1}\} \triangleq \begin{cases} 1 & \tau_i < c_i < \tau_{i+1} \\ \infty & \text{otherwise} \end{cases}$$

Proof of Lemma 9. First observe that Claim 1 implies for every τ, $\mathrm{R}_\tau^\alpha(c)$ defined in Equation (10) is convex. Hence, $\mathrm{R}_S^\alpha(c)$ is convex because it is an average of convex functions. We have that for a single product c,

$$\mathrm{R}_S^\alpha(c) = \frac{1}{n}\sum_{j=1}^n \left\{(r(\tau_j) - r(c))\,\mathbb{I}\{c \le \tau_j\} + \alpha\,(c-\tau_j)\,\mathbb{I}\{c > \tau_j\}\right\}$$

Note that $\mathrm{R}_S^\alpha(\tau_1) \le \mathrm{R}_S^\alpha(c)$ for every $c < \tau_1$ and $\mathrm{R}_S^\alpha(\tau_n) \le \mathrm{R}_S^\alpha(c)$ for every $c > \tau_n$. We can therefore focus on the domain $[\tau_1, \tau_n]$ to find the minimum. The function $\mathrm{R}_S^\alpha(c)$ is differentiable everywhere except for the points given by consumers' risk thresholds: $S = \{\tau_i\}_{i=1}^n$. This justifies the first term appearing in the $\min\{\cdot,\cdot\}$ term of Equation (14). For every $1 \le i \le n-1$, the function $\mathrm{R}_S^\alpha(c)$ on domain (τ_i, τ_{i+1}) is differentiable and can be written as:

$$\mathrm{R}_S^\alpha(c) = \frac{1}{n}\left[\alpha\sum_{j\le i}(c-\tau_j) + \sum_{j\ge i+1}(r(\tau_j) - r(c))\right]$$

The minimum of $\mathrm{R}_S^\alpha(c)$ on domain (τ_i, τ_{i+1}) is achieved on points c where

$$\frac{d}{dc}\mathrm{R}_S^\alpha(c) = \frac{1}{n}\left[\alpha i - r'(c)(n-i)\right] = 0 \quad \Longrightarrow \quad r'(c_i) = \frac{\alpha i}{n-i}$$

We note that $\mathrm{R}_S^\alpha(c)$ is a convex function by the first part of this Lemma implying that c_i (if belongs to the domain (τ_i, τ_{i+1})) is a local *minimum*. This justifies the second term appearing in the $\min\{\cdot,\cdot\}$ term of Equation (14) and completes the proof. □

Remark 1. *Given any set of weights $\mathbf{w} \in \mathbb{R}_{\ge 0}^n$ over consumers, Lemma 9 can be easily extended to optimizing the weighted regret of a set of consumers given by:*

$$\mathrm{R}_S^\alpha(\mathbf{c}, \mathbf{w}) = \sum_{i=1}^n w_i\,\mathrm{R}_{\tau_i}^\alpha(\mathbf{c})$$

In fact, for any S and $\mathbf{w}$, we have that $\mathrm{R}_S^\alpha(c, \mathbf{w})$ is a convex function and

$$\min_{c\in\mathbb{R}_{\ge 0}} \mathrm{R}_S^\alpha(c, \mathbf{w}) = \min\left\{\min_{1\le i\le n} \mathrm{R}_S^\alpha(\tau_i, \mathbf{w}), \min_{1\le i\le n-1} \mathrm{R}_S^\alpha(c_i, \mathbf{w})\cdot \mathbb{I}_\infty\{\tau_i < c_i < \tau_{i+1}\}\right\} \tag{15}$$

where for every $1 \le i \le n-1$, $c_i = (r')^{-1}\left(\frac{\alpha\sum_{j\le i} w_j}{\sum_{j\ge i+1} w_j}\right)$ ($c_i = \infty$ if $(r')^{-1}\left(\frac{\alpha\sum_{j\le i} w_j}{\sum_{j\ge i+1} w_j}\right)$ does not exist) and

$$\mathbb{I}_\infty\{\tau_i < c_i < \tau_{i+1}\} \triangleq \begin{cases} 1 & \tau_i < c_i < \tau_{i+1} \\ \infty & \text{otherwise} \end{cases}$$

We now provide the idea behind a dynamic programming approach for choosing p products that minimize the weighted regret of a population $S = \{\tau_i\}_{i=1}^n$. The approach relies on the simple observation that there exists an optimal solution such that if the consumers in set $S(p')$ are assigned to the p'-th product $c_{p'}$, $c_{p'} \in \operatorname{argmin}_{c\in\mathbb{R}^+} \mathrm{R}_{S(p')}(c)$ (for a single product c)[10].

Therefore, to find an optimal choice of products, it suffices to i) correctly guess which subset $S(p')$ of consumers are assigned to the p'-th product, then ii) optimize the choice of product for $S(p')$, which can be done using Lemma 9. Noting that $S(p')$ is an interval for all p', $S(p')$ is entirely characterized by z, the first agent assigned to $c_{p'}$, and n', the last agent assigned to $c_{p'}$. In turn, as in Section 3, our dynamic program can be characterized by a recursive relationship of the form

$$T(n', p') = \min_{z\in\{1,\ldots,n'\}} \left(T(z-1, p'-1) + \min_{c\in\mathbb{R}^+} \sum_{i=z}^{n'} w_i \, \mathrm{R}^{\alpha}_{\tau_i}(c) \right), \tag{16}$$

where $T(n', p')$ represents the minimum weighted regret that can be achieved by providing p' products to consumers 1 to n'. Our dynamic program will implement this recursive relationship.

The results of Section 4.2 immediately extends to this more general regret notion, given that dynamic program (16) can be used as an optimization oracle for the problem solved by the best-response player in Algorithm 1.

C Optimizing for Population vs. Least Well-Off Group: An Example

In this section, we show that optimizing for population regret may lead to arbitrarily bad maximum group regret, and optimizing for maximum group regret may lead to arbitrarily bad population regret. To do so, we consider the following example: there is a set S of n consumers, divided into two groups G_1 and G_2. We let $|G_1| = 1$ and $|G_2| = n-1$ and assume the single consumer in group G_1 has risk threshold τ_1, and the $n-1$ consumers in group G_2 all have the same risk threshold $\tau_2 > \tau_1$. We let $r_1 < r_2$ be the returns corresponding to risk thresholds τ_1, τ_2. Let $p = 1$, i.e. the designer can pick only one product; either $\mathbf{c} = \tau_1$ or $\mathbf{c} = \tau_2$.

When picking $\mathbf{c} = \tau_1$, we have that the average group and population regrets are given by:

$$\mathrm{R}_{G_1}(\tau_1) = 0, \;\; \mathrm{R}_{G_2}(\tau_1) = r_2 - r_1, \;\; \mathrm{R}_S(\tau_1) = \frac{(n-1)(r_2 - r_1)}{n}.$$

When picking $\mathbf{c} = \tau_2$ instead, we have

$$\mathrm{R}_{G_1}(\tau_2) = r_1, \;\; \mathrm{R}_{G_2}(\tau_2) = 0, \;\; \mathrm{R}_S(\tau_2) = \frac{r_1}{n}.$$

Suppose $0 < r_2 - r_1 < r_1$ and $n - 1 > \frac{r_1}{r_2 - r_1}$. Then, the optimal product to optimize for maximum group regret is $\mathbf{c}^{grp} = \tau_1$, and the optimal product to optimize for population regret is $\mathbf{c}^{pop} = \tau_2$. Then, we have that

1. The ratio of population regret using $\mathbf{c}^{grp}$ over that of $\mathbf{c}^{pop}$ is given by:

$$\frac{\mathrm{R}_S(\mathbf{c}^{grp})}{\mathrm{R}_S(\mathbf{c}^{pop})} = \frac{(n-1)(r_2 - r_1)/n}{r_1/n} = \frac{(n-1)(r_2 - r_1)}{r_1}.$$

   This ratio can be made arbitrarily large by letting $n \to +\infty$, at $\frac{r_2 - r_1}{r_1}$ constant.

[10] On the one hand, for all p' and given $S(p')$, picking $c_{p'}$ in such a manner provides a lower bound on the achievable average regret. On the other hand, $c_{p'}$ yields $S(p')$ as a set of consumers assigned to $c_{p'}$ in an optimal solution. Indeed, if consumers in $S(p')$ strictly preferred picking a different product, this could only be because they would get strictly better regret from doing so: by picking a different product, they would then decrease the population regret below our lower bound, which is a contradiction.

2. The ratio of maximum group regret using $\mathbf{c}^{pop}$ over that of $\mathbf{c}^{grp}$ is given by:

$$\frac{\max\left(\mathrm{R}_{G_1}(\mathbf{c}^{pop}),\ \mathrm{R}_{G_2}(\mathbf{c}^{pop})\right)}{\max\left(\mathrm{R}_{G_1}(\mathbf{c}^{grp}),\ \mathrm{R}_{G_2}(\mathbf{c}^{grp})\right)} = \frac{r_1}{r_2 - r_1}.$$

This ratio can be made arbitrarily large by letting $r_2 - r_1 \to 0$ at r_1 constant.

D Approximate Population Regret Minimization via Greedy Algorithm

Recall that given $S = \{\tau_i\}_{i=1}^n$ and set $\mathbf{c} \subseteq S$ of products, the population regret of S is given by

$$\mathrm{R}_S(\mathbf{c}) = \frac{1}{n}\sum_{i=1}^n \mathrm{R}_{\tau_i}(\mathbf{c}) = \frac{1}{n}\sum_{i=1}^n \left(r(\tau_i) - f_{\tau_i}(\mathbf{c})\right) = \frac{1}{n}\sum_{i=1}^n r(\tau_i) - f_S(\mathbf{c})$$

where for any τ, $f_\tau(\mathbf{c}) \triangleq \max_{c_j \le \tau} r(c_j)$, and

$$f_S : 2^S \to \mathbb{R}_{\ge 0}, \quad f_S(\mathbf{c}) \triangleq \frac{1}{n}\sum_{i=1}^n f_{\tau_i}(\mathbf{c}).$$

First, note that when no product is offered, consumers pick the cash option c_0 and get return $r(c_0) = 0$:

Fact 1 (Centering)**.** *For any S, $f_S(\emptyset) = 0$.*

Second, f_S is immediately monotone non-decreasing, as consumers deviate to an additional product only when they get higher return from doing so:

Fact 2 (Monotonicity)**.** *For any S, if $\mathbf{c} \subseteq \mathbf{d}$, then $f_S(\mathbf{c}) \le f_S(\mathbf{d})$.*

Finally, f_S is submodular:

Claim 2 (Submodularity)**.** *For any S, f_S is submodular.*

Proof. We first show that for every i, $f_{\tau_i}(\cdot)$ is submodular: for any $\mathbf{c}, \mathbf{d} \subseteq S$, we have that

$$f_{\tau_i}(\mathbf{c} \cup \mathbf{d}) + f_{\tau_i}(\mathbf{c} \cap \mathbf{d}) \le f_{\tau_i}(\mathbf{c}) + f_{\tau_i}(\mathbf{d}).$$

If $\mathbf{c} = \emptyset$ or $\mathbf{d} = \emptyset$, the claim trivially holds. So assume $\mathbf{c}, \mathbf{d} \neq \emptyset$. Let c be such that $f_{\tau_i}(\mathbf{c} \cup \mathbf{d}) = r(c)$. If $c = c_0 = 0$, the claim holds because all four terms above will be zero. If $c \in \mathbf{c}$, the claim holds because $f_{\tau_i}(\mathbf{c} \cup \mathbf{d}) = f_{\tau_i}(\mathbf{c})$ and $f_{\tau_i}(\mathbf{d}) \ge f_{\tau_i}(\mathbf{c} \cap \mathbf{d})$ by Fact 2. The same argument holds when $c \in \mathbf{d}$ by symmetry. The proof is complete by noting that any simple average of submodular functions (in general, any linear combination with non-negative coefficients) is submodular. □

Remark 2. *Claim 2 extends to any weighted set of consumers: fix any set $S = \{\tau_i\}_{i=1}^n$ of consumers and any nonnegative weight vector $\mathbf{w} \in \mathbb{R}^n_{\ge 0}$. Then the function $f_S : 2^S \to \mathbb{R}$ given by $f_S(\mathbf{c}) = \sum_{i=1}^n w_i f_{\tau_i}(\mathbf{c})$ is submodular.*

We therefore have that for any S and any target number of products p,

$$\min_{\mathbf{c} \in C_p(S)} \mathrm{R}_S(\mathbf{c}) = \frac{1}{n}\sum_{i=1}^n r(\tau_i) - \max_{\mathbf{c} \subseteq S,\, |\mathbf{c}| = p} f_S(\mathbf{c})$$

where by Facts 1, 2 and Claim 2, the maximization problem in right-hand side is: *maximization of a nonnegative monotone submodular function with a cardinality constraint.* Using the *Greedy Algorithm* (that runs in $O(np)$ time), we get p products represented by $\mathbf{c}^{grd}$ such that

$$R_S(\mathbf{c}^{grd}) \le \frac{1}{n}\sum_{i=1}^n r(\tau_i) - \left(1 - e^{-1}\right) \cdot \max_{\mathbf{c} \subseteq S,\, |\mathbf{c}| = p} f_S(\mathbf{c})$$

E An *ex post* Minmax Fair Strategy for Few Groups, Extended

Given bound B on the maximum group regret and a step size of α, we let

$$N^{(\alpha)} \triangleq \left\{ i\alpha : \ i \in \left\{ 0, \dots, \left\lceil \frac{B}{\alpha} \right\rceil \right\} \right\}$$

be a net of discretized regret values in $[0, B]$ with discretization size α. Given any regret R, we let $\text{ceil}_\alpha(\text{R})$ be the regret obtained by rounding R up to the closest higher regret value in $N^{(\alpha)}$. I.e., $\text{ceil}_\alpha(\text{R})$ is uniquely defined so as to satisfy $\text{R} \le \text{ceil}_\alpha(\text{R}) < \text{R} + \alpha$ and $\text{ceil}_\alpha(\text{R}) \in N^{(\alpha)}$. Our dynamic program implements the following recursive relationship:

$$\mathcal{F}^{(\alpha)}(n', p') \triangleq$$
$$\left\{ \left(\text{ceil}_\alpha \left(\text{R}_{G_k} + \sum_{i=z}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} \text{R}_{\tau_i}(\tau_z) \right) \right)_{k=1}^{g} \text{ s.t. } z \le n', \ (\text{R}_{G_k})_{k=1}^{g} \in \mathcal{F}^{(\alpha)}(z-1, p'-1) \right\} \tag{17}$$

where for all $n' \le n$,

$$\mathcal{F}^{(\alpha)}(n', 0) \triangleq \left\{ \left(\sum_{i=1}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} r(\tau_i) \right)_{k=1}^{g} \right\}. \tag{18}$$

Note that $\mathcal{F}^{(\alpha)}(n', 0)$ contains a single regret tuple, whose k-th coordinate is the weighted regret of agents $G_k \cap \{1, \dots, n'\}$ when using weight $1/|G_k|$ and offering no product.

Intuitively, $\mathcal{F}^{(\alpha)}(n', p')$ keeps track of a rounded up version of the feasible tuples of weighted group regrets when using weight $1/|G_k|$ in group G_k and when offering p' products and considering the regret of consumers 1 to n' only. The corresponding set of products used to construct the regret tuples in $\mathcal{F}^{(\alpha)}(n', p')$ can be kept in a hash table whose keys are the regret tuples in $\mathcal{F}^{(\alpha)}(n', p')$; we denote such a hash table by $\mathcal{I}^{(\alpha)}(n, p)$. While there can be several p'-tuples of products that lead to the same rounded regret tuple in $\mathcal{F}^{(\alpha)}(n', p')$, the dynamic program only stores one of them in the corresponding entry in the hash table at each time step. The program terminates after computing $\mathcal{F}^{(\alpha)}(n, p)$, $\mathcal{I}^{(\alpha)}(n, p)$, and outputting the product vector in $\mathcal{I}^{(\alpha)}(n, p)$ corresponding to the regret tuple with smallest regret for the worst-off group in $\mathcal{F}^{(\alpha)}(n, p)$. The sets $\mathcal{F}^{(\alpha)}(n, p)$, $\mathcal{I}^{(\alpha)}(n, p)$ satisfy the following guarantee:

Lemma 10. *Fix any $\alpha > 0$. Let $(\text{R}_{G_1}, \dots, \text{R}_{G_g})$ be any regret tuple that can be achieved using p products. There exist consumer indices $(z_1, \dots, z_p) \in \mathcal{I}^{(\alpha)}(n, p)$ such that the corresponding regret tuple $(\text{R}^{(\alpha)}_{G_1}, \dots, \text{R}^{(\alpha)}_{G_g}) \in \mathcal{F}^{(\alpha)}(n, p)$ satisfies*

$$\text{R}_{G_k}(\mathbf{c}) \le \text{R}^{(\alpha)}_{G_k} \le \text{R}_{G_k} + p\alpha, \ \forall k \in [g],$$

where $\mathbf{c} = (\tau_{z_1}, \dots, \tau_{z_p})$.

We provide the proof of Lemma 10 in Appendix E. In particular, Lemma 10 implies that the dynamic program run with discretization parameter α approximately minimizes the maximum regret across groups (i.e., the optimal value of Program (6)) within an additive approximation factor of $p\alpha$. Letting $\alpha = \frac{\varepsilon}{p}$ yields an ε-approximation to the minmax regret.

The running time of our dynamic program is summarized below:

Theorem 9. *Fix any $\alpha > 0$. There exists a dynamic programming algorithm that, given a collection of consumers $S = \{\tau_i\}_{i=1}^n$, groups $\{G_k\}_{k=1}^g$, and a target number of products p, computes $\mathcal{F}^{(\alpha)}(n, p)$ and $\mathcal{I}^{(\alpha)}(n, p)$ in time $O\left(n^2 p \left(\left\lceil \frac{B}{\alpha} \right\rceil + 1\right)^g\right)$.*

When the desired accuracy is ε, the dynamic program uses $\alpha = \frac{\varepsilon}{p}$ and has running time $O\left(n^2 p \left(\left\lceil \frac{Bp}{\varepsilon} \right\rceil + 1\right)^g\right)$.

Proof. Note that each set $\mathcal{F}^{(\alpha)}(n',p')$ and $\mathcal{I}^{(\alpha)}(n',p')$ built by the dynamic program has size at most $\left(\lceil \frac{B}{\alpha} \rceil + 1\right)^g$, as $\mathcal{F}^{(\alpha)}(n',p')$ only contains regret values in $N^{(\alpha)}$ by construction and $\mathcal{I}^{(\alpha)}(n',p')$ contains one product tuple per regret tuple in $\mathcal{F}^{(\alpha)}(n',p')$.

Each sum used in the dynamic program can be computed in time $O(1)$, given that $\sum_{i=1}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} r(\tau_i)$ and $\sum_{i=1}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|}$ have been pre-computed for all $n' \in [n], k \in [g]$ and stored in a hash table. The pre-computation and storage of these partial sums can be done in $O(gn)$ time.

Now, each time step of the dynamic program corresponds to the p'-th product with $p' \leq p$. In each time step p', we construct $O(n)$ sets $\mathcal{F}^{(\alpha)}(n',p')$, one for each value of n'. For each value of n', the dynamic program searches over i) $z \in [n]$ and ii) at most $\left(\lceil \frac{B}{\alpha} \rceil + 1\right)^g$ tuples of regret in $\mathcal{F}^{(\alpha)}(z, p'-1)$. Therefore, building $\mathcal{F}^{(\alpha)}(n,p)$ can be done in time $O\left(n^2 p \left(\lceil \frac{B}{\alpha} \rceil + 1\right)^g\right)$

Finally, finding the tuple with the smallest maximum regret in $\mathcal{F}^{(\alpha)}(n,p)$ requires searching over at most $\left(\lceil \frac{R}{\alpha} \rceil + 1\right)^g$ product tuples in $\mathcal{I}^{(\alpha)}(n,p)$.

Therefore, running the dynamic program requires time $O\left(n^2 p \left(\lceil \frac{B}{\alpha} \rceil + 1\right)^g\right)$. □

Proof of Lemma 10

We let $\mathcal{F}(n',p')$ be the set of weighted regret tuples that are achievable using p' thresholds when only agents 1 to n' are considered, and the regret of agents in group G_k is weighted by $1/|G_k|$ – importantly, this reweighting is independent of the choice of n' and computes the average regret of a group as if all of its consumers were present. The set $\mathcal{F}(n,p)$ contains all feasible tuples of average regret given groups $\{G_k\}_{k=1}^g$. Importantly, $\mathcal{F}(n,p)$ is different from $\mathcal{F}^{(\alpha)}(n,p)$: $\mathcal{F}(n,p)$ contains all achievable tuples of regret, even those that do not belong to the net $N^{(\alpha)}$; in turn $\mathcal{F}(n,p)$ may contain up to $\binom{n}{p}$ regret tuples. In comparison, $\mathcal{F}^{(\alpha)}(n,p)$ is a smaller set of size at most $\left(\lceil \frac{B}{\alpha} \rceil + 1\right)^g$ that only contains regret tuples with values in the net $N^{(\alpha)}$. $\mathcal{F}^{(\alpha)}(n,p)$ is built with the intent of approximating the true set of achievable regret tuples, $\mathcal{F}(n,p)$.

The proof idea is to show that $\mathcal{F}^{(\alpha)}(n,p)$ is a good approximation of $\mathcal{F}(n,p)$, as intended. More precisely, we want to show that for every feasible regret tuple $(\mathrm{R}_{G_k})_{k=1}^g$ in $\mathcal{F}(n,p)$, the discretized set $\mathcal{F}^{(\alpha)}(n,p)$ contains a regret tuple $\left(\mathrm{R}^{(\alpha)}_{G_k}\right)_{k=1}^g$ that is at most $p\alpha$ away from $(\mathrm{R}_{G_k})_{k=1}^g$; further, the corresponding product vector $\mathbf{c}^{(\alpha)} \in \mathcal{I}^{(\alpha)}(n,p)$ has true, *unrounded* regret also within $p\alpha$ of $(\mathrm{R}_{G_k})_{k=1}^g$.

We start by showing that $\mathcal{F}(n,p)$ obeys the following recursive relationship:

Claim 3.

$$\mathcal{F}(n',p') = \left\{ \left(\mathrm{R}_{G_k} + \sum_{i=z}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) \right)_{k=1}^{g} \; s.t.\; z \leq n',\; (\mathrm{R}_{G_k})_{k=1}^g \in \mathcal{F}(z-1, p'-1) \right\} \tag{19}$$

where

$$\mathcal{F}(n',0) \triangleq \mathcal{F}^{(\alpha)}(n',0) = \left\{ \left(\sum_{i=1}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} r(\tau_i) \right)_{k=1}^{g} \right\}. \tag{20}$$

Proof. When $p' = 0$, the result is immediate, noting that agents 1 to n' are assigned to the cash option c_0 with 0 return and incur regret $r(\tau_i)$ each. The total regret in group G_k, considering agents 1 to n' and reweighting regret by $1/|G_k|$, is given by

$$\sum_{i=1}^{n'} \frac{\mathbb{1}\{i \in G_k\}}{|G_k|} r(\tau_i).$$

Now, take $p' > 0$. Fix the highest offered product to be τ_z, corresponding to consumer z. First, agents $z, \ldots, n'$ are assigned to product τ_z corresponding to consumer z; the weighted regret incurred by these agents, limited to those in group G_k, is exactly

$$\sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \left(r(\tau_i) - r(\tau_z)\right) = \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z).$$

The remaining agents are 1 to $z-1$ and have $p'-1$ products available to them; hence, a regret tuple $(\mathrm{R}_{G_1}, \ldots, \mathrm{R}_{G_g})$ can be feasibly incurred by these agents if and only if $(\mathrm{R}_{G_1}, \ldots, \mathrm{R}_{G_g}) \in \mathcal{F}(z-1, p'-1)$, by definition of $\mathcal{F}(z-1, p'-1)$. To conclude the proof, it is enough to note that the total regret incurred by agents in group G_k is the sum of the regrets of agents $\{1, \ldots, z-1\} \cap G_k$ and the regret of agents $\{z, \ldots, n'\} \cap G_k$. □

We now show that for any $\alpha > 0$, $\mathcal{F}^{(\alpha)}(n, p)$ provides a $p\alpha$-additive approximation to the true set of possible regret tuples $\mathcal{F}(n, p)$:

Lemma 11. *For all $p \in \mathcal{N}$, for all $z \in [n]$, and for any regret tuple $(\mathrm{R}_{G_1}, \ldots, \mathrm{R}_{G_g}) \in \mathcal{F}(n, p)$, there exists a regret tuple $(\mathrm{R}^{(\alpha)}_{G_1}, \ldots, \mathrm{R}^{(\alpha)}_{G_g}) \in \mathcal{F}^{(\alpha)}(n, p)$ such that $\mathrm{R}^{(\alpha)}_{G_k} \leq \mathrm{R}_{G_k} + p\alpha$ for all $k \in [g]$.*

Proof. The proof follows by induction on p. At step $p' \leq p$, the induction hypothesis states that for all n', for any regret tuple $(\mathrm{R}_{G_1}, \ldots, \mathrm{R}_{G_g}) \in \mathcal{F}(n', p')$, there exists a regret tuple $(R^{(\alpha)}_{G_1}, \ldots, R^{(\alpha)}_{G_g}) \in \mathcal{F}^{(\alpha)}(n', p')$ such that $R^{(\alpha)}_{G_k} \leq \mathrm{R}_{G_k} + p'\alpha$ for all $k \in [g]$.

First, when $p' = 0$, the induction hypothesis holds immediately: by definition, $\mathcal{F}^{(\alpha)}(n', 0) = \mathcal{F}(n', 0)$. Now, suppose the induction hypothesis holds for $p'-1$. Pick any regret tuple $(\mathrm{R}_{G_1}(n', p'), \ldots, \mathrm{R}_{G_g}(n', p')) \in \mathcal{F}(n', p')$; we will show that the induction hypothesis holds for this tuple. First, note that there exists z and $(\mathrm{R}_{G_1}(z-1, p'-1), \ldots, \mathrm{R}_{G_g}(z-1, p'-1)) \in \mathcal{F}(z-1, p'-1)$ such that

$$\mathrm{R}_{G_k}(n', p') = \mathrm{R}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) \ \forall k \in [g]$$

by definition of $\mathcal{F}(n', p')$. Further, by induction hypothesis, there exists a g-tuple of rounded regret $(\mathrm{R}^{(\alpha)}_{G_1}(z-1, p'-1), \ldots, \mathrm{R}^{(\alpha)}_{G_g}(z-1, p'-1))$ in $\mathcal{F}^{(\alpha)}(z-1, p'-1)$ such that

$$\mathrm{R}^{(\alpha)}_{G_k}(z-1, p'-1) \leq \mathrm{R}_{G_k}(z-1, p'-1) + (p'-1)\alpha \ \forall k \in [g].$$

Combining the above two equations, we get that

$$\begin{aligned}
&\mathrm{R}^{(\alpha)}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) \\
&\leq \mathrm{R}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) + (p'-1)\alpha \\
&= \mathrm{R}_{G_k}(n', p') + (p'-1)\alpha \ \forall k \in [g].
\end{aligned}$$

Now, let $\mathrm{R}^{(\alpha)}_{G_k}(n', p') = \mathrm{ceil}_\alpha \left(\mathrm{R}^{(\alpha)}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}[i \in G_k]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z)\right)$. First, $(\mathrm{R}^{(\alpha)}_{G_1}(n', p'), \ldots, \mathrm{R}^{(\alpha)}_{G_g}(n', p'))$

is in $\mathcal{F}^{(\alpha)}(n', p')$ by definition. Second,

$$\begin{aligned}
\mathrm{R}^{(\alpha)}_{G_k}(n', p') &= \mathrm{ceil}_\alpha \left(\mathrm{R}^{(\alpha)}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) \right) \\
&\leq \mathrm{R}^{(\alpha)}_{G_k}(z-1, p'-1) + \sum_{i=z}^{n'} \frac{\mathbb{1}\left[i \in G_k\right]}{|G_k|} \mathrm{R}_{\tau_i}(\tau_z) + \alpha \\
&\leq \mathrm{R}_{G_k}(n', p') + p'\alpha.
\end{aligned}$$

This concludes the induction. □

To conclude the proof, let $(\mathrm{R}_{G_1}, \ldots, \mathrm{R}_{G_g})$ be a tuple of regret that can be achieved using p products. The tuple belongs to $\mathcal{F}(n, p)$, by definition of $\mathcal{F}(n, p)$. Therefore, by Lemma 11, there exists a regret tuple $(\mathrm{R}^{(\alpha)}_{G_1}, \ldots, \mathrm{R}^{(\alpha)}_{G_g}) \in \mathcal{F}^{(\alpha)}$ such that

$$\mathrm{R}^{(\alpha)}_{G_k} \leq \mathrm{R}_{G_k} + p\alpha \ \forall k \in [g].$$

Let $\mathbf{c} \triangleq \{c_1, \ldots, c_p\} \in \mathcal{I}^{(\alpha)}(n, p)$ be the product vector that was used to construct regret tuple $\left(\mathrm{R}^{(\alpha)}_{G_k}\right)_{k=1}^{g}$. Since at each step p', the dynamic program rounds regret tuples to higher values, a simple induction shows that

$$\mathrm{R}_{G_k}(\mathbf{c}) \leq \mathrm{R}^{(\alpha)}_{G_k} \ \ \forall k \in [g].$$

Combining the two above equations, we get the result:

$$\mathrm{R}_{G_k}(\mathbf{c}) \leq \mathrm{R}^{(\alpha)}_{G_k} \leq \mathrm{R}_{G_k} + p\alpha \ \forall k \in [g].$$

F Additional Experiment Details

All experiments were run on a consumer laptop without GPU (13-inch Macbook Pro 2016 with 2 GHz Intel Core i5 and 8 GB of RAM), using Python 3.6.5 (and in particular, NumPy 1.17.1 and Pandas 1.0.3). Experiment 1 took about 16.7 minutes in total. Experiment 2 took about 3.8 hours in total. For reproducibility, we began each experiment with a fixed random seed (10).

For the No-Regret experiments, we have two parameters to choose: the number of steps and the step size[11]. Note that theory suggests that step size should be a function of the number of steps desired (or vice versa), but common practice among related algorithms is to try many step sizes and run until convergence. We adopt this approach, guided by theory. To do so, we calculate a lower bound on the theoretical instance-specific step size using the sum of rewards as a loose upper bound B on the maximum possible group regret. We then consider applying the following multipliers to this lower bound on the step size: we take (1,10,100,1000,10000) as a small set of possible multipliers and examine convergence for each of them. In all experiments, we pick the step size multiplier after empirically observing that it both i) provides a good approximation to the optimal *ex ante* regret and ii) converges in few time steps; we note that our choices of multipliers enjoy good performance guarantees in practice, as evidenced by Figure 2.

G Omitted Proofs

G.1 Proof of Lemma 1

The first property is immediate, because when $p' = 0$, we do not choose any products, and there is only one valid solution that assigns all consumers to the zero-risk cash product. For this solution, the weighted regret

[11] We implement the Multiplicative Weight Update of Algorithm 1 in exponential form; i.e., we write $\beta = \exp(-\eta)$. The code takes the regret values $R_{G_k}(\mathbf{c})$ as losses, takes η/B as the step size, and updates the k-th weight in each round t by a multiplicative factor of $\exp(-\eta R_{G_k}(\mathbf{c}(t))/B)$, as per Algorithm 1. This allows us to easily translate the weights in log space so as to avoid numerical overflow issues.

is simply the sum of the weighted returns for each consumer's bespoke portfolio.

We now turn to proving the second property. For any consumer index $z \in [n']$, define $T(n', p', z)$ to be the optimal weighted regret for the first n' consumers using p' products subject to the constraint that the highest risk product has risk threshold set to τ_z. That is,

$$T(n', p', z) = \min_{\substack{\mathbf{c}=(c_1,\ldots,c_{p'})\subset S[n'] \\ c_1,\ldots,c_{p'}\leq\tau_z \\ c_{p'}=\tau_z}} \mathrm{R}_{S[n']}(\mathbf{c}, \mathbf{w}[n']).$$

The products achieving weighted regret $T(n', p', z)$ must choose $c_{p'} = \tau_z$ and choose $c_1, \ldots, c_{p'-1}$ to be optimal products for the consumers indexed $1, \ldots, z-1$, who are not served by the product $c_{p'}$ because it is too high risk. On the other hand, the weighted regret of consumers $z, \ldots, n'$ when assigned to a product with risk limit τ_z is given by $\sum_{i=z}^{n'} w_i \cdot \big(r(\tau_i) - r(\tau_z)\big)$. Together, this implies that

$$T(n', p', z) = T(z-1, p'-1) + \sum_{i=z}^{n'} w_i \cdot \big(r(\tau_i) - r(\tau_z)\big).$$

On the other hand, for any n' and p', we have that $T(n', p') = \min_{z\in[n']} T(n', p', z)$, because the optimal p' products for the first n' consumers has a largest risk threshold equal to some consumer risk threshold. Combining these equalities gives

$$T(n', p') = \min_{z\in[n']} T(n', p', z) = \min_{z\in[n']} T(z-1, p'-1) + \sum_{i=z}^{n'} w_i \cdot \big(r(\tau_i) - r(\tau_z)\big),$$

as required.

G.2 Proof of Theorem 2

Proof. Note there is a one-to-one relation (on some domain $[0, a]$ for risk thresholds) between any risk threshold τ and its corresponding return given by $r(\tau)$ by Equation (1) — we will therefore (only for simplicity of exposition) construct our instance by defining a set of returns instead of risk thresholds. Let $A \triangleq \{r, 2r, \ldots, (p+1)r\}$ be the set of all possible returns in our instance for some constant $r > 0$. We will take $r \equiv 1$ for simplicity but our proof extends to any $r > 0$.

Our instance construction is simple. If $g \leq p+1$, we partition A into g subsets of size at most $\lceil \frac{p+1}{g} \rceil$, and we let each group be defined as one of the partition elements. In this case, there will be one consumer for every return value in A. For e.g., if $p = 4$ and $g = 2$, we can define $G_1 = \{1, 2, 3\}$ and $G_2 = \{4, 5\}$. If $g > p+1$ (allowing consumers having the same return) we let each group be defined by a single return value in A. For e.g., if $p = 2$ and $g = 4$, we can define $G_1 = \{1\}$, $G_2 = \{2\}$, and $G_3 = G_4 = \{3\}$. To formalize this construction, define $s \triangleq \min\{g, p+1\}$ and let $\{P_i\}_{i=1}^{s}$ be a s-sized partition of R such that $\max_i |P_i| = \lceil \frac{p+1}{g} \rceil$. Instance S of size $n = \max\{g, p+1\}$ is defined as follows.

$$S = \{G_k\}_{k=1}^{g} \quad \text{where} \quad \forall\, k \in [g]: \quad G_k = \begin{cases} P_k & k \leq s \\ P_s & k > s \end{cases}$$

Let $A_p = \{B \subseteq A : |B| = p\}$ and observe that $|A_p| = p+1$. We have that

$$\mathcal{R}_{\text{fair}}(S, p) \overset{(1)}{=} \min_{B\in A_p} \left\{ \max_{1\leq k\leq g} \mathrm{R}_{G_k}(B) \right\} \overset{(2)}{=} \frac{1}{\max_k |G_k|} \overset{(3)}{=} \frac{1}{\lceil \frac{p+1}{g} \rceil}$$

where (1) follows from the definition of $\mathcal{R}_{\text{fair}}(S, p)$ in this specific instance that all consumer returns are specified by the set A of size $p+1$. (2) follows from the fact that for any set of products $B \in A_p$, all

groups G_k that have a consumer with return $A \setminus B$ will incur an average regret of $1/|G_k|$. (3) holds because $\max_k |G_k| = \max_i |P_i| = \lceil \frac{p+1}{g} \rceil$. Next, by looking at the uniform distribution over A_p,

$$\widehat{\mathcal{R}}_{\text{fair}}(S,p) \overset{(1)}{\leq} \max_{1\leq k\leq g} \frac{1}{p+1} \sum_{B\in A_p} \mathrm{R}_{G_k}(B) \overset{(2)}{=} \max_{1\leq k\leq g} \frac{1}{p+1} \sum_{r\in G_k} \frac{1}{|G_k|} = \frac{1}{p+1}$$

where (1) follows from the definition of $\widehat{\mathcal{R}}_{\text{fair}}(S,p)$. (2) follows from the fact that for every group k and every $r \in G_k$, there is one (and only one) set of products, namely $B = A \setminus \{r\}$, that makes G_k incur a regret of $1/|G_k|$. We therefore have that

$$\frac{\widehat{\mathcal{R}}_{\text{fair}}(S,p)}{\mathcal{R}_{\text{fair}}(S,p)} \leq \frac{1}{p+1} \left\lceil \frac{p+1}{g} \right\rceil.$$

□

G.3 Proof of Lemma 2

To ease notation, for any product sets $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(g)}$, we let $\mathbf{c}^{(h:k)} = \bigcup_{\ell=h}^{k} \mathbf{c}^{(\ell)}$ denote the union of the product sets with indices in $\{h, \ldots, k\}$.

Suppose our feasibility problem has a solution. It is enough to show that there exists product sets $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(g)}$ such that $\mathbf{c}^{(1:g)}$ is a feasible set of at most p products and $\mathbf{c}^{(k)} \subset G_k$ is efficient in $\mathrm{SAT}(G_k, p - |\mathbf{c}^{(1:k-1)}|, \max(\mathbf{c}^{(1:k-1)}), \kappa)$ for all $k \in [g]$. Note that by the definition of efficiency and a straightforward induction, any product sets $\mathbf{c}^{(1)}, \ldots \mathbf{c}^{(g)}$ and alternative product sets $\mathbf{c}'^{(1)}, \ldots, \mathbf{c}'^{(g)}$ satisfying these properties must have $|\mathbf{c}'^{(k)}| = |\mathbf{c}^{(k)}|$ and $\max(\mathbf{c}'^{(k)}) = \max(\mathbf{c}^{(k)})$ for all $k \in [g]$, hence

$$\begin{aligned} \mathrm{SAT}_k &\triangleq \mathrm{SAT}(G_k, p - |\mathbf{c}^{(1:k-1)}|, \max(\mathbf{c}^{(1:k-1)}), \kappa) \\ &= \mathrm{SAT}(G_k, p - |\mathbf{c}'^{(1:k-1)}|, \max(\mathbf{c}'^{(1:k-1)}), \kappa) \end{aligned}$$

does not depend on the specific choice of $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(g)}$ that satisfies the above assumptions. Since the algorithm may only output Infeasible if SAT_k is empty for some k, it can only do so when no $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(g)}$ satisfying the above assumptions exists. When such product sets exist, the algorithm outputs one, and $\mathbf{c}^{(1:g)}$ is guaranteed to use at most p products and have regret at most κ in each group by definition of $\mathrm{SAT}(G_k, p - |\mathbf{c}^{(1:k-1)}|, \max(\mathbf{c}^{(1:k-1)}), \kappa)$.

Assuming our problem is feasible with p products, we show the following induction hypothesis: for all $k \leq g$ there exist product sets $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(k)}$ such that $\mathbf{c}^{(j+1)}$ is efficient in $\mathrm{SAT}(G_j, p - |\mathbf{c}^{(1:j)}|, \max(\mathbf{c}^{(1:j)}), \kappa)$ for all $j \leq k-1$, that can be completed in a set products $\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}$ that has size at most p and guarantees regret at most κ in each group, where $\mathbf{d}^{(j)} \subset G_j$.

First, the induction hypothesis immediately hold for $k = 0$: since the problem is feasible, there exists a product vector $\mathbf{d}$ that uses at most p products and guarantees group regret of at most κ. Now, suppose the induction hypothesis holds for k; we will show it holds for $k+1$. Take $\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}$ that has size at most p and guarantees regret at most κ in each group, such that $\mathbf{c}^{(j+1)}$ is efficient in $\mathrm{SAT}(G_j, p - |\mathbf{c}^{(1:j)}|, \max(\mathbf{c}^{(1:j)}), \kappa)$ for all $j \leq k-1$. Take $\mathbf{c}^{(k+1)}$ to be efficient in $\mathrm{SAT}(G_j, p - |\mathbf{c}^{(1:k)}|, \max(\mathbf{c}^{(1:k)}), \kappa)$; we have two cases:

1. $|\mathbf{c}^{(k+1)}| = |\mathbf{d}^{(k+1)}|$ and $\max(\mathbf{c}^{(k+1)}) \geq \max(\mathbf{d}^{(k+1)})$. Let us consider product set $\mathbf{c}^{(1:k+1)} \cup \mathbf{d}^{(k+2:g)}$. Then, $|\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}| = |\mathbf{c}^{(1:k+1)} \cup \mathbf{d}^{(k+2:g)}| \leq p$. Further, no group has regret over κ. Indeed, compared to when using $\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}$, we have that: i) the regret of groups G_1 to G_k is unaffected as they only use products $\mathbf{c}^{(1:k)}$; ii) the regret of group G_{k+1} stays below κ by satisfiability of $\mathbf{c}^{(k+1)}$; iii) the regret of groups $G_{k+2}, \ldots, G_g$ cannot increase because agents who used product $\max(\mathbf{d}^{(k+1)})$ get weakly lower regret from using $\max(\mathbf{c}^{(k+1)}) \geq \max(\mathbf{d}^{(k+1)})$, and the remaining agents can keep using the same products in $\mathbf{d}^{(k+2:g)}$. Since the regret of all groups remains below κ under products $\mathbf{c}^{(1:k+1)} \cup \mathbf{d}^{(k+2:g)}$ by the induction hypothesis, this remains true when using products $\mathbf{c}^{(1:k+1)} \cup \mathbf{d}^{(k+2:g)}$.

2. $|\mathbf{c}^{(k+1)}| < |\mathbf{d}^{(k+1)}|$. In that case, let $\tilde{\mathbf{d}}^{(k+2)} = \mathbf{d}^{(k+2)} \cup \{b_{k+2}\}$ (where b_{k+2} is the smallest threshold in group G_{k+2}). Consider product offering $\mathbf{c}^{(1:k+1)} \cup \tilde{\mathbf{d}}^{(k+2)} \cup \mathbf{d}^{(k+3:g)}$. First, we note that

$$\begin{aligned}|\mathbf{c}^{(1:k+1)} \cup \tilde{\mathbf{d}}^{(k+2)} \cup \mathbf{d}^{(k+3:g)}| &\leq |\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1)} \cup \tilde{\mathbf{d}}^{(k+2)} \cup \mathbf{d}^{(k+3:g)}| - 1 \\ &= |\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}| \\ &\leq p.\end{aligned}$$

Second, note that the regrets of all groups remain under κ. Indeed, compared to when offering products $\mathbf{c}^{(1:k)} \cup \mathbf{d}^{(k+1:g)}$, we have: i) the regrets of groups $G_1, \ldots, G_k$ stay the same, as before; ii) the regret of group G_{k+1} stays below κ by satisfiability of $\mathbf{c}^{(k+1)}$, as before; iii) the regret of groups $G_{k+2}, \ldots, G_g$ can also only decrease, because all agents who were assigned to $\max(\mathbf{d}^{(k+1)}) \leq b_{k+2}$ are now assigned to the higher return product b_{k+2}, while the remaining agents stay assigned to the same product in $\mathbf{d}^{(k+2:g)}$.

This concludes the proof.

G.4 Proofs of Generalization Theorems

Proof of Theorem 7. Let $f_{\mathbf{c}}(\tau) \triangleq \max_{c_j \leq \tau} r(c_j)$ and observe that using this notation, for any τ, $\mathrm{R}_\tau(\mathbf{c}) = r(\tau) - f_{\mathbf{c}}(\tau)$. To prove the claim of the theorem, first note that,

$$\sup_{\mathbf{c} \in \mathbb{R}^p_{\geq 0}} |\mathrm{R}_S(\mathbf{c}) - \mathrm{R}_{\mathcal{D}}(\mathbf{c})| \leq \left| \mathop{\mathbb{E}}_{\tau \sim S}[r(\tau)] - \mathop{\mathbb{E}}_{\tau \sim \mathcal{D}}[r(\tau)] \right| + \sup_{\mathbf{c} \in \mathbb{R}^p_{\geq 0}} \left| \mathop{\mathbb{E}}_{\tau \sim S}[f_{\mathbf{c}}(\tau)] - \mathop{\mathbb{E}}_{\tau \sim D}[f_{\mathbf{c}}(\tau)] \right| \tag{21}$$

where "$\tau \sim S$" means sampling τ from the uniform distribution over S. We have that by an application of additive Chernoff-Hoeffding bound (see Lemma 5), with probability at least $1 - \delta/2$, given the assumption on sample size n,

$$\left| \mathop{\mathbb{E}}_{\tau \sim S}[r(\tau)] - \mathop{\mathbb{E}}_{\tau \sim \mathcal{D}}[r(\tau)] \right| \leq \sqrt{\frac{B^2 \log(4/\delta)}{2n}} \leq \frac{\varepsilon}{2} \tag{22}$$

Now let us focus on the second term appearing in Equation (21). Consider any vector of products $\mathbf{c} = (c_1, \ldots, c_p) \in \mathbb{R}^p_{\geq 0}$ where, without loss of generality, we assume $c_1 \leq \cdots \leq c_p$. Letting $c_0 = 0$ and $c_{p+1} = \infty$, these products partition $\mathbb{R}_{\geq 0}$ into p intervals $[c_j, c_{j+1})$ for $j \in [p]$ such that $f_{\mathbf{c}}(\tau) = r(c_j)$ for all $\tau \in [c_j, c_{j+1})$. We can rewrite $f_{\mathbf{c}}(\tau)$ as a telescoping sum that adds a term $r(c_j) - r(c_{j-1})$ for each interval j up to and including the interval containing τ:

$$\begin{aligned}
f_{\mathbf{c}}(\tau) &= \sum_{j=0}^{p} r(c_j) \mathbb{I}\{c_j \leq \tau < c_{j+1}\} \\
&= \sum_{j=0}^{p} r(c_j) \left(\mathbb{I}\{c_j \leq \tau\} - \mathbb{I}\{c_{j+1} \leq \tau\} \right) \\
&= \sum_{j=0}^{p} r(c_j) \mathbb{I}\{c_j \leq \tau\} - \sum_{j=0}^{p} r(c_j) \mathbb{I}\{c_{j+1} \leq \tau\} \\
&= \sum_{j=0}^{p} r(c_j) \mathbb{I}\{c_j \leq \tau\} - \sum_{j=1}^{p+1} r(c_{j-1}) \mathbb{I}\{c_j \leq \tau\} \\
&= r(c_0) \mathbb{I}\{c_0 \leq \tau\} - r(c_p) \mathbb{I}\{c_{p+1} \leq \tau\} + \sum_{j=1}^{p} \mathbb{I}\{\tau \geq c_j\} \big(r(c_j) - r(c_{j-1}) \big) \\
&= \sum_{j=1}^{p} \mathbb{I}\{\tau \geq c_j\} \big(r(c_j) - r(c_{j-1}) \big)
\end{aligned}$$

remembering for the last equality that c_0 is the risk-free cash option and has return $r(c_0) = 0$ and that $c_{k+1} = +\infty$ is the final dummy product that corresponds to having an infinite risk and always satisfies $\tau < c_{p+1}$. Taking expectations of this expression converts the indicator into the complementary CDF (CCDF), and therefore we have

$$\begin{aligned}
\sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0}} \left| \mathop{\mathbb{E}}_{\tau\sim S}[f_{\mathbf{c}}(\tau)] - \mathop{\mathbb{E}}_{\tau\sim\mathcal{D}}[f_{\mathbf{c}}(\tau)] \right| &= \sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0}} \left| \sum_{j=1}^{p} \left(\Pr_{\tau\sim S}(\tau \geq c_j) - \Pr_{\tau\sim\mathcal{D}}(\tau \geq c_j) \right) \cdot \big(r(c_j) - r(c_{j-1})\big) \right| \\
&\leq \sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0}} \sum_{j=1}^{p} \left| \Pr_{\tau\sim S}(\tau \geq c_j) - \Pr_{\tau\sim\mathcal{D}}(\tau \geq c_j) \right| \cdot \big(r(c_j) - r(c_{j-1})\big) \\
&\leq \sup_{t\in\mathbb{R}} \left| \Pr_{\tau\sim S}(\tau \geq t) - \Pr_{\tau\sim D}(\tau \geq t) \right| \cdot \sum_{j=1}^{p} \big(r(c_j) - r(c_{j-1})\big) \qquad (23) \\
&\leq \sqrt{\frac{\log(4/\delta)}{2n}} \cdot (r(c_p) - r(c_1)) \\
&\leq \sqrt{\frac{B^2 \log(4/\delta)}{2n}} \\
&\leq \frac{\varepsilon}{2}
\end{aligned}$$

where the first inequality follows from the triangle inequality and the fact that r is non-decreasing: $r(c_j) \geq r(c_{j-1})$ for $j = 1, \ldots, p$. The third inequality holds with probability $1 - \delta/2$ and follows from the Dvoretzky-Kiefer-Wolfowitz inequality (see Lemma 7). The last inequality follows by the assumption on sample size n. Combining Equations (21) and (22) and (23), completes the proof of the theorem. □

Proof of Theorem 8. Let $S = \{\tau_i\}_{i=1}^n$ be a set of consumers of size n drawn from the distribution $\mathcal{D}$ that is partitioned into g groups: $\{G_k\}_{k=1}^g$. Let $n_k = |G_k|$ denote the size of group k. It follows from the uniform convergence of Theorem 7 that for any group k, so long as $n_k \geq 2B^2\varepsilon^{-2}\log(4/\delta')$, with probability $1 - \delta'$, $\sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0}} |\mathrm{R}_{G_k}(\mathbf{c}) - \mathrm{R}_{\mathcal{D}_k}(\mathbf{c})| \leq \varepsilon$. This implies by a union bound that with probability at least $1 - \delta/2$, so long as $n_k \geq 2B^2\varepsilon^{-2}\log(8g/\delta)$ for all k,

$$\sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0},\, k\in[g]} |\mathrm{R}_{G_k}(\mathbf{c}) - \mathrm{R}_{\mathcal{D}_k}(\mathbf{c})| \leq \varepsilon$$

But note for any group k, $n_k \sim \mathrm{Bin}(n, \pi_k)$ where $\mathrm{Bin}(m, q)$ denotes a binomial random variable with m trials and success probability q. By an application of the Multiplicative Chernoff bound (see Lemma 8), as well as a union bound, we have that with probability $1 - \delta/2$, for any group k,

$$n_k \geq n\pi_k - \sqrt{2n\pi_k \log(2g/\delta)} \geq 2B^2\varepsilon^{-2}\log(8g/\delta)$$

where the second inequality follows by the assumption on n in the theorem statement. We can therefore conclude by another union bound that, with probability at least $1 - \delta$,

$$\sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0},\, k\in[g]} |\mathrm{R}_{G_k}(\mathbf{c}) - \mathrm{R}_{\mathcal{D}_k}(\mathbf{c})| \leq \varepsilon$$

The proof is complete by noting that

$$\begin{aligned}
\sup_{\mathcal{C}\in\Delta\left(\mathbb{R}^p_{\geq 0}\right),\, k\in[g]} \left| \mathop{\mathbb{E}}_{\mathbf{c}\sim\mathcal{C}} [\mathrm{R}_{G_k}(\mathbf{c})] - \mathop{\mathbb{E}}_{\mathbf{c}\sim\mathcal{C}} [\mathrm{R}_{\mathcal{D}_k}(\mathbf{c})] \right| &\leq \sup_{\mathcal{C}\in\Delta\left(\mathbb{R}^p_{\geq 0}\right),\, k\in[g]} \mathop{\mathbb{E}}_{\mathbf{c}\sim\mathcal{C}} [|\mathrm{R}_{G_k}(\mathbf{c}) - \mathrm{R}_{\mathcal{D}_k}(\mathbf{c})|] \\
&= \sup_{\mathbf{c}\in\mathbb{R}^p_{\geq 0},\, k\in[g]} |\mathrm{R}_{G_k}(\mathbf{c}) - \mathrm{R}_{\mathcal{D}_k}(\mathbf{c})|
\end{aligned}$$

□

Proof of Lemma 4. Our goal is to show that the class of functions $\mathcal{F}_p$ can P-shatter at least p points (or consumer risk limits). Let $\tau_1 < \cdots < \tau_p \in [a, b]$ be any sequence of increasing consumer risk limits. The high-level idea is to design a target γ_i and a product c_i for consumer i so that the return for consumer i is at least the target γ_i if and only if the product c_i is included (as opposed to a default product lower than any target). Given that we have p products to choose, we can decide to include the product for each consumer or not independently, and therefore we can achieve all above/below target patterns to shatter the consumers.

Formally, define targets $\gamma_1, \ldots, \gamma_p$ and a collection of candidate products $c_0, \ldots, c_p$ so that

$$r(c_0) < \gamma_1 < r(c_1) < r(\tau_1) < \gamma_2 < r(c_2) < r(\tau_2) < \cdots < \gamma_p < r(c_p) < r(\tau_p).$$

Note that this is always possible since the return function r is strictly increasing on the interval $[a, b]$. For any product vector $\mathbf{d} \in \mathbb{R}^p$ of product risk thresholds, we have that $f_{\mathbf{d}}(\tau_i) \geq \gamma_i$ if and only if there is some product d_i such that $r(d_i) \in [\gamma_i, r(\tau_i)]$. Now, for any $T \subset [p]$, define a product risk threshold vector $\mathbf{d} \in \mathbb{R}^p$ by $d_i = c_i$ if $i \in T$ and $d_i = c_0$ if $i \notin T$. By the above argument, and the definition of c_i and γ_i, it follows that $f_{\mathbf{d}}(\tau_i) \geq \gamma_i$ if and only if $i \in T$. Therefore $\mathcal{F}_p$ shatters $\tau_1, \ldots, \tau_p$ and $\text{PDim}(\mathcal{F}_p) \geq p$. □