Fairness Methods in Optimization and Artificial Intelligence

Violet (Xinying) Chen

Tepper School of Business
Carnegie Mellon University

Committee: John Hooker (Chair)
Hoda Heidari
Fatma Kılınç-Karzan
Alec Morton

This dissertation is submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Operations Research

May 2022

Acknowledgements

I am extremely fortunate to have received the support, guidance and friendship from many people throughout my journey to the completion of this dissertation and the PhD program.

First of all, I would like to express my deepest gratitude to my advisor, John Hooker. I am indebted to him for opening up my horizon on ethics, an area I never expected to cross into when I started the program. I have been, and will always be, inspired by his passion and insights for interdisciplinary research and diligence to exploring new questions.

I am also deeply grateful to all my committee members. I thank Fatma Kılınç-Karzan for her extensive guidance and advice on research, teaching and service throughout my years in the PhD program. I thank Hoda Heidari for all the interesting discussions related to fairness in artificial intelligence, which greatly expanded my knowledge about the area, and her genuine support throughout my job search. I thank Alec Morton for the insightful questions raised during both my proposal and defense. I enjoyed thinking through these questions and discovering new perspectives.

I would like to thank my wonderful collaborators, Joshua Williams and Derek Leben, in completing Chapter six. I was fortunate to learn new techniques and knowledge from both of them during the collaboration. I also wish to acknowledge the helpful guidance from Thiago Serra, who shared service opportunities with me to get involved with the broader community and offered valuable suggestions for my job search. I also thank all the professors that I have had the opportunity to interact with and learn from throughout these years.

I thank all the precious friendships which have made even the challenging times during the past five years enjoyable. Many thanks to my cohorts: Sagnik Das, Özgün Elçi, Melda Korkut, Kyra Gan, Su Jia, Thomas Lavastida, Yuyan Wang. Special thanks to Yuyan (and her dog Mochi) for sharing a roof with me for three years and being there for all the ups-and-downs, and Sagnik Das for always selflessly helping out whenever needed. I also thank Musa Çeldir, Serim Hwang, Anthony Karahalios, Neda Mirzaeian, Yanhan (Savannah) Tang, Ziye Tang and Ryan Shi for all the happy memories. Additionally, I thank the support from Daniel de Roux, Tom Krumpolc and Mik Zlatin while working together as the board of CMU Informs Chapter.

Lastly, I am deeply indebted to my family in China. To my parents, Jian Chen and Yufeng Gao, I cannot begin to express my gratitude. They are the most supportive and loving parents. They have always believed in me and provided the best opportunities for me. To my dog, Model, I thank her for all the happiness and love.

Abstract

With increasing deployment of optimization and Artificial Intelligence (AI) to assist high-stake real life decisions, fairness has become an essential factor of consideration for both the designers and users of these tools. This dissertation studies new approaches for formulating, attaining and eliciting fairness. Chapter one begins with a brief introduction of the background on fairness and a selection of common fairness measures.

Chapter two studies balancing fairness and efficiency through optimization models. We propose new social welfare functions (SWFs) as combined measures of two well-known criteria, Rawlsian leximax fairness and utilitarianism. We then design a procedure to sequentially maximize these SWFs with mixed integer/linear programming models to find socially optimal solutions. This approach has practical potentials on a wide range of resource allocation applications, and is demonstrated on realistic size applications in healthcare provision and shelter assignment for disaster preparation.

Chapter three considers an optimization task motivated by fair machine learning (ML). When developing fair ML algorithms, it is useful to understand the computational costs of fairness in comparison to the standard non-fair setting. For fair ML methods that utilize optimization models for training, specialized optimization algorithms have potentials to offer better computational performances than generic solvers. In this chapter, I explore this question for support vector machines (SVMs), and design block coordinate descent type algorithms to train SVMs containing linear fairness constraints. Numerical experiments highlight that the new specialized algorithms are more efficient than an off-the-shelf solver for training fair SVMs.

Chapter four examines social welfare optimization as a general paradigm for formalizing welfare-based fairness in AI systems. Contrary to commonly used statistical bias metrics in fair AI, optimizing a social welfare objective supports broader perspective on fairness motivated by distributive justice considerations. We propose in-processing and post-processing integration schemes between social welfare optimization and AI, in particular, ML and rule-based AI. We implement and evaluate the integration schemes on a simulated loan processing instance. The empirical results demonstrate the advantages of the proposed integration strategies. We conclude this chapter by highlighting research directions to pursue for a holistic view of welfare-based fairness.

The next two chapters explore the human-centric perspective to elicit people's moral values through preference learning. Chapter five studies a general preference learning framework based on online learning (OL) from revealed preferences: a learner learns an agent's private utility function through interactions in a changing environment. Through designing a new convex loss function, we

design a flexible OL framework that enables a unified treatment of usual loss functions from literature and supports a variety of online convex optimization algorithms. This framework has advantages in regret performance and solution time over other OL algorithms from the literature.

Lastly, chapter six explores a moral decision-making inspired task. This chapter considers the modelling and elicitation of people's dynamic ethical judgments in the sequential allocation of resources. We utilize a Markov Decision Process model to represent a sequential allocation task, where the state rewards capture people's moral preferences, thus people's ethical judgments are reflected via policy rewards. We design a preference inference model which relies on active preference-based reward learning to infer the unknown reward function. The learning framework is applied in human-subject experiments on Amazon Mechanical Turk to understand people's moral reasoning in a hypothetical scenario of allocating scarce medical resources.

Table of contents

Chapter 1

Introduction

[1] Fairness is important to people's daily lives and society. As a key component of ethics, fairness seeks equitable distribution of resources and opportunities to promote equality, justice and social well-beings. In real-life decisions and policies, it is morally desirable and socially sustainable to pursue fairness. With optimization and Artificial intelligence (AI) tools increasingly applied to assist high-stake decisions, such as, optimizing humanitarian operations with resource allocation models, using machine learning models for criminal risk assessment, fairness is no longer a nice-to-have advantage, but rather a necessary property of practical and trustworthy decision support models and algorithms. This dissertation speaks to the rising demand of better understanding of fairness, and develops new approaches and insights for fairness in optimization and AI.

In contrast to the widely recognized significance of fairness, there is no uniformly superior interpretation and operationalization of fairness. Besides the large variety of fairness related theories proposed across academic fields, people hold diverse fairness perceptions as influenced by their personal backgrounds and specific decision contexts. This dissertation explores two perspectives for the fundamental questions, what is fairness and how to be fair

One perspective is the formulation of fairness optimization models in alignment with ethics principles. Optimization has long been used to support decision making. Most practical optimization problems involves the allocation decisions of some resources. In recent years, optimization also has been extensively studied in artificial intelligence, especially in machine learning (ML) where optimization models are often core components in a ML framework. Conventionally, optimization models have a efficiency-driven objective function. For resource allocation decisions, the objective describes seeking the most effective use of resources. For example, a government agency aims to maximize the total population benefits from healthcare provision, a company tries to minimize costs and maximize profits during facility expansion. For optimization-based ML methods, the standard objective focuses on minimizing the total prediction losses to seek high accuracy. By pursuing efficiency goals, these conventional optimization models may lead to unfair results, so a crucial task is to incorporate fairness and equity into existing optimization models. While it is normally

[1]This chapter uses excerpts from a joint work with John Hooker

straightforward to formulate an ef ciency objective, fairness can be understood in multiple ways, with no generally accepted method for representing any of them in a mathematical idiom. Moreover, the growth of integrated methods, such as, optimization based on ML predictions, in modern decision-making adds to the challenge of formulating fairness models. As discussed above, this perspective opens up interesting modeling and computational questions in the broad eld of AI. Chapter 2, 3, 4 studies such questions.

The other perspective is to elicit fairness beliefs, and more generally ethical values, from human stakeholders. Conventional approaches to de ning and justifying fairness are driven by principles, namely, the central planner determines what are the suitable ethical values and what should be the resulting fairness notion. Due to the large variety of fairness concepts and the potentially different positions held by the planner and stakeholders, the chosen fairness notion could be incompatible with stakeholders' moral beliefs and ethics principles. The possible incompatibility has led to the recent research thread of human-centric fairness, a bottom-up strategy aiming to learn what people believe to be fair under different decision contexts then bring their judgments into the formulation of fairness. While there are situations where people's perspectives should not be incorporated due to irrationality or bias, in general, decision makers would seek to align fairness interventions with what the people desire, so that they would be more welcoming to these interventions. A core component of this perspective is the modeling, elicitation and learning of preferences: chapter 5 studies a general preference learning question in an online setup, and chapter 6 explores a concrete moral preference learning task.

1.1 Background on Fairness

Fairness has a long history of being studied across elds including philosophy, sociology, psychology and economics. These early literature provide ethical and conceptual foundations for the more recent discussion of fairness in operations research and AI, where the goal is to operationalize fairness via optimization and machine learning for concrete applications. A well known framework to distinguish fairness concepts is to perceive fairness via distributive justice versus procedural justice. The high-level intuition is that distributive justice concerns fairness in the outcomes from a decision, whereas procedural justice emphasizes fairness in the process of decision making. This dissertation focuses on outcome fairness via the distributive justice perspective.

Mathematical formalization of outcome fairness can be further distinguished into utility-based and parity-based de nitions. Utility-based fairness is broadly used in resource allocation to attain a fair distribution of utilities, which can be pro ts, negative costs, or some other bene ts appropriate to the application. Utilities were initially proposed as part of the theory of utilitarianism, which seeks to maximize the overall well-beings of the population. In particular, the standard ef ciency formulation follows utilitarianism and uses the total or average utility as a measure of the population's overall welfare. Parity-based fairness is typically considered in machine learning to seek unbiased treatments towards the involved groups or individuals. There is a large number of fairness definitions in both types, and samples of popular choices in literature and practice are summarized below.

1.1.1 Utility-based Fairness

Suppose $\mathbf{u} = (u_1, \ldots, u_n)$ is a vector of utilities distributed across parties $1, \ldots, n$ and $W(\mathbf{u})$ is a utility-based fairness measure. $W(\mathbf{u})$ is a function aggregating the utility values to evaluate $\mathbf{u}$ with respect to the selected fairness notion. We categorize utility-based fairness criteria into three types.

Inequality Measures

The first type measures fairness via the degree of equality in the distribution of utilities, for which several statistical metrics have been proposed (Cowell 2000, Jenkins and Van Kerm 2011). There is a wide variety of philosophical opinion on the ethical significance of equality, ranging from the view that we have an irreducible obligation to strive for equality, to the view that inequality is unfair only when it reduces total utility (Frankfurt 2015, Parfit 1997, Scanlon 2003). In any event, it is generally acknowledged that equality is not the same concept (or cluster of concepts) as fairness, even when the two are closely related. An equality metric can be appropriate in a context where a specifically egalitarian distribution is the primary goal, without regard for efficiency or other forms of equity. Inequality measures have been used for inequity averse optimization in a broad range of applications. More recently, inequality measures are also considered in the growing area of algorithmic fairness. We next introduce several commonly studied definitions.

Measures of relative dispersion[2]. The *relative range* of utilities is an inequality metric, that, when negated, yields the fairness measure,

$$W(\mathbf{u}) = -(1/\bar{u})\left(u_{max} - u_{min}\right)$$

where $u_{max} = \max_i\{u_i\}$, $u_{min} = \min_i\{u_i\}$, and $\bar{u} = (1/n)\sum_i u_i$.

Another dispersion metric is the *relative mean deviation*, which measures inequality more comprehensively by considering all utilities rather than only the minimum and maximum. The corresponding fairness measure is,

$$W(\mathbf{u}) = -(1/\bar{u})\sum_i |u_i - \bar{u}|.$$

The *coefficient of variation* is the normalized standard deviation, leading to the fairness measure below. It may be appropriate when large deviations from the mean are disproportionately significant, but it has the possible drawback of computational difficulty due to the quadratic component.

$$W(\mathbf{u}) = -\frac{1}{\bar{u}}\Big[\frac{1}{n}\sum_i (u_i - \bar{u})^2\Big]^{\frac{1}{2}}.$$

[2] All of the following dispersion measures are normalized by the mean utility so as to be invariant under rescaling of utilities.

Gini coefficient and Hoover index. The *Gini coefficient* is by far the best known measure of inequality, as it is routinely used to measure income and wealth inequality (Gini 1912). It is proportional to the area between the Lorenz curve and a diagonal line representing perfect equality. Lorenz curve plots the proportion of the total wealth or benefits accumulated to the bottom $x\%$ of the population, thus indicating the degree of inequality via its deviation from the perfect equality line. By definition, Gini coefficient vanishes under perfect equality. We can use the negative Gini coefficient as a fairness measure,

$$W(\mathbf{u}) = -\frac{1}{2\bar{u}n^2}\sum_{i,j}|u_i - u_j|.$$

The *Hoover index* is also related to the Lorenz curve, as it is proportional to the maximum vertical distance between the Lorenz curve and a diagonal line representing perfect equality (Hoover 1936). It can be interpreted as the fraction of total utility that would have to be redistributed to achieve perfect equality. The corresponding fairness measure is

$$W(\mathbf{u}) = -\frac{1}{2n\bar{u}}\sum_{i}|u_i - \bar{u}|.$$

Fairness for the Disadvantaged

Rather than focus solely on inequality, fairness measures can prioritize the disadvantaged. Far and away the most famous of such measures is the difference principle of John Rawls (1999), a maximin criterion that is based on careful philosophical argument and debated in a vast literature (surveyed in Freeman 2003, Richardson and Weithman 1999). The difference principle can be plausibly extended to a lexicographic maximum principle. There is also the McLoone index, which is a statistical measure that emphasizes the lot of the less advantaged.

The Rawlsian maximin criterion has been a popular fairness measure for decades. Early works on fair resource allocation, such as bandwidth allocation, often choose the maximin criterion to seek the best possible performance for the worst-off service among services competing for bandwidth. Recent research has applied the criterion to more diverse problem contexts, including peer review paper assignment, ridesharing, etc.

Rawlsian criteria. The Rawlsian *difference principle* states that inequality should exist only to the extent that it is necessary to improve the lot of the worst-off. It is defended with a social contract argument that, in its simplest form, maintains that the structure of society must be negotiated in an "original position" in which people do not yet know their station in society. Rawls argues that one can rationally assent to the possibility of ending up on the bottom only if that person would have been even worse off in any other social structure, whence an imperative to maximize the lot of the worst-off. The principle is intended to apply only to the design of social institutions, and only to the distribution of "primary goods," which are goods that any rational person would want. Yet it can be adopted as a general criterion for distributing utility, namely a *maximin* criterion that maximizes the smallest utility as the fairness measure,

$$W(\mathbf{u}) = \min_i \{u_i\}.$$

The maximin criterion can be plausibly extended to *lexicographic maximization* (*leximax*). Leximax is achieved by first maximizing the smallest utility subject to resource constraints, then the second smallest, and so forth.

McLoone index. The *McLoone index* compares the total utility of individuals at or below the median utility to the utility they would enjoy if all were brought up to the median utility. The index is 1 if nobody's utility is strictly below the median, and it approaches 0 if the utility distribution has a very long lower tail (on the assumption that all utilities are positive). The McLoone index benefits the disadvantaged by rewarding equality in the lower half of the distribution, but it is unconcerned by the existence of very rich individuals in the upper half. The definition is,

$$W(\mathbf{u}) = \frac{1}{|I(\mathbf{u})|\tilde{u}} \sum_{i \in I(\mathbf{u})} u_i$$

where $\tilde{u}$ is the median of utilities in $\mathbf{u}$ and $I(\mathbf{u})$ is the set of indices of utilities at or below the median, so that $I(\mathbf{u}) = \{i \mid u_i \leq \tilde{u}\}$.

Combined Fairness and Efficiency

The previous examples are all pure fairness measures, which are appropriate when there is no need to balance fairness against the overall well-being of the population. However, practical situations frequently call for both fairness and efficiency to be explicitly considered. We next review three common schemes for combining the two criteria and give examples of the combined measures following each scheme. Chapter 2 provides further discussion of these measures.

Convex combination. The most obvious approach is to maximize a convex combination of fairness and efficiency. Suppose $F(\mathbf{u})$ is a pure fairness measure, then a combined measure is

$$W(\mathbf{u}) = (1-\lambda)\sum_i u_i + \lambda F(\mathbf{u}).$$

The combination strategy is applicable to both inequality indices and fairness for the disadvantaged. For instance, Eisenhandler and Tzur (2019) and Mostajabdaveh et al. (2019) propose combined measures with variations of the Gini coefficient as $F(\mathbf{u})$. Yager (1997) and Ogryczak and Śliwiński (2003) consider the combinations with the Rawlsian maximin criterion using $F(\mathbf{u}) = \min_i\{u_i\}$.

Alpha Fairness. Alpha fairness provides an alternative and perhaps more satisfactory means of combing fairness and efficiency than convex combinations. Alpha fairness regulates the combination with a continuous parameter α, where larger values of α signify a greater emphasis on fairness. It

(Mo and Walrand 2000, Verloop et al. 2010) is represented by a family of functions having the form

$$W_\alpha(\mathbf{u}) = \begin{cases} \dfrac{1}{1-\alpha} \sum_i u_i^{1-\alpha} & \text{for } \alpha \geq 0,\ \alpha \neq 1 \\ \sum_i \log(u_i) & \text{for } \alpha = 1 \end{cases}$$

Alpha fairness characterizes a continuum that stretches from a utilitarian criterion ($\alpha = 0$) to a maximin criterion as $\alpha \to \infty$. Lan et al. (2010) provide an axiomatic treatment of α-fairness in the context of network resource allocation, and Bertsimas et al. (2012) study worst-case fairness/efficiency trade-offs implied by this criterion. The parameter α can be interpreted as quantifying the trade-off as follows: utility u_j must be reduced by $(u_j/u_i)^\alpha$ units to compensate for a unit increase in u_i ($u_i < u_j$) while maintaining constant social welfare. This gives priority to less-advantaged parties, as we desire, with α indicating how much priority. Yet it is not obvious what kind of trade-off, and therefore what value of α, is appropriate for a given application. There is no apparent interpretation of α independent of its role in the α-fairness function.

Threshold Criteria. Williams and Cookson (2000) suggest two ways to combine utilitarian and maximin objectives using threshold criteria for two persons. One, based on an efficiency threshold, begins with a maximin criterion but switches to a utilitarian criterion when the overall utility cost of fairness becomes too great. More explicitly, Williams and Cookson uses a maximin criterion when the two utilities are sufficiently close to each other with $|u_1 - u_2| \leq \Delta$, otherwise it uses a utilitarian criterion. Hooker and Williams (2013) provide an n-person extension for this criterion. Parities with $u_i - u_{min} \leq \Delta$ are treated in solidarity with the worst-off through the maximin criterion, and the remaining greater utilities are treated as themselves. The combined fairness measure is the following,

$$W(\mathbf{u}) = (n-1)\Delta + \sum_{i=1}^{n} \max\{u_i - \Delta, u_{min}\}.$$

It is evident that $\Delta = 0$ corresponds to a purely utilitarian objective and $\Delta = \infty$ to a purely maximin objective.

Williams and Cookson also propose the reverse perspective based on a fairness threshold: a utilitarian criterion is used when $|u_1 - u_2| \leq \Delta$, then a maximin criterion is used since the inequity has become too severe. Following the same argument from Hooker and Williams (2012), this definition can be extended to the n-person case and leads to the combined measure below. The main difference with the previous definition is that, parities with $u_i - u_{min} \leq \Delta$ are now counted as equal to u_{min} and the other utilities are counted as themselves.

$$W(\mathbf{u}) = n\Delta + \sum_{i=1}^{n} \min\{u_i - \Delta, u_{min}\}.$$

Here, $\Delta = 0$ corresponds to a purely maximin objective and $\Delta = \infty$ to a purely utilitarian objective.

In Chapter 2, we illustrate that threshold-based combinations that rely on the maximin notion for fairness are sensitive to the utility level of only the very worst-off party. The equity situation of other disadvantaged parties become irrelevant, so long as their utilities are within Δ of the lowest. As a result, fairness among the disadvantaged plays almost no role in the solution. This situation can be addressed to a great degree by replacing maximin fairness with leximax fairness. We defer details on these measures combining leximax fairness to Chapter 2.

1.1.2 Parity-based Fairness

In the field of machine learning, there have been rising demands for fairness considerations to eliminate prejudice favoritism toward an individual or a group based on their inherent or acquired characteristics. One well-known example that motivates extensive interest in fair ML is the series of research efforts on whether the COMPAS software, supported by a recidivism risk prediction algorithm, is biased against African-Americans (Angwin et al. 2016, Dieterich et al. 2016). The focus of fair ML is primarily on mitigating this kind of bias and ensuring that certain minority groups, often defined by law, receive fair treatment. Among ML frameworks, fairness in supervised learning has been most widely studied. The community has seized upon traditional statistical measures of classification error to detect bias, so that it can be avoided when possible.

In a typical scenario, an ML model is trained to make yes–no decisions as to who receives a certain benefit, such as a mortgage loan, a job interview, parole, and so forth, based on various features they possess. A fairness test compares decisions for a minority or protected group with those for the remainder of the population. Four outcomes are possible for each individual: a true positive (the ML model correctly selects the individual for a benefit), a false positive (it incorrectly selects), a true negative (it correctly rejects), and a false negative (it incorrectly rejects). We will refer to the number of individuals in these four groups, respectively, as TP, FP, TN, and FN. Various metrics involving these four statistics are compared between the minority group and the rest of the population, each yielding a measure of parity between the groups.

We set $d_i = 1$ when individual i should be selected, and $d_i = 0$ otherwise, and $\mathbf{d}_i = 1$ when i is selected and $\mathbf{d}_i = 0$ otherwise. We let N_1 be an index set for individuals in the protected group, and N_0 for those in the remainder of the population. For simplicity of exposition, we consider the minority group as the protected group, and the majority as the remainder. Group fairness measures $W(\mathbf{d})$ are defined directly in terms of the decision vector $\mathbf{d}$. Four of the widely studied definitions are summarized below. We note that fair ML literature also considers other types of definitions, including individual fairness, counterfactual fairness, and more recently, utility-based fairness (Chapter 4 reviews relevant literature).

Demographic parity. The simplest bias metric is based on *demographic parity*, also known as proportional/statistical parity. This metric was introduced by Dwork et al. (2012). It seeks to equalize the fraction of minority individuals selected for benefits and the fraction of majority individuals selected. It is defined by comparing the ratio $(TP + FP)/(TP + FP + TN + FN)$ across the two groups.

The demographic parity is defined as:

$$W(\mathbf{d}) = \frac{1}{|N_1|}\sum_{i \in N_1} d_i - \frac{1}{|N_0|}\sum_{i \in N_0} d_i .$$

A main critic of demographic parity is its strict equality of outcomes. By seeking demographic parity and requiring individuals in both groups receive benefits at equal probabilities, we may discriminate against a minority group that happens to be to be more qualified for benefits than the majority on the average.

Equalized odds. The *equalized odds* metric is based on two related but distinct criteria. One is that the fraction of *qualified* minority persons selected is the same as the fraction of qualified majority persons selected (Hardt et al. 2016). The other is that the fraction of *unqualified* minority persons selected is the same as the fraction of unqualified majority persons selected (Zafar et al. 2017). The former is also known as *equality of opportunity* and is defined by comparing the ratio $TP/(TP+FN)$:

$$W(\mathbf{d}) = \frac{\sum_{i \in N_1} d_i \delta_i}{\sum_{i \in N_1} \delta_i} - \frac{\sum_{i \in N_0} d_i \delta_i}{\sum_{i \in N_0} \delta_i} .$$

The latter criterion is based on the ratio $FP/(FP+TN)$:

$$W(\mathbf{d}) = \frac{\sum_{i \in N_1} (1-\delta_i) d_i}{\sum_{i \in N_1} (1-\delta_i)} - \frac{\sum_{i \in N_0} (1-\delta_i) d_i}{\sum_{i \in N_0} (1-\delta_i)} .$$

Accuracy parity. The two-sided evaluation in equalized odds can be obviated simply by measuring the fraction of predictions that are accurate, which is the ratio $(TP+TN)/(TP+TN+FP+FN)$, namely,

$$W(\mathbf{d}) = \frac{1}{|N_1|}\sum_{i \in N_1} \left(d_i \delta_i + (1-d_i)(1-\delta_i) \right) - \frac{1}{|N_0|}\sum_{i \in N_0} \left(d_i \delta_i + (1-d_i)(1-\delta_i) \right) .$$

Predictive rate parity. When one wishes to compare what fraction individuals selected from each group should have been selected, the relevant measure is *predictive rate parity*. This measure aims to equalize $TP/(TP+FP)$, namely,

$$W(\mathbf{d}) = \frac{\sum_{i \in N_1} d_i \delta_i}{\sum_{i \in N_1} d_i} - \frac{\sum_{i \in N_0} d_i \delta_i}{\sum_{i \in N_0} d_i} .$$

Predictive parity is primarily considered in risk assessment contexts, such as, recidivism prediction (Dieterich et al. 2016, Chouldechova 2017), child maltreatment screening (Chouldechova et al. 2018).

1.2 Outline and Contribution

Chapter 2 studies the trade-off between fairness and ef ciency, an important task in many practical decisions. We propose a principled and practical method for balancing these two criteria in an optimization model. Following an assessment of existing schemes, we de ne a set of social welfare functions (SWFs) that combine Rawlsian leximax fairness and utilitarianism and overcome some of the weaknesses of previous approaches. In particular, we regulate the equity/ef ciency trade-off with a single parameter that has a meaningful interpretation in practical contexts. We formulate the SWFs using mixed integer constraints and sequentially maximize them subject to constraints that de ne the problem at hand. We demonstrate the method on problems of realistic size involving healthcare resource allocation and disaster preparation, with solution times of several seconds at most.

Chapter 3 examines an optimization problem rising from training fair support vector machine (SVM). SVM is a popular supervised learning model that constructs hyperplane(s) for tasks including classi cation and regression. Motivated by the computational advantages of specialized algorithms over general-purpose optimization solvers in training SVM, we propose specialized algorithms to train SVMs with fairness constraints. We focus on fairness constraints that are linear functions of the weight vector of the separating hyperplane in a SVM. Utilizing the structure of the dual formulation of this particular form of fair SVM, we design a coordinate descent type subroutine to extend two dual based standard SVM training algorithms, the dual coordinate descent (DCD) algorithm for SVM with a linear kernel and the sequential minimal optimization (SMO) algorithm for SVM with an arbitrary kernel, to train fair SVMs. We establish convergence properties of these new algorithms. In numerical experiments, our training algorithms train fair SVMs much more ef ciently than an off-the-shelf quadratic program solver. Moreover, compared to training standard SVMs, our algorithms allow fairness constraints to be included with minor runtime increase.

Chapter 4 proposes optimization as a general paradigm for formalizing welfare-based fairness in AI systems. We argue that optimization models allow formulation of a wide range of fairness criteria as social welfare functions, while enabling AI to take advantage of highly advanced solution technology. We highlight that social welfare optimization supports a broad perspective on fairness motivated by distributive justice considerations, as literature has designed social welfare functions capturing various concepts of equity. We formalize in-processing and post-processing integration schemes between social welfare optimization and AI, in particular machine learning and rule-based AI. We then present empirical results from a simulated loan processing example to demonstrate the viability and potentials of the integration strategies. Our optimization-centric integrated decision framework broadens the research scope of welfare-based fairness, and opens up interesting directions to pursue a more comprehensive view of fair decision-making.

Chapter 5 studies the problem of online learning (OL) from revealed preferences: a learner wishes to learn a non-strategic agent's private utility function through observing the agent's utility-maximizing actions in a changing environment. We adopt an online inverse optimization setup, where the learner observes a stream of agent's actions in an online fashion and the learning performance is measured by regret associated with a loss function. We design a new loss function that is convex under relatively mild assumptions. Through establishing that the regret with respect to the new loss bounds the regret with respect to the three classical loss functions used in literature, we provide a exible OL framework that enables a uni ed treatment of loss functions and supports a variety of online convex optimization algorithms. We demonstrate with theoretical and empirical evidence that our framework based on the new loss function (coupled in particular with the online Mirror Descent) has signi cant advantages in terms of regret performance and solution time over other OL algorithms from the literature and bypasses the previous technical assumptions as well.

Chapter 6 investigates the modeling and inferring of people's moral preferences. We consider a setting in which a social planner or policymaker has to make a sequence of decisions regarding the allocation of scarce resources in high-stakes social domains. Our goal is to understand stakeholders' moral judgments regarding such allocation policies. In particular, we evaluate the sensitivity of these judgments to the context/history of allocations and their perceived future impact on various socially salient groups. We propose a mathematical model to capture and infer stakeholders' potentially-dynamic moral preferences. We illustrate our model through small-scale human-subject experiments, which elicit crowd workers' moral judgments regarding scarce medical resource distributions during a hypothetical viral epidemic. We observe that participants' preferences are indeed history- and impact-dependent. Additionally, our preliminary experimental results reveal intriguing patterns speci c to medical resources—a topic that is especially salient in the backdrop of the global covid-19 pandemic.

Chapter 2

Combining Leximax Fairness and Efficiency in a Mathematical Programming Model

2.1 Introduction

[1] Fairness is an important consideration across a wide range of optimization models. It can be a central issue in health care provision, disaster planning, workload allocation, public facility location, telecommunication network management, traffic signal timing, and many other contexts. While it is normally straightforward to formulate an objective function that reflects efficiency or cost, it is not obvious how to express fairness in mathematical form. When both fairness and efficiency are desired, as is typical in practice, there is the additional challenge of mathematically integrating them in a tractable model.

For example, when a natural disaster brings down the electric power grid, crisis managers may dispatch crews to urban areas first in order to restore power to more households quickly, thus maximizing efficiency. Yet this may cause rural areas to experience very long blackouts, which could be seen as unfair. A more satisfactory solution might give some amount of priority to rural customers, but without imposing too much harm on the population as a whole. Similarly, traffic signal timing that minimizes total delay may result in impracticably long wait times for traffic on minor streets that cross a main thoroughfare. Again a balance between equity and efficiency may be desirable. The issue can be especially acute in health care. Expensive treatments or research programs that prolong the life of a relatively few gravely ill patients may divert funds from preventive health measures that would spare thousands the suffering brought by less serious diseases.

In this chapter, we develop a practical and yet principled approach to balancing efficiency and fairness that can be implemented with mixed integer/linear programming (MILP) models. While

[1]This chapter is based on joint work with John Hooker, and has been published in Chen and Hooker (2020, 2022a).

there are many possible measures of fairness, we choose a criterion based ultimately on John Rawls' concept of justice-as-fairness (Rawls 1999). One consequence of the Rawlsian analysis is his famous difference principle, which states roughly that a fair distribution of resources is one that maximizes the welfare of the worst-off. Rawls defends the principle with a social contract argument that can be plausibly extended to lexicographic maximization. That is, the welfare of the worst-off is first maximized subject to resource constraints, then the second worst-off, and so forth. The Rawlsian perspective has been defended by closely reasoned philosophical arguments in a vast literature (Richardson and Weithman 1999, Freeman 2003).

The Rawlsian argument goes roughly as follows. Let's suppose that all concerned parties adopt an agreed-upon social policy in an original position behind a “veil of ignorance” as to their identity. It must be a policy that all parties can rationally accept upon learning who they are. Rawls argues that no rational decision maker will accept a policy in which she is the least advantaged, unless she would have been even worse off under any other policy. A fair outcome should therefore maximize the welfare of the worst-off. The argument can be employed recursively to defend a leximax criterion. Rawls intended his principle to apply only to the design of social institutions, and to pertain only to the distribution of “primary goods,” which are goods that any rational person would want. Yet it can be plausibly extended to distributive justice in general, particularly if it is appropriately combined with an efficiency criterion.

A fundamental question that arises in the integration of equity and efficiency is how to regulate the trade-off between the two. We find in a survey of existing models that it is rarely clear how trade-off parameters can be selected and interpreted in a practical context. However, the modeling scheme of Hooker and Williams (2012) offers a potentially appealing approach to this problem. It governs the trade-off between a Rawlsian maximin and a utilitarian criterion with a single parameter Δ that has the same units as utility and can be related naturally to the problem at hand. The value of Δ is chosen so that parties whose utility is within Δ of the lowest are seen as sufficiently disadvantaged to deserve priority. Larger values of Δ result in greater equity. The model also has a practical mixed integer/linear programming (MILP) formulation.

The Hooker–Williams (H–W) scheme has a serious limitation, however. Because its fairness component is the maximin criterion, the actual utility levels of disadvantaged parties other than the very worst-off have no bearing on social welfare. As a result, the solution can be insensitive to most equity considerations. This outcome is particularly unsatisfactory when resource limitations tightly constrain the benefits available to a few parties, a situation we have found to be common in practice. The H–W model awards what utility it can to the most highly-constrained party, whereupon the welfare of other disadvantaged parties becomes irrelevant, and all solutions become virtually indistinguishable with respect to equity. The fairness criterion plays essentially no role in the determination of an optimum among a potentially large number of outcomes considered equally desirable by the H–W scheme, even for arbitrarily large values of Δ.

2.1.1 Our Approach and Results

A natural way to address this problem is to combine efficiency with a lexicographic criterion rather than a maximin criterion. This allows the utility levels of all disadvantaged parties to factor into social welfare. However, it poses a difficult modeling challenge at both a theoretical and a computational level. We meet the challenge by maximizing a sequence of social welfare functions that, except for the first, are quite different from the single function used in the H–W model. Nonetheless the parameter Δ has a similar interpretation, with increasing Δ corresponding to greater prioritization of the fairness criterion.

We show how to formulate these optimization problems as a set of practical MILP models. These MILP problems have substantially different constraint sets and polyhedral properties than the H–W formulation. We also describe a family of valid inequalities that can be added to tighten the models. We extend these results to the common situation in which utility is distributed to groups rather than individuals, such as organizations, regions, or demographic groups. We conclude this chapter by demonstrating the practical applicability of our approach on a healthcare resource allocation problem and an emergency preparedness problems. The former allows us to compare results with those reported by Hooker and Williams (2012) on the same problem. The latter is a shelter location and assignment problem of realistic size. We find that our approach yields reasonable and nuanced socially optimal solutions for both problems, with computation times ranging from a fraction of a second to 18 seconds for a given Δ.

2.1.2 Basic Definitions

An optimization model for integrating fairness and efficiency can be viewed as maximizing a *social welfare function* (SWF) $F(\mathbf{u})$. The value of the function is interpreted as measuring the desirability of a given distribution $\mathbf{u} = (u_1, \ldots, u_n)$ of utilities, where u_i is the amount of utility allocated to party i. We assume the optimization model has the general form

$$\max_{\mathbf{u}\geq 0} \left\{ F(\mathbf{u}) \mid \mathbf{u} \in U \right\} \tag{2.1}$$

where U is the set of feasible utility vectors. In practice, U is defined by a constraint set that generally contains additional variables. We illustrate realistic models of this kind in Section 6.3.

We define a *leximax* solution of (2.1) with respect to a nondecreasing ordering of utilities rather than a predetermined ordering, since the former is relevant to the Rawlsian criterion. Thus given two feasible solutions $\mathbf{u}, \mathbf{u}' \in U$ of (2.1), we let $i_1, \ldots, i_n$ be any permutation of $1, \ldots, n$ for which $u_{i_1} \leq \cdots \leq u_{i_n}$, and $k_1, \ldots, k_n$ any permutation for which $u'_{k_1} \leq \cdots \leq u'_{k_n}$. Then $\mathbf{u} \in U$ is a leximax solution of (2.1) if for any $\mathbf{u}' \in U$ and any $\ell \in \{1, \ldots, n\}$ such that $u_{i_j} = u'_{k_j}$ for $j = 1, \ldots, \ell - 1$, we have $u_{i_\ell} \geq u'_{k_\ell}$.

Two properties that SWFs can possess will be helpful for assessing the equity measures discussed in the next section. The *Pigou–Dalton* (P–G) condition is frequently used to assess social welfare functions, particularly those that measure equality (Dalton 1920, Moulin 2004). It is satisfied when any utility transfer from a better-off party to a worse-off party increases (or does not reduce) social welfare. Given a utility vector $\mathbf{u}$ with $u_i < u_j$, we say that a *P–G transfer* is a transfer $\epsilon > 0$ of utility from u_j to u_i such that $u_i + \epsilon \leq u_j - \epsilon$. Then $F(\mathbf{u})$ satisfies the P–G condition if for any $\mathbf{u}$, and any $\mathbf{u}'$ that results from a P–G transfer, we have $F(\mathbf{u}') \geq F(\mathbf{u})$.

The *Chateauneuf-Moyes* (C–M) condition is a slightly weaker condition that considers certain utility transfers from a better-off class to a worse-off class rather than from one individual to another. We will refer to these as *C–M transfers*. Chateauneuf and Moyes (2005) suggest that their condition is preferable to the P–G condition for assessing equity measures. Their simplest argument is that while a pairwise P–G transfer reduces inequality between two individuals, it may increase inequality between those individuals and others. A C–M transfer does not incur this problem, because the donor and recipient classes respectively lie completely above and below the rest of the population.

To define the C–M condition formally, we say that a C–M transfer derives $\mathbf{u}'$ from $\mathbf{u}$ when $u_1 \leq \cdots \leq u_n$ as well as $u'_1 \leq \cdots \leq u'_n$, and for some pair of integers ℓ, h with $1 \leq \ell < h \leq n$, we have $u_\ell < u_h$ and

$$\mathbf{u}' = \mathbf{u} + \frac{\epsilon}{\ell}\sum_{i=1}^{\ell} \mathbf{e}_i - \frac{\epsilon}{n-h+1}\sum_{i=h}^{n} \mathbf{e}_i$$

for some $\epsilon > 0$, where $\mathbf{e}_i$ is the ith unit vector. Then $F(\mathbf{u})$ satisfies the C–M condition if $F(\mathbf{u}') \geq F(\mathbf{u})$ for any C–M transfer that derives $\mathbf{u}'$ from $\mathbf{u}$. Any function that satisfies the P–G condition also satisfies the C–M condition.

2.1.3 Related Literature

We now survey some of the primary schemes that have been proposed for combining fairness and efficiency. Each can be formulated as a social welfare function that can be maximized in an optimization model of the form (2.1).

Convex Combinations

The most obvious device for combining fairness and efficiency is a convex combination of the two. This corresponds to a SWF of the form

$$F(\mathbf{u}) = (1-\lambda)\sum_i u_i + \lambda F(\mathbf{u}) \tag{2.2}$$

where $F(\mathbf{u})$ is a fairness measure. A number of functions $F(\mathbf{u})$ have been proposed, such as inequality metrics, the Rawlsian maximin principle, and leximax fairness (Cowell 2000, Jenkins and Van Kerm 2011, Karsu and Morton 2015).

A perennial problem with convex combinations is that it is difficult to interpret λ, particularly since $F(\mathbf{u})$ is typically measured in units other than utility. For example, if we select the widely-used Gini coefficient $G(\mathbf{u})$ as a measure of equity, then we must combine utility with a dimensionless quantity $F(\mathbf{u}) = 1 - G(\mathbf{u})$, where

$$G(\mathbf{u}) = \frac{\sum_{i<j} |u_i - u_j|}{n \sum_i u_i} \tag{2.3}$$

Another difficulty is that fairness measures are almost always nonlinear, which can pose tractability problems.

Eisenhandler and Tzur (2019) use a product rather than a convex combination of utility and $1 - G(\mathbf{u})$, which reduces to an SWF that is easily linearized:

$$F(\mathbf{u}) = \sum_i u_i - \frac{1}{n} \sum_{i<j} |u_j - u_i|$$

Yet we now have a convex combination of total utility and another equality metric (negative mean absolute difference) in which $\lambda = \frac{1}{2}$. One may ask why this particular value of λ is suitable.

Mostajabdaveh et al. (2019) use a linear combination that is equivalent to $\sum_i u_i + \mu(1 - G(\mathbf{u}))\sum_i u_i$, where $\mu \in [0,1]$. This at least combines quantities measured in the same units. Yet we again have the problem of justifying a weight μ. In fact, this combination is equivalent to the convex combination implied by the Eisenhandler and Tzur criterion, except that $\lambda = \mu/(1+2\mu)$ rather than $\frac{1}{2}$.

Since equality is often unsuitable as a fairness measure (Frankfurt 2015, Scanlon 2003), one may wish to use the Rawlsian criterion $F(\mathbf{u}) = \min_i\{u_i\}$. It results in a convex combination of quantities that are measured in the same units, but it is again unclear how to select a suitable value of λ. Note that if we index utilities so that $u_1 \le \cdots \le u_n$, the convex combination becomes simply a weighted sum $u_1 + (1-\lambda)\sum_{i>1} u_i$ that gives somewhat more weight to the lowest utility. It is unclear how much more weight is appropriate.

One might also attempt to formulate a convex combination of efficiency with a leximax rather than a maximin criterion. Yet it is unclear how to capture leximax in a function $F(\mathbf{u})$ when the utilities cannot be ordered by size in advance. Ogryczak and Sliwinski (2006) show how to formulate leximax in an optimization model without pre-ordering, but this requires coefficients that vary enormously in size and can introduce numerical instability. There is also no evident means for incorporating an efficiency criterion into the model.

Alpha Fairness and Kalai-Smorodinsky Fairness

Alpha fairness is a parameterized combination of equity and efficiency that does not rely on a convex combination. It is based on an SWF of the form

$$F_\alpha(\mathbf{u}) = \begin{cases} \dfrac{1}{1-\alpha} \sum_i u_i^{1-\alpha} & \text{for } \alpha \ge 0,\ \alpha \ne 1 \\ \sum_i \log(u_i) & \text{for } \alpha = 1 \end{cases}$$

This SWF satisfies the P–G and therefore the C–M condition for all $\alpha \geq 0$. Larger values of α imply a greater emphasis on equity, with $\alpha = 0$ corresponding to a pure utilitarian criterion and $\alpha = \infty$ to a maximin criterion. Lan et al. (2010) provide an axiomatic treatment, and Bertsimas et al. (2012) study worst-case equity/efficiency trade-offs. An interpretation of α is that utility u_j must be reduced by $(u_j/u_i)^{\alpha}$ units to compensate for a unit increase in u_i ($u_i < u_j$) while maintaining constant social welfare. Yet it is again unclear what kind of trade-off, and therefore what value of α, is appropriate for a given application. There is also the computational issue that $F_{\alpha}(\mathbf{u})$ is nonlinear.

Another issue with alpha fairness is that it can assign equality the same social welfare as arbitrarily extreme inequality. In a 2-player situation, for example, the distribution $\mathbf{u} = (s, s)$ has the same social welfare value as (t, T), where

$$t = \begin{cases} s^2/T & \text{if } \alpha = 1 \\ \left(2s^{1-\alpha} - T^{1-\alpha}\right)^{1/(1-\alpha)} & \text{if } \alpha > 1 \text{ and } 2s^{1-\alpha} > T^{1-\alpha} \end{cases}$$

Thus if we hold social welfare fixed, we have $t \to 0$ as $T \to \infty$ for $\alpha = 1$, and $t \to 2^{1/(1-\alpha)}s$ as $T \to \infty$ for $\alpha > 1$. This means that when $\alpha \geq 1$, alpha fairness can judge an egalitarian solution to be no better than a solution in which one party has arbitrarily more wealth than the other. The same is true of the maximin criterion, of course, since it assigns (s, s) and (s, T) the same social value for arbitrarily large T.

A well-known special case of α-fairness arises when $\alpha = 1$. This results in proportional fairness, which is equivalent to the Nash bargaining solution (Nash 1950). Nash (1950) showed that his bargaining solution for two persons is implied by a set of axioms for utility theory, including a strong and perhaps questionable axiom of cardinal noncomparability across parties (Hooker 2013). Harsanyi (1977), Rubinstein (1982), and Binmore et al. (1986) showed that the Nash solution is the asymptotic outcome of certain rational bargaining procedures, again based on strong assumptions.

Kalai and Smorodinsky (1975) proposed an alternative to the Nash bargaining solution that minimizes each player's relative concession. The approach is defended by Thompson (1994) and is consistent with the contractarian ethical philosophy of Gauthier (1983). Mathematically, the objective is to find the largest scalar β such that $\mathbf{u} = (1-\beta)\mathbf{d} + \beta\mathbf{u}^{max}$ is a feasible utility vector, where each u_i^{max} is the maximum of u_i over all feasible utility vectors $\mathbf{u}$ and each d_i is the starting utility of i. The bargaining solution is the vector $\mathbf{u}$ that maximizes β. This can be interpreted geometrically as the furthest feasible point from the origin on the line segment connecting the origin and $\mathbf{u}^{max}$. It is equivalent to maximizing the SWF

$$F(\mathbf{u}) = \begin{cases} \sum_i u_i, & \text{if } \mathbf{u} = (1-\beta)\mathbf{d} + \beta\mathbf{u}^{max} \text{ for some } \beta \text{ with } 0 \leq \beta \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

The SWF is not only discontinuous but violates the C–M (and therefore the P–G) condition. For example, if we have a 2-person utility distribution $(u_1, u_2) = (\beta u_1^{max}, \beta u_2^{max})$ for some β with $0 < \beta \leq 1$, then a utility transfer that tends to equalize the distribution reduces social welfare.

The K–S scheme provides no parameter for adjusting the equity-efficiency trade-off. While it can be suitable for such applications as wage or price negotiation, it can yield solutions in other contexts that many would consider unjust. For example, it can divert treatment resources from cancer patients to persons suffering from the common cold to provide them the same fraction of their maximum health potential.

Threshold Models

Williams and Cookson (2000) proposed a pair of 2-person SWFs based on a utility or equity threshold. A utility-threshold model uses the maximin criterion unless the sacrifice in total utility exceeds a threshold, in which case it switches to a utilitarian criterion. An equity-threshold model uses a utilitarian criterion unless inequality becomes excessive, when it switches to maximin. Hooker and Williams (2012) extended the utility-threshold model to the n-person criterion described earlier, and McElfresh and Dickerson (2018) proposed a similar scheme based on a leximax rather than maximin criterion. Our aim in the present paper is likewise to combine leximax and efficiency in a threshold model, but we will argue that it offers several advantages relative to the McElfresh and Dickerson approach.

The 2-person SWF implied by Williams and Cookson's utility-threshold model can be formulated

$$F(u_1, u_2) = \begin{cases} u_1 + u_2, & \text{if } |u_1 - u_2| \leq \Delta \\ 2\min\{u_1, u_2\} + \Delta, & \text{otherwise} \end{cases} \tag{2.4}$$

The function is utilitarian when $|u_1 - u_2| \leq \Delta$ and represents a maximin criterion otherwise. Indifference curves (contours) of the SWF are illustrated in Fig. 2.1. The maximin criterion $\min\{u_1, u_2\}$ is modified in (2.4) to obtain continuous contours. We will see that maintaining continuity is a major factor in the design of threshold-based SWFs.

The feasible set in Fig. 2.1 is the portion of the nonnegative quadrant under the curve. It represents all feasible utility outcomes that are permitted by the resource budget and other constraints. The shape of the curve indicates that when party 1's utility reaches a certain point, further improvement requires extraordinary sacrifice by party 2 due to the transfer of resources. The utilitarian solution (black dot in the figure) might therefore be viewed as preferable to the maximin solution (small open circle) and in fact yields slightly more social welfare as indicated by the contours.

Hooker and Williams (2012) extend this social welfare function to n persons as follows:

$$F_1(\mathbf{u}) = (n-1)\Delta + n u_{\langle 1\rangle} + \sum_{i=1}^{n} (u_i - u_{\langle 1\rangle} - \Delta)^+ \tag{2.5}$$

Fig. 2.1 Piecewise linear social welfare contours for 2 persons.

where $(a)^+ = \max\{0, a\}$. Here we adopt the convention that $(u_{\langle 1\rangle}, \ldots, u_{\langle n\rangle})$ is the tuple $(u_1, \ldots, u_n)$ arranged in non-decreasing order. We refer to the function as F_1 because it will be the first in a series of functions $F_1, \ldots, F_n$ we define later. It may be more intuitive to rewrite (2.5) as

$$F_1(\mathbf{u}) = \big(t(\mathbf{u}) - 1\big)\Delta + \sum_{i=1}^{t(u)} u_{\langle 1\rangle} + \sum_{i=t(u)+1}^{n} u_{\langle i\rangle}$$

where $t(\mathbf{u})$ is defined so that $u_{\langle 1\rangle}, \ldots, u_{\langle t(\mathbf{u})\rangle}$ are within Δ of $u_{\langle 1\rangle}$; that is, $u_{\langle i\rangle} - u_{\langle 1\rangle} \le \Delta$ if and only if $i \le t(\mathbf{u})$. We will refer to utilities $u_{\langle 1\rangle}, \ldots, u_{\langle t(\mathbf{u})\rangle}$ as being *in the fair region* and utilities $u_{\langle t(\mathbf{u})+1\rangle}, \ldots, u_{\langle n\rangle}$ as being *in the utilitarian region*. The function $F_1(\mathbf{u})$ therefore has the effect of summing all the utilities, but with the proviso that utilities in the fair region are counted as equal to $u_{\langle 1\rangle}$. The term $(t(\mathbf{u}) - 1)\Delta$ is added to ensure continuity of the function.

The parameter Δ therefore has an interpretation that can be described independently of its role in the SWF. Namely, any party with utility within Δ of the lowest is viewed as disadvantaged and deserving of special consideration. The SWF then defines the special consideration to be an identification of the disadvantaged party with the worst-off party, which is given disproportionate weight in the summation of utilities—namely, weight equal to the number of utilities within Δ of the lowest.

A problem with (2.5), however, is that the actual utility levels of the disadvantaged parties, other than that of the very worst-off, have no effect on the value of the SWF. This is illustrated in the 3-person example of Fig. 2.2, which shows the contours of $F_1(u_1, u_2, u_3)$ with $\Delta = 3$ and u_1 fixed to zero. The SWF is constant in the shaded region, meaning that the utilities allocated to persons 2 and 3 have no effect on social welfare as measured by $F_1(\mathbf{u})$, so long as they remain in the fair region. As

Fig. 2.2 Contours of $\Phi_1(0,u_2,u_3)$. The function is constant in the shaded region.

a result, there are infinitely many utility vectors that maximize social welfare, some of which differ greatly with respect to utilities in the fair region. One can add a tie-breaking term $\epsilon(u_2+u_3)$ to the social welfare function, where $\epsilon > 0$ is small, so as to maximize utility as a secondary objective. Yet this still does not account for equity considerations within the fair region.

To obtain a threshold model that is sensitive to the actual utility levels of all the disadvantaged parties, one might combine utility with a leximax criterion rather than a maximin criterion. McElfresh and Dickerson (2018) propose one method of doing so in the context of kidney exchange. Their method is related to the H–W approach, but it relies on the assumption that the parties can be given a preference ordering in advance. It first maximizes a SWF that combines utilitarian and maximin criteria in a way that treats the most-preferred party as the worst-off. If all optimal solutions of this problem lie in the utilitarian region, a utilitarian criterion is used to select one of the optimal solutions. (Here, a utility vector $\mathbf{u}$ is said to be in the fair region if $\max_i\{u_i\} - \min_i\{u_i\} \le \Delta$, and otherwise in the utilitarian region.) Otherwise a leximax criterion is used for all of the optimal solutions, subject to the preference ordering. If we index the parties in order of decreasing preference, the SWF is

$$\Phi(\mathbf{u}) = \begin{cases} nu_1, & \text{if } |u_i - u_j| \le \Delta \text{ for all } i,j \\ \sum_i u_i + \operatorname{sgn}(u_1 - u_i)\Delta, & \text{otherwise} \end{cases} \tag{2.6}$$

McElfresh and Dickerson state that $\Phi(\mathbf{u})$ has continuous contours, but this is true only for $n = 2$. For a counterexample with $n = 3$, we note that $\Phi(0,0,\Delta+\epsilon) = \epsilon$ and $\Phi(0,\epsilon,\Delta+\epsilon) = 2\epsilon - \Delta$ for arbitrarily small $\epsilon > 0$. The discontinuity of the SWF raises questions regarding its suitability for application, since a slight change in the utility distribution could bring about a large and unexpected change in the measurement of social welfare. We also note that $\Phi(\mathbf{u})$ violates the C–M condition and therefore the P–D condition when $n \geq 3$. For example, a C–M transfer that converts (u_1, u_2, u_3) from $(\epsilon, 0, \Delta+\epsilon)$ to $(\epsilon, \epsilon, \Delta)$ reduces social welfare from $2\epsilon + \Delta$ to ϵ.

This approach has two additional limitations. It is often not possible to pre-specify a preference ranking of parties, as was done in the kidney exchange problem. Another is that the leximax criterion is not used until optimal solutions of the SWF are already obtained, and then applied only to the optimal solutions. We wish to allow the leximax criterion to play a role in evaluating all the possible solutions. These limitations are overcome by our proposal, described in the next section.

An earlier version of our scheme appears in a brief conference paper (Chen and Hooker 2020), which uses somewhat different SWFs.

2.2 Defining the Social Welfare Functions

To combine leximax and utilitarian criteria in a threshold model, we propose to maximize a *sequence* of social welfare functions $F_1(\mathbf{u}), \ldots, F_n(\mathbf{u})$, each of which combines maximin and utilitarian measures. The first function $F_1(\mathbf{u})$ is the H–W function (2.5) defined earlier and is maximized over $\mathbf{u} = (u_1, \ldots, u_n)$ to obtain a value for $u_{\langle 1\rangle}$. Each subsequent function $F_k(\mathbf{u})$ is maximized over $u_{\langle k\rangle}, \ldots, u_{\langle n\rangle}$, while fixing utilities $u_{\langle 1\rangle}, \ldots, u_{\langle k-1\rangle}$ to the values already obtained, and while giving $u_{\langle k\rangle}$ a certain amount of priority. The solution of this maximization problem determines the value of $u_{\langle k\rangle}$.

The process terminates when maximizing $F_k(\mathbf{u})$ yields a value of $u_{\langle k\rangle}$ that lies outside the fair region. At this point, $F_k(\mathbf{u})$ is utilitarian, and utilities $u_{\langle k\rangle}, \ldots, u_{\langle n\rangle}$ are determined simultaneously by maximizing $F_k(\mathbf{u})$ while fixing $u_{\langle 1\rangle}, \ldots, u_{\langle k-1\rangle}$ to the values already obtained. We refer to a utility vector $(u_{\langle 1\rangle}, \ldots, u_{\langle n\rangle})$ that results from this process as *socially optimal*.

We describe this sequential optimization procedure more precisely in Section 2.4, but we must first define and explain the functions $F_k(\mathbf{u})$ for $k \geq 2$. Three main considerations govern the design of these functions and give them a significantly different character than $F_1(\mathbf{u})$.

- The fair region must be viewed as already defined, because $u_{\langle 1\rangle}$ was fixed by maximizing $F_1(\mathbf{u})$.
- The utility $u_{\langle k\rangle}$ should receive less priority as k increases, since the it becomes less disadvantaged relative to the fixed lowest utility $u_{\langle 1\rangle}$.
- It turns out that the priority given $u_{\langle k\rangle}$ cannot depend on the number of utilities in the fair region, as it does for $k = 1$, because this results in an irreducibly discontinuous SWF. We therefore design $F_k(\mathbf{u})$ so that the priority depends only on k.

To develop SWFs that are somewhat analogous to the H–W function $F_1(\mathbf{u})$ while reflecting these considerations, it is helpful to write $F_1(\mathbf{u})$ as

$$F_1(\mathbf{u}) = \tau(\mathbf{u})u_{\langle 1\rangle} + \big(\tau(\mathbf{u}) - 1\big)\Delta + \sum_{i=\tau(\mathbf{u})+1}^{n} u_{\langle i\rangle}$$

The function assigns weight $t(\mathbf{u})$ to utility $u_{\langle 1\rangle}$ and weight 1 to utilities in the utilitarian region. This is algebraically equivalent to the formulation

$$F_1(\mathbf{u}) = nu_{\langle 1\rangle} + (n-1)D + \sum_{i=t(\mathbf{u})+1}^{n} (u_{\langle i\rangle} - u_{\langle 1\rangle} - D)$$

We modify this pattern as follows:

$$F_k(\mathbf{u}) = \begin{cases} \displaystyle\sum_{i=1}^{k} (n-i+1)u_{\langle i\rangle} + \sum_{i=t(\mathbf{u})+1}^{n} (u_{\langle i\rangle} - u_{\langle 1\rangle} - D); & \text{if } t(\mathbf{u}) \geq k \\ \displaystyle\sum_{i=1}^{n} u_{\langle i\rangle}; & \text{if } t(\mathbf{u}) < k \end{cases} \tag{2.7}$$

This SWF assigns weight $n-k+1$ to utility $u_{\langle k\rangle}$ and weight 1 to utilities in the utilitarian region. As desired, the priority given to $u_{\langle k\rangle}$ depends only on k and decreases as k increases. While $F_k(\mathbf{u})$ for $k = 1$ assigns weight to $u_{\langle 1\rangle}$ equal to the number of utilities in the fair region, this weight is reduced by one each time k increases by one, to reflect the fact that one of these utilities is fixed and removed from the optimization problem. We also note that additional multiples of D are not required to ensure continuity of $F_k(\mathbf{u})$ when $k \geq 2$.

Thus as the functions $F_k(\mathbf{u})$ are sequentially maximized for increasing values of k, each utility $u_{\langle k\rangle}$ in the fair region receives priority at some point in the process. This scheme incorporates lexicographic optimization in the sense that the smaller utilities are determined earlier in the sequence, although rather than maximizing $u_{\langle k\rangle}$ in step k, we maximize a SWF that gives priority to $u_{\langle k\rangle}$. Utilitarianism in incorporated because each maximization problem considers total utility as well as fairness.

This process yields a purely utilitarian solution when $D = 0$. For in this case we have $t(\mathbf{u}) = 1$ for all $\mathbf{u}$, and $F_1(\mathbf{u})$ reduces to a utilitarian criterion. The fair region is the single point $u_{\langle 1\rangle}$, and we solve the social welfare problem simply by maximizing $F_1(\mathbf{u})$, which yields a utilitarian solution. At the opposite extreme, when $D \to \infty$, maximizing $F_k(\mathbf{u})$ is equivalent to maximizing $u_{\langle k\rangle}$. The reason is that for sufficiently large D, $t(\mathbf{u}) = n$ for all feasible $\mathbf{u}$, and $F_k(\mathbf{u})$ is $(n-k+1)u_{\langle k\rangle}$ plus a constant for $k = 1,\ldots,n$. (Recall that $u_{\langle 1\rangle},\ldots,u_{\langle k-1\rangle}$ are fixed when $F_k(\mathbf{u})$ is maximized.) Thus by sequentially maximizing $F_1(\mathbf{u}),\ldots,F_n(\mathbf{u})$, we maximize the smallest utility u_i, then maximize the second smallest while holding u_i fixed, and so forth. When there is a tie for the smallest utility, the resulting solution may not be leximax, depending on how the tie is broken, but one of the solutions generated in this manner will be leximax.

Figure 2.3 illustrates how maximizing $F_1(\mathbf{u}),\ldots,F_n(\mathbf{u})$ sequentially is more sensitive to equity than maximizing $F_1(\mathbf{u})$, which has the flat region shown in Fig. 2.2, as noted earlier. Suppose we determine a value for u_1 by maximizing $F_1(\mathbf{u})$, say $u_1 = 0$. Then the function $F_2(\mathbf{u})$ has no flat regions, as is evident in Fig. 2.3, and the solutions in the flat region of Fig. 2.2 are now distinguished.

Fig. 2.3 Contours of $F_2(0,u_2,u_3)$ with $\Delta = 3$ and contour interval 1.

2.3 Properties of the Social Welfare Functions

We now investigate some mathematical properties of the social welfare functions $F_k(\mathbf{u})$. First, we note that the contours in Fig. 2.2 are continuous, and the continuity of $F_k(\mathbf{u})$ can be shown in general.

Theorem 1. The functions $F_k(\mathbf{u})$ are continuous for $k = 1,\ldots,n$.

Proof. To prove continuity of $F_1(\mathbf{u})$, it suffices to show that each term of (2.5) is continuous, because a sum of continuous functions is continuous. The first term of (2.5) is a constant, and the second term is continuous because order statistics are continuous functions. Each term of of the summation is continuous because it is the maximum of two continuous functions. To show that $F_k(\mathbf{u})$ is continuous for $k \geq 2$, it is convenient to write (2.7) as

$$\begin{aligned}F_k(\mathbf{u}) = \sum_{i=1}^{k-1}(n-i+1)u_{\langle i\rangle} &+ (n-k+1)u_{\langle k\rangle} \\ &- (n-k)(u_{\langle k\rangle} - u_{\langle 1\rangle} - \Delta)^+ + \sum_{i=k+1}^{n}(u_{\langle i\rangle} - u_{\langle 1\rangle} - \Delta)^+\end{aligned}$$

which simplifies to

$$\begin{aligned}F_k(\mathbf{u}) = &\sum_{i=1}^{k-1}(n-i+1)u_{\langle i\rangle} \\ &+ (n-k+1)\min\{u_{\langle 1\rangle} + \Delta, u_{\langle k\rangle}\} + \sum_{i=k}^{n}(u_{\langle i\rangle} - u_{\langle 1\rangle} - \Delta)^+\end{aligned} \tag{2.8}$$

Because order statistics are continuous, $u_{\langle k\rangle}$, and $u_{\langle 1\rangle}$ are continuous functions of $\mathbf{u}$. Also $\min\{u_{\langle 1\rangle} + \Delta, u_{\langle k\rangle}\}$ and $(u_{\langle i\rangle} - u_{\langle 1\rangle} - \Delta)^+$ are continuous because they are the minimum or maximum of continuous functions.

The SWFs have a monotonicity property as well. Because $F_1(\mathbf{u})$ can be written

$$F_1(\mathbf{u}) = (n-1)\Delta + \sum_{i=1}^{n} \max\{u_i - \Delta, u_{\langle 1\rangle}\} \tag{2.9}$$

the following is evident on inspection of (2.8) and (2.9).

Theorem 2. $F_1(\mathbf{u})$ is monotone nondecreasing. When $u_{\langle 1\rangle}$ is fixed, $F_k(\mathbf{u})$ is monotone nondecreasing for $k \geq 2$.

We also note that while $F_k(\mathbf{u})$ can violate the P–G condition for $n \geq 3$, it satisfies the C–M condition. A counterexample to the P–G condition is illustrated for $F_1(\mathbf{u})$ in Fig. 2.2, where the utility-preserving transfer from A to B reduces social welfare as measured by $F_1(\mathbf{u})$. However, we have the following result.

Theorem 3. The social welfare functions $F_k(\mathbf{u})$ satisfy the Chateauneuf–Moyes condition for $k = 1,\ldots,n$.

Proof. We wish to show that $F_k(\mathbf{u})$ satisfies the C–M condition for $k = 1,\ldots,n$. Recall that a C–M transfer is a transfer of utility from $\mathbf{u}$ to $\mathbf{u}'$ such that $u_1 \leq \cdots \leq u_n$ as well as $u'_1 \leq \cdots \leq u'_n$, and for some pair of integers ℓ, h with $1 \leq \ell < h \leq n$, we have $u_\ell < u_h$ and

$$\mathbf{u}' = \mathbf{u} + \frac{\epsilon}{\ell}\sum_{i=1}^{\ell} \mathbf{e}_i - \frac{\epsilon}{n-h+1}\sum_{i=h}^{n} \mathbf{e}_i$$

for some $\epsilon > 0$. Then $F(\mathbf{u})$ satisfies the C–M condition if

$$F(\mathbf{u}') \geq F(\mathbf{u}) \tag{2.10}$$

for any C–M transfer from $\mathbf{u}$ to $\mathbf{u}'$.

We first prove the theorem for $k = 1$. We wish to show that (2.10) holds for any C–M transfer from $\mathbf{u}$ to $\mathbf{u}'$. There are three types of C–M transfer, illustrated in Fig. 2.4: (a) $\ell < h \leq t(\mathbf{u})$, (b) $\ell \leq t(\mathbf{u}) < h$, and (c) $t(\mathbf{u}) < \ell < h$. The resulting utility gain by individuals $1,\ldots,\ell$, and loss by individuals $h,\ldots,n$, are indicated in Table 2.1. It is clear on inspection of Fig. 2.4 that the gain is at least ϵ in each case, and the loss never more than ϵ. The C-M condition is therefore satisfied.

Fig. 2.4 Illustration of C–M transfers relevant to $F_1(\mathbf{u})$.

Table 2.1 Verifying the Chateauneuf-Moyes condition for $F_1(\mathbf{u})$

Case	Gain	Loss
(a)	$\frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	$\frac{n-t(\mathbf{u})}{n-h+1}\epsilon < \epsilon$
(b)	$\frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	ϵ
(c)	ϵ	ϵ

We now prove the theorem for $k \geq 2$. It is clear that a C–M transfer satisfies (2.10) when $k > t(\mathbf{u})$, because in this case $F_k(\mathbf{u})$ is simply utilitarian. We therefore need only consider the six types of C–M transfer illustrated in Fig. 2.5, in which $k \leq t(\mathbf{u})$.

It is convenient to write $F_k(\mathbf{u})$ in the following form:

$$F_k(\mathbf{u}) = t(\mathbf{u})u_{\langle 1\rangle} + \sum_{i=2}^{k}(n-i+1)u_{\langle i\rangle} + \sum_{i=t(\mathbf{u})+1}^{n}(u_{\langle i\rangle} - \Delta)$$

The resulting gain by individuals $1,\ldots,\ell$, and loss by individuals $h,\ldots,n$, are indicated in Table 2.2. In cases (b)–(f), it is clear on inspection of Fig. 2.5 that the gain is more than ϵ in each case, and the loss never more than ϵ. In case (a), we note first that the gain can be written

$$\left(n - \frac{\ell-1}{2}\right)\frac{n-t(\mathbf{u})}{\ell}$$

To show that the loss is no greater than the gain, it suffices to show this when $h = \ell+1$, since $h \geq \ell+1$ and the loss is nonincreasing with respect to h. Thus it suffices to show

$$\left(n - \frac{\ell-1}{2}\right)\frac{n-t(\mathbf{u})}{\ell} \geq \frac{1}{n-\ell}\sum_{i=\ell+1}^{k}(n-i+1) + n - t(\mathbf{u})$$

Fig. 2.5 Illustration of C–M transfers relevant to $F_k(\mathbf{u})$, $k \geq 2$.

Table 2.2 Verifying the Chateauneuf-Moyes condition for $F_k(\mathbf{u})$, $k \ge 2$

Case	Gain	Loss
(a)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{\ell}(n-i+1)\right)\epsilon$	$\frac{1}{n-h+1}\left(\sum_{i=h}^{k}(n-i+1)+n-t(\mathbf{u})\right)\epsilon$
(b)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{\ell}(n-i+1)\right)\epsilon \ge \frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	$\frac{n-t(\mathbf{u})}{n-h+1}\epsilon < \epsilon$
(c)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{k}(n-i+1)\right)\epsilon \ge \frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	$\frac{n-t(\mathbf{u})}{n-h+1}\epsilon < \epsilon$
(d)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{\ell}(n-i+1)\right)\epsilon \ge \frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	$\frac{n-h+1}{n-h+1}\epsilon = \epsilon$
(e)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{k}(n-i+1)\right)\epsilon \ge \frac{t(\mathbf{u})}{\ell}\epsilon > \epsilon$	$\frac{n-h+1}{n-h+1}\epsilon = \epsilon$
(f)	$\frac{1}{\ell}\left(t(\mathbf{u})+\sum_{i=2}^{k}(n-i+1)+\ell-t(\mathbf{u})\right)\epsilon \ge \epsilon$	$\frac{n-h+1}{n-h+1}\epsilon = \epsilon$

Since $k \ge t(\mathbf{u})$ and each term of the summation is at most n, it suffices to show

$$n - \frac{\ell-1}{2} - \frac{n-t(\mathbf{u})}{\ell} \ge \frac{t(\mathbf{u}) - \ell(n-\ell)+n-t(\mathbf{u})}{n-\ell}$$

Rearranging, we obtain

$$(n-t(\mathbf{u}))\left(\frac{1}{\ell}+\frac{1}{n-\ell}-1\right) \le \frac{\ell+1}{2} \tag{2.11}$$

This inequality is clearly satisfied when the following is false:

$$\frac{1}{\ell}+\frac{1}{n-\ell} \ge 1 \tag{2.12}$$

We therefore assume (2.12) is true. Since (2.11) is clearly satisfied when $\ell = 1$, we suppose $\ell \ge 2$, in which case (2.12) implies $n < \ell^2/(\ell-1)$. Since $\ell < h \le n$, we can state

$$\ell+1 \le n < \frac{\ell^2}{\ell-1}$$

or $\ell^2 - 1 \le n(\ell-1) < \ell^2$. Since n and ℓ are positive integers, this implies $n = \ell+1$, in which case (2.11) reduces to

$$\frac{\ell+1-t(\mathbf{u})}{\ell} \le \frac{\ell+1}{2}$$

This holds because $t(\mathbf{u}) \le \ell+1$, and the theorem follows.

Finally, we note that unlike maximin and alpha fairness, the functions F_k cannot assign equality and arbitrarily extreme inequality the same social welfare value. For a given egalitarian distribution $\hat{\mathbf{u}} = (s, \ldots, s)$, we have $F_1(\hat{\mathbf{u}}) = ns + (n-1)\Delta$ and $F_k(\hat{\mathbf{u}}) = (n-k+1)s$ for $k \geq 2$. We obtain the same social welfare value, for all k, by setting one utility in $\mathbf{u}$ to $(n-k+1)s+\Delta$ and the other $n-1$ utilities to zero. It follows that the maximum gap between utilities is $(n-k+1)s+\Delta$, rather than arbitrarily large.

2.4 The Sequential Optimization Procedure

The sequential optimization procedure combines leximax and utilitarian criteria by first giving priority to the worst-off party, then somewhat less priority to the second worst-off party, and so forth (the leximax element), but while never demanding excessive sacrifice from the well-off (the utilitarian element). This is accomplished by sequentially maximizing the SWFs $F_1(\mathbf{u}), F_2(\mathbf{u}), \ldots$ to determine the utility level of the worst-off, second worst-off, etc., because the SWFs are designed to adjust the priorities in this manner. The process continues until the utilities of all disadvantaged parties are fixed, whereupon the remaining utilities are determined in a purely utilitarian fashion. The disadvantaged parties are those with utilities in the fair region, which the user defines by setting the parameter Δ.

We first simplify notation by removing the initial constants from $F_k(\mathbf{u})$ for $k \geq 2$, resulting in the SWF

$$\bar{F}_k(\mathbf{u}) = (n-k+1)u_{\langle k \rangle} + \sum_{i=k}^{n} (u_{\langle i \rangle} - u_{\langle 1 \rangle} - \Delta)^+ \qquad (2.13)$$

This obviously has no effect on the optimal solution that results from maximizing the SWF. For convenience, we define $\bar{F}_1(\mathbf{u}) = F_1(\mathbf{u})$.

We next maximize the social welfare functions $\bar{F}_1(\mathbf{u}), \ldots, \bar{F}_n(\mathbf{u})$ sequentially, subject to resource constraints and the condition that the unfixed utilities should be no smaller than the largest utility already fixed. The unfixed u_i with the smallest value in the solution becomes the utility determined by maximizing $\bar{F}_k(\mathbf{u})$.

We indicate resource limits by writing $\mathbf{u} \in U$. In practice, they would be formulated in a MILP model by introducing variables and constraints that specify resource limitations and how resource allocations to individual parties translate to utilities. This will be illustrated in our experiments in Section 6.3.

To state the optimization procedure more precisely, we recursively define a sequence of maximization problems $P_1, \ldots, P_n$, where P_1 maximizes $\bar{F}_1(\mathbf{u})$ subject to $\mathbf{u} \in U$, and P_k for $k = 2, \ldots, n$

is

$$
\begin{aligned}
\max\ & \bar{F}_k(\mathbf{u}) \\
\text{s.t.}\ & u_i \ge \bar{u}_{i_{k-1}},\ i \in I_k \\
& u_{i_j} = \bar{u}_{i_j},\ j = 1,\ldots,k-1 \\
& \mathbf{u} \in U
\end{aligned}
\tag{2.14}
$$

The indices i_j are defined so that u_{i_j} is the utility determined by solving P_j. Thus

$$i_j = \operatorname*{argmin}_{i \in I_j} \{u_i^{[j]}\}$$

where $\mathbf{u}^{[j]}$ is an optimal solution of P_j and $I_j = \{1,\ldots,n\} \setminus \{i_1,\ldots,i_{j-1}\}$. We denote by $\bar{u}_{i_j} = u_{i_j}^{[j]}$ the solution value obtained for u_{i_j} in P_j. We need only solve P_k for $k = 1,\ldots,K+1$, where K is the largest k for which $\bar{u}_{i_k} \le \bar{u}_{i_1} + \Delta$. The solution of the social welfare problem is then

$$u_i^* = \begin{cases} \bar{u}_i & \text{for } i = i_1,\ldots,i_{K-1} \\ u_i^{[K]} & \text{for } i \in I_K \end{cases}$$

As is frequently the case with optimization models, alternate optimal solutions may exist. In particular, there may be multiple utilities u_i with the same minimum value in the solution $\mathbf{u}^{[k]}$ of a given problem P_k. Any of these utilities may be fixed when solving P_{k+1}, possibly (but not necessarily) giving rise to multiple socially optimal solutions. This is illustrated in the example of the next section.

At least one optimal solution of each problem P_k is Pareto optimal over the set U, due to the monotonicity of each $\bar{F}_k(\mathbf{u})$ (Theorem 2). Yet the final socially optimal solution is not guaranteed to be Pareto optimal for an arbitrary feasible set. For example, if a 3-person problem has only two feasible solutions $\mathbf{u} = (1,2,2), (1,3,2)$, then both solutions are socially optimal when $\Delta = 3$, but only one is Pareto optimal. However, we will find that the great majority of socially optimal solutions are Pareto optimal in our experiments. Furthermore, a simple postprocessing step checks whether a socially optimal solution $\mathbf{u}^*$ is Pareto optimal and, if not, converts it to a Pareto optimal solution. We need only solve the optimization problem

$$
\begin{aligned}
\max\ & \sum_{i=1}^{n} u_i \\
\text{s.t.}\ & u_i \ge u_i^*,\ i = 1,\ldots,n \\
& \mathbf{u} \in U
\end{aligned}
\tag{2.15}
$$

If the optimal solution $\hat{\mathbf{u}}$ has $\hat{u}_i > u_i^*$ for some i, then $\mathbf{u}^*$ is not Pareto optimal, but $\hat{\mathbf{u}}$ is. We find in the experiments that the adjustment to achieve Pareto optimality is slight even when it is necessary.

2.5 A Small Example

We illustrate the sequential optimization procedure on an example with 3 utilities, $\mathbf{u} = (u_1, u_2, u_3)$. Suppose there are exactly 5 feasible solutions, shown in the first column of Table 2.3. The remainder of the table displays SWF values $\bar{F}_1(\mathbf{u}), \bar{F}_2(\mathbf{u}), \bar{F}_3(\mathbf{u})$ for various Δ settings. Only $\bar{F}_1(\mathbf{u})$ is shown for $\Delta = 0$, since only P_1 is solved in this case. Recall that for $k \geq 2$, $\bar{F}_k(\mathbf{u})$ is modified from $F_k(\mathbf{u})$ by removing constant terms.

Table 2.3 Values of $\bar{F}_1(\mathbf{u}), \bar{F}_2(\mathbf{u}), \bar{F}_3(\mathbf{u})$ for a small example. Only $\bar{F}_1(\mathbf{u})$ is shown for $\Delta = 0$.

$\mathbf{u}$	$\Delta = 0$	$\Delta = 2$	$\Delta = 5$
$\mathbf{u}^1 = (4, 6, 6)$	16	16,12,6	22,12,6
$\mathbf{u}^2 = (2, 6, 9)$	17	17,19,14	18,14,11
$\mathbf{u}^3 = (1, 1, 14)$	16	18,13,25	21,10,22
$\mathbf{u}^4 = (1, 2, 13)$	16	17,14,23	20,11,20
$\mathbf{u}^5 = (2, 1, 13)$	16	17,14,23	20,11,20

Socially optimal solutions are found as follows.

- $\Delta = 0$: Problem P_1 is purely utilitarian, and $\mathbf{u}^2$ maximizes $\bar{F}_1(\mathbf{u})$. The process stops with socially optimal solution $\mathbf{u}^* = \mathbf{u}^{[1]} = \mathbf{u}^2$ without solving P_2 and P_3, because $u^{[1]}_{i_1} = u^{[1]}_1 = 2$ is in the utilitarian region.

- $\Delta = 2$: Problem P_1 has optimal solution $\mathbf{u}^{[1]} = \mathbf{u}^3$ because $\mathbf{u}^3$ maximizes $\bar{F}_1(\mathbf{u})$. We can select u_1 or u_2 as the utility u_{i_2} fixed by this solution, since both are minimum in $\mathbf{u}^3$. If we select u_1, then the feasible set for P_2 is $\{\mathbf{u}^3, \mathbf{u}^4\}$, and $\mathbf{u}^{[2]} = \mathbf{u}^4$ maximizes $\bar{F}_2(\mathbf{u})$ over this feasible set. This leaves $\mathbf{u}^4$ as the only feasible solution of P_3, and the socially optimal solution is $\mathbf{u}^* = \mathbf{u}^4$. If we select u_2 as the fixed utility, P_2 has the feasible set $\{\mathbf{u}^3, \mathbf{u}^5\}$ and optimal solution $\mathbf{u}^{[2]} = \mathbf{u}^5$, resulting in socially optimal solution $\mathbf{u}^* = \mathbf{u}^5$. Thus $\mathbf{u}^4$ and $\mathbf{u}^5$ are both socially optimal.

- $\Delta = 5$: Problem P_1 has optimal solution $\mathbf{u}^{[3]} = \mathbf{u}^1$. Here $u^{[1]}_{i_1} = u^{[1]}_1 = 4$, and the only feasible solution of P_2 and P_3 is $\mathbf{u}^1$, which is socially optimal.

We highlight some key observations from this example. The procedure returns a utilitarian solution when $\Delta = 0$ and a leximax solution when $\Delta = 5$, while the intermediate value $\Delta = 2$ yields a solution that is neither purely utilitarian nor purely leximax. Unlike the H–W criterion, it is sensitive to the utility of a disadvantaged player other than the very worst-off. The H–W criterion $F_1(\mathbf{u})$ prefers solution $\mathbf{u}^3$ to solutions $\mathbf{u}^4$ and $\mathbf{u}^5$ when $\Delta = 2$, while our method has the opposite preference because it considers the lot of the second-worst-off player. We also prefer solutions $\mathbf{u}^4$ and $\mathbf{u}^5$ to solution $\mathbf{u}^1$ when $\Delta = 2$ because of the large sacrifice $\mathbf{u}^1$ imposes on player 3. When Δ is increased to 5, however, the improved lot of players 1 and 2 in $\mathbf{u}^1$ outweighs this sacrifice due to the greater importance of equity.

2.6 Mixed Integer Programming Model

For practical solution of the optimization problems P_k, we wish to formulate them as MILP models. We drop the resource constraints $u \in U$ from problems $P_1,\ldots,P_n$ to obtain $P_1^0,\ldots,P_n^0$, because we wish to analyze the MILP formulations of the SWFs without the complicating factor of resource constraints. These constraints can later be added to the optimization models before they are solved. In addition, problems $P_1^0,\ldots,P_n^0$ contain innocuous auxiliary constraints that make the problems MILP representable.

The MILP model for P_1^0 follows a different pattern than the models for $P_2^0,\ldots,P_n^0$, and we therefore treat the two cases separately. Problem P_1^0 can be written

$$
\begin{aligned}
&\max z_1 \\
&\text{s.t. } z_1 \le n u_{\langle 1\rangle} + (n-1)\Delta + \sum_{i=2}^{n} \left(u_{\langle i\rangle} - u_{\langle 1\rangle} - \Delta\right)^+ && \text{(a)} \\
&\qquad u_i \ge 0, \text{ all } i && \text{(b)} \\
&\qquad u_i - u_j \le M, \text{ all } i, j && \text{(c)}
\end{aligned}
\tag{2.16}
$$

Constraints (c) ensure MILP representability, as explained in Hooker and Williams (2012), because they imply that the hypograph of (2.16) is a finite union of polyhedra having the same recession cone (Jeroslow 1987). The constraints have no practical import for sufficiently large M, although for theoretical purposes we assume only $M > \Delta$.

The MILP model for P_1^0 can be written as follows:

$$
\begin{aligned}
&\max z_1 \\
&\text{s.t. } z_1 \le (n-1)\Delta + \sum_{i=1}^{n} v_i && \text{(a)} \\
&\qquad u_i - \Delta \le v_i \le u_i - \Delta\delta_i, \; i = 1,\ldots,n && \text{(b)} \\
&\qquad w \le v_i \le w + (M-\Delta)\delta_i, \; i = 1,\ldots,n && \text{(c)} \\
&\qquad u_i \ge 0, \; \delta_i \in \{0,1\}, \; i = 1,\ldots,n
\end{aligned}
\tag{2.17}
$$

The following is proved in Hooker and Williams (2012).

Theorem 4. *Model (2.17) is a correct formulation of P_1^0*

When $k \geq 2$, the expression (2.13) for $\bar{F}_k(\mathbf{u})$ implies that problem P_k^0 can be written

$$
\begin{array}{lll}
\max z_k & & \\
\text{s.t. } z_k \leq (n-k+1)\min\{\bar{u}_{i_1}+\Delta, u_{\langle k\rangle}\} + \sum_{i\in I_k}(u_i-\bar{u}_{i_1}-\Delta)^+ & \text{(a)} & \\
u_i \geq \bar{u}_{i_{k-1}},\; i\in I_k & \text{(b)} & (2.18) \\
u_i - \bar{u}_{i_1} \leq M,\; i \in I_k & \text{(c)} &
\end{array}
$$

The constraints (2.18c) are included to ensure that the problem is MILP representable. Since $\bar{u}_{i_1}$ is a constant, the hypograph is a union of bounded polyhedra whose recession cones consist of the origin only and are therefore identical.

The MILP model for P_k^0 when $k = 2,\ldots,n$ is

$$
\begin{array}{lll}
\max z_k & & \\
\text{s.t. } z_k \leq (n-k+1)s + \sum_{i\in I_k} v_i & \text{(a)} & \\
0 \leq v_i \leq M d_i;\; i\in I_k & \text{(b)} & \\
v_i \leq u_i - \bar{u}_{i_1} - \Delta + M(1-d_i);\; i\in I_k & \text{(c)} & \\
s \leq \bar{u}_{i_1} + \Delta & \text{(d)} & \\
s \leq w & \text{(e)} & \\
w \leq u_i;\; i\in I_k & \text{(f)} & (2.19) \\
u_i \leq w + M(1-e_i);\; i\in I_k & \text{(g)} & \\
\sum_{i\in I_k} e_i = 1 & \text{(h)} & \\
w \geq \bar{u}_{i_{k-1}} & \text{(i)} & \\
u_i - \bar{u}_{i_1} \leq M;\; i\in I_k & \text{(j)} & \\
d_i, e_i \in \{0,1\};\; i\in I_k & &
\end{array}
$$

Theorem 5. Model (2.19) is a correct formulation of P_k^0 for $k = 2,\ldots,n$.

Proof. We first show that given any $(\mathbf{u}, z_k)$ that is feasible for (2.18), where $u_{i_j} = \bar{u}_{i_j}$ for $j = 1,\ldots,k-1$, there exist $\mathbf{v},\mathbf{d},\mathbf{e},w,s$ for which $(\mathbf{u},z_k,\mathbf{v},\mathbf{d},\mathbf{e},w,s)$ is feasible for (2.19). Constraint (2.19j) follows directly from (2.18c). To satisfy the remaining constraints in (2.19), we set

$$(d_i, e_i, v_i) = \begin{cases} (0,0,0), & \text{if } u_i - \bar{u}_{i-1} \le D \text{ and } i \ne k \\ (0,1,0), & \text{if } u_i - \bar{u}_{i-1} \le D \text{ and } i = k \\ (1,0,u_i - \bar{u}_{i-1} - D), & \text{if } u_i - \bar{u}_{i-1} > D \text{ and } i \ne k \\ (1,1,u_i - \bar{u}_{i-1} - D), & \text{if } u_i - \bar{u}_{i-1} > D \text{ and } i = k \end{cases}, \; i \in I_k$$

$$w = u_k$$
$$s = \min\{\bar{u}_{i-1} + D, u_k\} \tag{2.20}$$

where k is an arbitrarily chosen index in I_k such that $u_k = u_{\langle k \rangle}$. It is easily checked that these assignments satisfy constraints (b)–(h). They satisfy (i) because (2.18b) implies that $u_k \ge \bar{u}_{i_k - 1}$. To show they satisfy (2.19a), we note that (2.19a) is implied by (2.18a) because $\min\{\bar{u}_{i-1} + D, u_k\} \ge s$ and $(u_i - \bar{u}_{i-1} - D)^+ \le v_i$ for $i \in I_k$. Since (2.18a) is satisfied by $(\mathbf{u}, z)$, it follows that (2.19a) is satisfied by (2.20).

For the converse, we show that for any $(\mathbf{u}, z_k, \mathbf{v}, \mathbf{d}, \mathbf{e}, w, s)$ that satisfies (2.19), $(\mathbf{u}, z_k)$ satisfies (2.18). Constraint (2.18b) follows from (2.19f) and (2.19i), and (2.18c) is identical to (2.19j). To verify that (2.18a) is satisfied, we let k be the index for which $e_k = 1$, which is unique due to (2.19g). It suffices to show that (2.19a) implies (2.18a) when the remaining constraints of (2.19) are satisfied. For this it suffices to show that

$$s \le \min\{\bar{u}_{i-1} + D, u_k\} \tag{2.21}$$

$$v_i \ge (u_i - \bar{u}_{i-1} - D)^+, \; i \in I_k \tag{2.22}$$

(2.21) follows from (d), (e), and (f) of (2.19). (2.22) follows from (b) and (c) of (2.19). This proves the theorem.

2.7 Valid Inequalities

In this section, we identify some valid inequalities that can strengthen the MILP model for P^0_k for $k \ge 2$. The MILP model (2.17) for P^0_1 is already sharp, meaning that the inequality constraints of the model describe the convex hull of the feasible set, and there is therefore no benefit in adding valid inequalities. The sharpness property may be lost when budget constraints are added, but the resulting model may remain a relatively tight formulation. When $n \le 3$, the models P^0_k for $k \ge 2$ become sharp when the valid inequalities described below are added. This is not true when $n \ge 4$, but the valid inequalities nonetheless tighten the formulation.

The sharpness of the MILP model (2.17) for P^0_1 is proved in Hooker and Williams (2012). We present a simpler proof in Appendix.[2]

Theorem 6. The MILP model (2.17) is a sharp representation of P^0_1 (2.16).

[2] The proof in Hooker and Williams (2012) can be simplified by using only the multipliers $\alpha_i = \frac{M}{nD}\left(a_i - 1 + \frac{D}{M}\right)$ for $i = 1, \ldots, n$, because each $a_i \ge 1 - D/M$. The multipliers b_{ij} in their proof are unnecessary.

We now describe a class of valid inequalities that can be added to the MILP model (2.19) of P'_k for $k \ge 2$ to tighten the formulation.

Theorem 7. The following inequalities are valid for P'_k for $k \ge 2$:

$$z_k \le \sum_{i \in I_k} u_i \tag{2.23}$$

$$z_k \le (n-k+1)u_i + \beta \sum_{j \in I_k \setminus \{i\}} (u_j - \bar{u}_{i_{k-1}}); \; i \in I_k \tag{2.24}$$

where

$$\beta = \frac{M - \Delta}{M - (\bar{u}_{i_{k-1}} - \bar{u}_{i_1})} = \left(1 - \frac{\Delta}{M}\right)\left(1 - \frac{\bar{u}_{i_{k-1}} - \bar{u}_{i_1}}{M}\right)^{-1} \tag{2.25}$$

Proof. It suffices to show that for any $(\mathbf{u}; z_k; \mathbf{v}; \mathbf{d}; \mathbf{e}; w)$ that satisfies (2.19), where $u_{i_j} = \bar{u}_{i_j}$ for $j = 1, \dots, k-1$, the vector $\mathbf{u}$ satisfies (2.23) and (2.24). Since we know from Theorem 5 that $\mathbf{u}$ is feasible in (2.18), it suffices to show that (2.18) implies (2.23) and (2.24). To derive (2.23), we write (2.18a) as

$$z_k \le \sum_{i \in I_k} \left[\min\{\bar{u}_{i_1} + \Delta, u_{\langle k \rangle i}\} + (u_i - \bar{u}_{i_1} - \Delta)^+ \right] \tag{2.26}$$

For any term i in the summation, we consider two cases. If $u_i \le \bar{u}_{i_1} + \Delta$, then $u_{\langle k \rangle i} \le \bar{u}_{i_1} + \Delta$ (because $u_{\langle k \rangle i} \le u_i$), and the term reduces to $u_{\langle k \rangle i}$. If $u_i > \bar{u}_{i_1} + \Delta$, term i becomes

$$\min\{\bar{u}_{i_1} + \Delta, u_{\langle k \rangle i}\} + (u_i - \bar{u}_{i_1} - \Delta) = \min\{0, u_{\langle k \rangle i} - \bar{u}_{i_1} - \Delta\} + u_i \le u_i$$

In either case, term i is less than or equal to u_i, and (2.23) follows.

To establish (2.24), it is enough to show that (2.24) is implied by (2.18) for each $i \in I_k$. We consider the same two cases as before.

Case 1: $u_i - \bar{u}_{i_1} \le \Delta$, which implies $u_{\langle k \rangle i} - \bar{u}_{i_1} \le \Delta$. Since $\mathbf{u}$ satisfies (2.18a), we have

$$z_k \le (n-k+1)u_{\langle k \rangle i} + \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} (u_j - \bar{u}_{i_1} - \Delta) \tag{2.27}$$

It suffices to show that this implies

$$z_k \le (n-k+1)u_i + \beta \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} \le \Delta}} (u_j - \bar{u}_{i_{k-1}}) + \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} (u_j - \bar{u}_{i_{k-1}}) \; ; \tag{2.28}$$

because (2.28) is equivalent to the desired inequality (2.24). But (2.27) implies (2.28) because $u_{\langle k \rangle i} \le u_i$ by definition of $u_{\langle k \rangle i}$, $u_j - \bar{u}_{i_{k-1}} \ge 0$ for all $j \in I_k$ due to (2.18b), and it can be shown that

$$\beta(u_j - \bar{u}_{i_{k-1}}) \ge u_j - \bar{u}_{i_1} - \Delta \tag{2.29}$$

for any $j \in I_k$. To show (2.29), we note that the definition of β implies the following identity:

$$\bar{u}_{i_1} - \bar{u}_{i_{k-1}} + \Delta = (1-\beta)(M + \bar{u}_{i_1} - \bar{u}_{i_{k-1}}).$$

Adding $(1-\beta)\bar{u}_{i_{k-1}}$ to both sides, we obtain

$$\bar{u}_{i_{k-1}} - \beta\bar{u}_{i_{k-1}} + \Delta = (1-\beta)(M + \bar{u}_{i_1}) \ge (1-\beta)u_j, \tag{2.30}$$

where the inequality holds because $M + \bar{u}_{i_1} \ge u_j$ due to (2.18). We obtain (2.29) by rearranging (2.30).

Case 2: $u_i - \bar{u}_{i_1} > \Delta$. It again suffices to show that (2.18) implies (2.28). Due to the case hypothesis, we have from (2.18) that

$$z_k \le (n-k+1)\min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\} + (u_i - \bar{u}_{i_1} - \Delta) + \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} (u_j - \bar{u}_{i_1} - \Delta)^+$$

This can be written

$$\begin{aligned} z_k \le (n-k+1)u_i - (n-k+1)\left(u_i - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\right) \\ + (u_i - \bar{u}_{i_1} - \Delta) + \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} (u_j - \bar{u}_{i_1} - \Delta)^+ \end{aligned}$$

which can be written

$$\begin{aligned} z_k \le (n-k+1)u_i - (n-k)\left(u_i - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\right) \\ - \left(\bar{u}_1 + \Delta - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\right) + \sum_{\substack{j \in I_k \setminus \{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} (u_j - \bar{u}_{i_1} - \Delta)^+ \end{aligned} \tag{2.31}$$

The second term is nonpositive because $u_i > \bar{u}_1 + \Delta$ by the case hypothesis, and $u_i \ge u_{\langle k\rangle}$. The third term is clearly nonpositive. Thus (2.31) implies (2.28) because $u_j - \bar{u}_{i_{k-1}} \ge 0$ and (2.29) holds for $j \in I_k$ as before.

2.8 Modeling Groups of Individuals

In many applications, utility is naturally allocated to groups rather than individuals, where individuals within each group receive an equal allocation. This occurs in the examples of Section 6.3, in which groups correspond to classes of patients with the same disease/prognosis or to neighborhood populations. In other applications, the number of individuals may be too large for practical solution, since problem P_i must be solved for each individual i. In such cases, individuals can typically be grouped into a few classes within which the individual differences are small or irrelevant, thus making the problem tractable and the results easier to digest. We therefore modify the above SWFs to accommodate groups rather than individuals.

We suppose there are n groups of possibly different sizes. We let u_i denote the utility of each individual in group i and s_i the number of individuals in the group.

We begin by deriving $G_1(\mathbf{u})$. Let $u^0_{i'}$ be the utility of individual i', and let u_i be the utility of each individual in group i. There are n^0 individuals and n groups. Let s_i be the size of group i so that

$$n^0 = \sum_{i=1}^{n} s_i \tag{2.32}$$

Then

$$F_1(\mathbf{u}^0) = n^0 u^0_{\langle 1\rangle} + (n^0 - 1)\Delta + \sum_{i'=1}^{n^0} (u^0_{i'} - u^0_{\langle 1\rangle} - \Delta)^+$$

Since $u_{\langle 1\rangle} = u^0_{\langle 1\rangle}$ and group i has size s_i, we have

$$G_1(\mathbf{u}) = \left(\sum_{i=1}^{n} s_i - 1\right)\Delta + \sum_{i=1}^{n} s_i u_{\langle 1\rangle} + \sum_{i=1}^{n} s_i (u_i - u_{\langle 1\rangle} - \Delta)^+ \tag{2.33}$$

This is the formula used in Hooker and Williams (2012). Hooker and Williams prove the following.

Theorem 8. The problem $\hat{P}^0_1$, modified for groups, is equivalent to the MILP model

$$\begin{aligned}
&\max z_1 \\
&\text{s.t. } z_1 \le \left(\sum_{i=1}^{n} s_i - 1\right)\Delta + \sum_{i=1}^{n} s_i v_i \\
&\quad u_i - \Delta \le v_i \le u_i - \Delta\delta_i;\ i = 1,\ldots,n \\
&\quad w \le v_i \le w + (M - \Delta)\delta_i;\ i = 1,\ldots,n \\
&\quad u_i \ge 0;\ \epsilon_i; \delta_i \in \{0,1\};\ i = 1,\ldots,n
\end{aligned} \tag{2.34}$$

We now derive $\bar{G}_k(\mathbf{u})$ for $k \ge 2$. Recall that the SWF for individuals is

$$\bar{F}_{k'}(\mathbf{u}^0) = (n^0 - k' + 1)\min\{u^0_{\langle 1\rangle} + \Delta, u^0_{\langle k'\rangle}\} + \sum_{i'=k'}^{n^0} (u^0_{\langle i'\rangle} - \bar{u}^0_{\langle 1\rangle} - \Delta)^+ \tag{2.35}$$

To obtain $\bar{G}_k(\mathbf{u})$, we again assume the individuals in each group i have the same utility u_i. The first individual in (2.35) that belongs to group k is individual

$$k' = 1 + \sum_{j=1}^{k-1} s_j \tag{2.36}$$

Due to (2.32) and (2.36), the first term on the RHS of (2.35) is

$$\left(n' - 1 - \sum_{j=1}^{k-1} s_{i_j} + 1\right)\min\{u_{h_k i} + D, u_{h_k i}\} = \sum_{i=k}^{n} s_{h_i i}\min\{u_{h_k i} + D, u_{h_k i}\}$$

since all the utilities in a group are the same. Thus we have

$$\bar{G}_k(\mathbf{u}) = \sum_{i=k}^{n} s_{h_i i}\min\{u_{h_1 i} + D, u_{h_k i}\} + \sum_{i=k}^{n} s_{h_i i}(u_{h_i i} - u_{h_1 i} - D)^+ \qquad (2.37)$$

Theorem 9. The functions $\bar{G}_k(\mathbf{u})$ are continuous in $u_{h_k i}, \ldots, u_{h_n i}$ for $k = 1, \ldots, n$.

Proof. It suffices to show each term of (2.37) is a continuous function of $u_{h_k i}, \ldots, u_{h_n i}$, with $u_{h_1 i}, \ldots, u_{h_{k-1} i}$ and the corresponding group sizes $s_{h_1 i}, \ldots, s_{h_{k-1} i}$ fixed. The first term is continuous because it is equal to a constant time the minimum of order statistics $u_{h_1 i}$ and $u_{h_k i}$, which are continuous functions of $\mathbf{u}$. Similarly, each term of the summation is a constant times the maximum of a continuous expression and zero.

The MILP model is very similar to the one we developed for (2.18):

$$\begin{aligned}
&\max z_k \\
&\text{s.t. } z_k \le \sum_{i \in I_k} s_i \sigma + \sum_{i \in I_k} s_i v_i && \text{(a)} \\
&\qquad (2.19b)\text{–}(2.19j) && \text{(b)–(j)} \\
&\qquad d_i, e_i \in \{0, 1\},\ i \in I_k
\end{aligned} \qquad (2.38)$$

Theorem 10. The problem $\bar{P}_k$, reformulated for groups, is equivalent to (2.38) for $k = 2, \ldots, n$.

Proof. We first show that given any $(\mathbf{u}, z_k)$ that is feasible for (2.18), where $u_{i_j} = \bar{u}_{i_j}$ for $j = 1, \ldots, k-1$, there exist $\mathbf{v}, \mathbf{d}, \mathbf{e}, w, \sigma$ for which $(\mathbf{u}, z_k, \mathbf{v}, \mathbf{d}, \mathbf{e}, w, \sigma)$ is feasible for (2.38). Constraint (2.38j) follows directly from (2.18c). To satisfy the remaining constraints in (2.38), we assign values to $\mathbf{v}, \mathbf{d}, \mathbf{e}, w, \sigma$ as in (2.20), where k is an arbitrarily chosen index in I_k such that $u_k = u_{h_k i}$. It is easily checked that these assignments satisfy constraints (2.38b)–(2.38h). They satisfy (2.38i) because (2.18b) implies that $u_k \ge \bar{u}_{i_{k-1}}$. To show they satisfy (2.38a), we note that (2.38a) is implied by (2.18a) because $\min\{\bar{u}_{i_1} + D, u_k\} \ge \sigma$ and $(u_i - \bar{u}_{i_1} - D)^+ \ge v_i$ for $i \in I_k$. Since (2.18a) is satisfied by $(\mathbf{u}, z)$, it follows that (2.38a) is satisfied by (2.20).

For the converse, we show that for any $(\mathbf{u}, z_k, \mathbf{v}, \mathbf{d}, \mathbf{e}, w, \sigma)$ that satisfies (2.38), $(\mathbf{u}, z_k)$ satisfies (2.18). Constraint (2.18b) follows from (2.19f) and (2.19i), and (2.18c) is identical to (2.38j). To verify that (2.18a) is satisfied, we let k be the index for which $e_k = 1$, which is unique due to (2.38g). It suffices to show that (2.38a) implies (2.18a) when the remaining constraints of (2.38) are satisfied. For this it suffices to show that

$$\sigma \le \min\{\bar{u}_{i_1} + D, u_k\} \qquad (2.39)$$

$$v_i \leq (u_i - \bar{u}_{i_1} - D)^+; \; i \in I_k \tag{2.40}$$

(2.39) follows from (d), (e), and (f) of (2.38). (2.40) follows from (b) and (c) of (2.38). This proves the theorem.

Hooker and Williams prove that (2.17) is a sharp representation of P_1^0, and (2.34) a sharp representation of P_1^0 reformulated for groups. We present a simpler proof of both theorems. It is necessary only to prove the latter, because the former is a special case of it.

Theorem 11. The MILP model (2.34) is a sharp representation of P_1^0 reformulated for groups.

of Theorems 6 and 11. We prove Theorem 11, of which Theorem 6 is a special case in which $s_i = 1$ for each i. It suf ces to show that any inequality $z_1 \leq \mathbf{a}^T\mathbf{u} + b$ that is valid for P_1^0 is a surrogate (nonnegative linear combination) of inequalities in (2.34). Let

$$N = \sum_{i=1}^{n} s_i$$

We rst show that the following is a surrogate of (2.34) for any i:

$$z_1 \leq (N-1)D + \left(s_i + (N - s_i)\frac{D}{M}\right)u_i + \left(1 - \frac{D}{M}\right)\sum_{j \neq i} s_j u_j: \tag{2.41}$$

We then show that $z_1 \leq \mathbf{a}^T\mathbf{u} + b$ is a surrogate of the inequalities (2.41). The theorem follows.

To show that (2.41) is a surrogate of (2.34), we rst note that the following is a linear combination of the upper bounds on v_j in (2.34b) and (2.34c), using multipliers D/M and $1/(M - D)$, respectively:

$$v_j \leq \frac{D}{M}w + \left(1 - \frac{D}{M}\right)u_j: \tag{2.42}$$

We also have the following from (2.34b) and (2.34c):

$$v_i \leq u_i: \tag{2.43}$$

$$w \leq v_i: \tag{2.44}$$

We now obtain the following, for any given i and j, as a linear combination of (2.42) and (2.44), using multipliers 1 and D/M, respectively:

$$v_j \leq \frac{D}{M}v_i + \left(1 - \frac{D}{M}\right)u_j: \tag{2.45}$$

Finally, we obtain (2.41) for any given i by summing (2.34a) with multiplier 1, (2.43) with multiplier

$$s_i + (N - s_i)\frac{D}{M}$$

and (2.45) over all $j \neq i$ with multiplier s_j.

It remains to show that $z_1 \le \mathbf{a}^T\mathbf{u} + b$ is a surrogate of (2.41) for $i = 1,\ldots,n$. We first observe that $(\mathbf{u}, z) = (0, (N-1)D)$ is feasible in P_1^0 and must therefore satisfy $z_1 \le \mathbf{a}^T\mathbf{u} + b$, which implies $b \ge (N-1)D$. We can assume without loss of generality that $b = (N-1)D$, since otherwise we can add an appropriate multiple of the valid inequality $0 \le b$ to obtain the desired inequality $z_1 \le \mathbf{a}^T\mathbf{u} + b$. We also note that $(\mathbf{u}, z) = (-M,\ldots,-M, NM + (N-1)D)$ is feasible and must satisfy $z_1 \le \mathbf{a}^T\mathbf{u} + (N-1)D$, which means

$$\sum_{j=1}^{m} a_j \le -N \tag{2.46}$$

Finally, we note that $(\mathbf{u}, z) = (M\mathbf{e}_i, (N-1)D + s_i(M-D))$ is feasible for P_1^0, where $\mathbf{e}_i$ is the ith unit vector. Substituting this into $z_1 \le \mathbf{a}^T\mathbf{u} + (N-1)D$, we obtain

$$a_i \ge \left(1 - \frac{D}{M}\right) s_i \tag{2.47}$$

Due to (2.46), we can suppose without loss of generality that $\sum_{j=1}^{m} a_j = N$, since otherwise we can add appropriate multiples of the valid inequalities $0 \le a_j$ to obtain $z_1 \le \mathbf{a}^T\mathbf{u} + b$.

To obtain $z \le \mathbf{a}^T\mathbf{u} + b$ as a surrogate of (2.41), we sum (2.41) over all j using the multipliers

$$a_i = \frac{M}{ND}\left(a_i - \left(1 - \frac{D}{M}\right) s_i\right) \tag{2.48}$$

for each i. It is easily checked that $\sum_{i=1}^{m} a_i = 1$, so that the linear combination has the form

$$z \le \mathbf{d}^T\mathbf{u} + (N-1)D \tag{2.49}$$

We wish to show that $\mathbf{d} = \mathbf{a}$. Note that

$$d_i = \left(s_i + N\frac{D}{M}\right) a_i + \left(1 - \frac{D}{M}\right) s_i \sum_{j \ne i} a_j$$

Using the fact that $\sum_{j=1}^{m} a_j = 1$, this becomes

$$d_i = N\frac{D}{M} a_i + \left(1 - \frac{D}{M}\right) s_i$$

which immediately reduces to $d_i = a_i$. We conclude that (2.49) is a linear combination of the inequalities (2.41) using multipliers a_i. It remains to show that each a_i is nonnegative, but this follows from (2.47) and (2.48).

Finally, we describe a set of valid inequalities for the MILP model (2.38) for $k \ge 2$.

Theorem 12. The following inequalities are valid for the group problem P_k^0 for $k \ge 2$:

$$z_k \le \sum_{i\in I_k} s_i u_i \tag{2.50}$$

$$z_k \le \sum_{j\in I_k} s_i \, u_j + \beta \sum_{j\in I_k\setminus\{i\}} s_j(u_j - \bar{u}_{i_{k-1}}); \; i \in I_k \tag{2.51}$$

where β is given by (2.25).

Proof. Recall that the group version of P_k^0 for $k \ge 2$:

$$\begin{aligned}
&\max z_k \\
&\text{s.t. } z_k \le \sum_{i\in I_k} s_i \min\{\bar{u}_{i_1} + \Delta, u_{\langle k\rangle}\} + \sum_{i\in I_k} s_i (u_i - \bar{u}_{i_1} - \Delta)^+ && \text{(a)} \\
&\qquad u_i \ge \bar{u}_{i_{k-1}}; \; i \in I_k && \text{(b)} \\
&\qquad u_i - \bar{u}_{i_1} \le M; \; i \in I_k && \text{(c)}
\end{aligned} \tag{2.52}$$

It suffices to show that for any $(\mathbf{u}, z_k, \mathbf{v}, \mathbf{d}, \mathbf{e}, w)$ that satisfies (2.38), where $u_{i_j} = \bar{u}_{i_j}$ for $j = 1, \ldots, k-1$, the vector $\mathbf{u}$ satisfies (2.23) and (2.24). Since we know from Theorem 10 that $\mathbf{u}$ is feasible in (2.52), it suffices to show that (2.52) implies (2.50) and (2.51). To derive (2.50), we write (2.52a) as

$$z_k \le \sum_{i\in I_k} s_i \left(\min\{\bar{u}_{i_1} + \Delta, u_{\langle k\rangle}\} + (u_i - \bar{u}_{i_1} - \Delta)^+ \right) \tag{2.53}$$

For any term i in the summation, we consider two cases. If $u_i \le \bar{u}_{i_1} + \Delta$, then $u_{\langle k\rangle} \le \bar{u}_{i_1} + \Delta$ (because $u_{\langle k\rangle} \le u_i$), and the term reduces to $s_i u_{\langle k\rangle}$. If $u_i > \bar{u}_{i_1} + \Delta$, term i becomes

$$s_i \left(\min\{\bar{u}_{i_1} + \Delta, u_{\langle k\rangle}\} + (u_i - \bar{u}_{i_1} - \Delta) \right) = s_i \left(\min\{0, u_{\langle k\rangle} - \bar{u}_{i_1} - \Delta\} + u_i \right) \le s_i u_i$$

In either case, term i is less than or equal to $s_i u_i$, and (2.50) follows.

To establish (2.51), it is enough to show that (2.51) is implied by (2.52) for each $i \in I_k$. We consider the same two cases as before.

Case 1: $u_i - \bar{u}_{i_1} \le \Delta$, which implies $u_{\langle k\rangle} - \bar{u}_{i_1} \le \Delta$. Since $\mathbf{u}$ satisfies (2.52a), we have

$$z_k \le \sum_{j\in I_k} s_j \, u_{\langle k\rangle} + \sum_{\substack{j\in I_k\setminus\{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} s_j (u_j - \bar{u}_{i_1} - \Delta) \tag{2.54}$$

It suffices to show that this implies

$$z_k \le \sum_{j\in I_k} s_j \, u_i + \beta \sum_{\substack{j\in I_k\setminus\{i\} \\ u_j - \bar{u}_{i_1} \le \Delta}} s_j (u_j - \bar{u}_{i_{k-1}}) + \sum_{\substack{j\in I_k\setminus\{i\} \\ u_j - \bar{u}_{i_1} > \Delta}} s_j (u_j - \bar{u}_{i_{k-1}}) \; ; \tag{2.55}$$

because (2.55) is equivalent to the desired inequality (2.51). But (2.54) implies (2.55) because $u_{\langle k\rangle} \ge u_i$ by definition of $u_{\langle k\rangle}$, $u_j - \bar{u}_{i_{k-1}} \ge 0$ for all $j \in I_k$ due to (2.52b), and (2.12) for all $j \in I_k$.

Case 2: $u_i - \bar{u}_{i_1} > \Delta$. It again suffices to show that (2.52) implies (2.55). Due to the case hypothesis, we have from (2.52a) that

$$z_k \le \sum_{j\in I_k} s_j \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\} + s_i(u_i - \bar{u}_{i_1} - \Delta) + \sum_{\substack{j\in I_k\setminus\{i\}\\ u_j - \bar{u}_{i_1} > \Delta}} s_j(u_j - \bar{u}_{i_1} - \Delta)$$

This can be written

$$\begin{aligned} z_k - \sum_{j\in I_k} s_j u_i \le {} & -\sum_{j\in I_k} s_j \big(u_i - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\big) \\ & + s_i(u_i - \bar{u}_{i_1} - \Delta) + \sum_{\substack{j\in I_k\setminus\{i\}\\ u_j - \bar{u}_{i_1} > \Delta}} s_j(u_j - \bar{u}_{i_1} - \Delta) \end{aligned}$$

which can be written

$$\begin{aligned} z_k - \sum_{j\in I_k} s_j u_i \le {} & -\sum_{j\in I_k\setminus\{i\}} s_j \big(u_i - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\big) \\ & - s_i\big(\bar{u}_1 + \Delta - \min\{\bar{u}_1 + \Delta, u_{\langle k\rangle}\}\big) + \sum_{\substack{j\in I_k\setminus\{i\}\\ u_j - \bar{u}_{i_1} > \Delta}} s_j(u_j - \bar{u}_{i_1} - \Delta) \end{aligned} \tag{2.56}$$

The second term is nonpositive because $u_i \ge \bar{u}_1 + \Delta$ by the case hypothesis, and $u_i \le u_{\langle k\rangle}$. The third term is clearly nonpositive. Thus (2.56) implies (2.55) because $u_j - \bar{u}_{i_{k-1}} \ge 0$ and (2.29) holds for $j \in I_k$ as before.

2.9 Applications

We now implement our approach on a healthcare resource allocation problem and a disaster management problem. We solve all MILP instances using Gurobi 8.1.1 on a desktop PC running Windows 10.

2.9.1 Healthcare Resource Allocation

A proper balance between fairness and efficiency is crucial in the allocation of healthcare resources. Hooker and Williams (2012) study a problem in which treatments are made available to patients on the basis of their disease and prognosis. In discussing this case, we caution that the results we report should not be taken as general recommendations for the allocation of medical resources. They are based on cost and clinical data specific to a particular set of circumstances. We use this example because it allows comparison with the published H–W results on the same problem instance.

Patients are divided into groups based on their disease and prognosis. There is one treatment potentially available to each patient group, and for policy consistency, it is provided to either all or none of the group members. Binary variable y_i is 1 if group i receives the recommended treatment and 0 otherwise. The average utility u_i experienced by members of group i is measured in terms of quality adjusted life years (QALYs). q_i is the net gain in QALYs for a member of group i when receiving the recommended treatment, and a_i is the expected QALYs experienced with medical management without the treatment. Thus

$$u_i = a_i + q_i y_i, \quad i = 1, \ldots, n \tag{2.57}$$

The budget constraint is

$$\sum_i^n s_i c_i y_i \le B \tag{2.58}$$

where s_i is the group size, c_i the cost of treating one patient in group i, and B the total available budget. The budget is set so as to force some hard decisions. The constraints (2.57)–(2.58), along with

(a) Instance cp92 ($n = 50, m = 25$)

(b) Instance cp122 ($n = 50, m = 50$)

Fig. 2.6 Runtime in the shelter allocation example

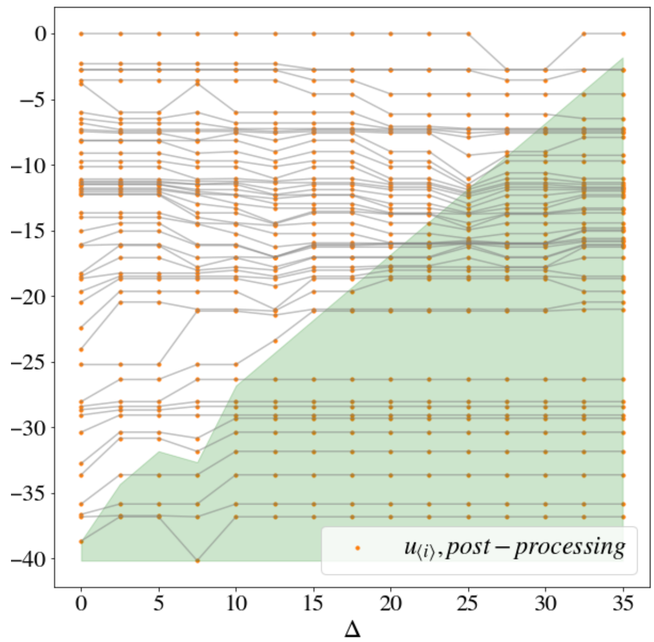


Fig. 2.7 Utility distributions in shelter allocation instance cp92 ($n = 50, m = 25$)

Figs. 2.7 and 2.8 show the evolution of per capita utility in individual neighborhoods as Δ increases. The shaded region indicates which utilities are in the fair region (within Δ of the worst). We see immediately that the problem is highly constrained, because the lowest utilities quickly reach a plateau and remain at a low level even for large Δ values. These neighborhoods are located at a considerable distance from candidate shelter locations, and so they remain disadvantaged even when given high priority. In this type of situation, it is particularly important to use a leximax rather than a maximin criterion of fairness, so as to take into account the situation of disadvantaged neighborhoods other than the very worst-off. If a maximin criterion is used, only the most distant neighborhood is given special priority, and the other disadvantaged neighborhoods benefit only from tie-breaking and the fact that maximizing the worst-off imposes a floor on their utility level. This can be seen in Fig. 2.9,

SVM models, Zafar et al. (2017a, 2019) directly apply solvers for convex and non-convex programs, and Olfat and Aswani (2018) devise an iterative algorithm where each iteration relies on a non-convex program solver. As off-the-shelf solvers are designed to handle general models with only necessary structural assumptions, they may not be efficient for all problem instances, hence these fair SVM training methods may encounter computational issues in complex, large-scale real applications.

3.1.1 Our Approach and Results

An overlooked opportunity in algorithmic development of fair SVM is to build connections with standard SVM training, then make use of the extensive SVM training techniques in literature. In this paper, we examine this opportunity and study whether algorithmic and computational techniques used to train standard SVMs can be extended to train fair SVMs. Our main contribution is that we extend two dual-based SVM training algorithms, the DCD algorithm for linear SVM and the SMO algorithm for general SVM, to handle fair SVM formulations with fairness constraints that are linear in the weights of SVM. The extension utilizes a coordinate descent type subroutine for updating the dual multipliers corresponding to fairness constraints. Moreover, the numerical techniques of shrinking and caching that are effective for improving standard SVM training are valid for the modified algorithms as well. We evaluate the empirical performance of these specialized training methods for fair SVM and demonstrate that their computational costs are comparable with the corresponding standard counterparts. Moreover, they can be more efficient than a state-of-the-art solver on large instances.

SVM Formulations and Notations. We study the standard soft-margin C-SVM formulation for binary classification. Suppose a training set containing n sample points is $D = \{(x_i, y_i, z_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$ are feature vectors, $y_i \in \{-1, 1\}$ is i's true label, and $z_i \in \{0, 1\}$ indicates i's group membership in a protected class (e.g. gender, race). Note that the group features will be used to define fairness constraints. Let $\phi(x_i)$ denote a map of feature x_i and C be the penalty parameter, then (3.1) and (3.2) are respectively the primal and dual SVM formulations.

$$\min_{\theta, b, \xi} \ \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^n \xi_i \quad \text{s.t. } \xi_i \geq 1 - y_i(\langle \theta, \phi(x_i)\rangle + b), \ \xi_i \geq 0 \ \forall i. \tag{3.1}$$

$$\min_{\alpha} \ -e^T\alpha + \frac{1}{2}\alpha^T Q\alpha \quad \text{s.t. } \sum_{i=1}^n \alpha_i y_i = 0, \ 0 \leq \alpha_i \leq C \ \forall i. \tag{3.2}$$

In (3.2), $\alpha \in \mathbb{R}^n$ denotes the vector of dual variables and e is the all 1 vector in $\mathbb{R}^n$. Q is a $n \times n$ matrix with $Q_{i,j} = y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle$ represents the kernel function. For notation ease, we use the abbreviation $K_{ij} = K(x_i, x_j)$. One notable advantage of SVM is that using different kernel functions in (3.2) leads to a wide range of predictive models. For example, two popular options are: linear function kernel $K(x_i, x_j) = \langle x_i, x_j\rangle$, Gaussian radial basis function (RBF) kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2)$.

Suppose α is an optimal dual solution, then the weight θ and bias b in the optimal classifier θ can be computed as follows. We use x_{sv}, y_{sv} to denote a training sample point that is a support vector, namely its corresponding dual optimal solution satisfies $\alpha_{sv} \in (0, C)$:

$$\theta = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i); \quad b = y_{sv} - \sum_{i=1}^{n} \alpha_i y_i K(x_i, x_{sv}). \tag{3.3}$$

A special class of SVM that is often studied separately is the linear SVM, which uses a linear kernel function, or equivalently $\phi(x_i) = x_i$. In this case (3.1) can be simplified by treating the bias term b as an additional dimension in θ. As shown in Hsieh et al. (2008), the dual formulation for linear SVM does not need the equation constraint and the optimal θ, b can be obtained together from the optimal dual solution.

$$\min_{\alpha} \; -e^T\alpha + \frac{1}{2}\alpha^T Q\alpha \quad \text{s.t. } 0 \le \alpha_i \le C \;\; \forall i. \tag{3.4}$$

$$\theta = \sum_{i=1}^{n} \alpha_i y_i x_i; \quad b = \sum_{i=1}^{n} \alpha_i y_i. \tag{3.5}$$

3.1.2 Related Literature

There is a large number of SVM training algorithms in literature. As summarized in the survey paper Shawe-Taylor and Sun (2011), majority of proposed methods solve the dual formulation (3.2) and well-studied methodologies include interior point method, decomposition method, active set method and coordinate descent. We focus on reviewing literature related to decomposition method for general SVM and coordinate descent for linear SVM, as they are more relevant to our study.

Decomposition methods solve the complete dual formulation, a quadratic program (QP), by breaking it down into a series of smaller QP subproblems. This technique is useful for handling the potential computational challenge rising from a large number of training samples and features. Each subproblem optimizes the same objective function but with the restriction that only a subset of the dual variables, typically referred to as *working set*, can be modified. A decomposition method iteratively uses subproblems to search among feasible solutions until convergence to an optimal solution. Two popular software packages for decomposition methods are $\text{SVM}^{\text{light}}$ (Joachims (1998)) and LIBSVM (Chang and Lin (2011)).

Early works such as Osuna et al. (1997a) and Osuna et al. (1997b) consider constant size working sets in decomposition methods. The seminal paper Joachims (1998) formalize the working set selection as an optimization problem and design the $\text{SVM}^{\text{light}}$ algorithm as an efficient implementation of decomposition methods to train SVM with arbitrary kernel function. Later, Platt (1998) propose the sequential minimal optimization (SMO) algorithm, a simpler and faster decomposition method where each working set contains exactly two variables. Follow-up works have been improving SMO algorithm with more efficient working sets selection heuristics, e.g. Fan et al. (2005); Glasmachers et al. (2006); Keerthi et al. (2001); List and Simon (2004); Yang et al. (2019). Some key developments include the first-order information based *Maximum Violating Pair* selection rule from Keerthi et al.

(2001), second-order information based heuristics from Fan et al. (2005); Glasmachers et al. (2006); List and Simon (2004), and more recently Yang et al. (2019) incorporate higher order information to select working sets. The LIBSVM library Chang and Lin (2011) implements SMO as described in Fan et al. (2005) for common SVM formulations and kernel functions.

Parallel with the research on algorithm design, another stream of papers focus on theoretical properties of various decomposition methods for SVM. Keerthi and Gilbert (2002) prove the finite termination of the SMO algorithm stated in Keerthi et al. (2001) which set the stopping criteria as satisfying the KKT conditions associated with (3.1) and (3.2) within up to t violation. Lin (2002b) extend the analysis to general decomposition methods with working sets of size greater than two such as SVM^{light}. As noted in papers including Lin (2001b), these finite termination results are not equivalent to convergence to the optimal solution due to the violation tolerance. Lin (2001b) prove the asymptotic convergence of SVM^{light} given in Joachims (1998). Lin (2002a) extend the proof to show asymptotic convergence of the SMO algorithm in Keerthi et al. (2001). In addition, Lin (2001a) prove the linear convergence of both algorithms Joachims (1998); Keerthi et al. (2001) when the dual objective function is strictly convex, which is possible for some kernel choices such as the Gaussian RBF kernel. A comprehensive convergence analysis of general decomposition methods for SVMs can be found in Chen et al. (2006).

Coordinate descent is another optimization technique that has been applied to train SVMs. Coordinate descent is an iterative algorithm that finds an optimal solution through successive coordinate-wise optimization. Early research on applying coordinate descent methods to the dual SVM formulations can be found in Mangasarian and Musicant (1999, 2001). Later works explore the potentials of this technique in large-scale linear SVMs, for example, Chang et al. (2008) and Hsieh et al. (2008) respectively design coordinate descent methods to solve the primal and the dual linear SVM models. The algorithms from Hsieh et al. (2008) are implemented in the LIBLINEAR (Fan et al. (2008)) library for large-scale linear classification. The convergence analysis can take advantage of relevant results for general descent methods from convex programming. For instance, Hsieh et al. (2008) apply the techniques from Luo and Tseng (1993) to prove the global convergence of their dual coordinate descent algorithm for linear SVM, which implies both asymptotic convergence and finite termination to a desirable accuracy level.

Numerical techniques in the implementation of these fore-mentioned specialized algorithms are useful for further speeding up SVM training. Two best-known techniques are shrinking and caching, both initially proposed in Joachims (1998). They are routinely implemented in decomposition methods (e.g. Fan et al. (2005); Joachims (1998)) and coordinate descent methods (e.g. Hsieh et al. (2008)). The shrinking technique takes advantage of the fact that an optimal dual solution contains a relatively small number of variables not at bounds, which correspond to the support vectors, and the remaining variables are bounded, i.e. $\alpha_i = 0$ or C. To accelerate training, we can identify dual variables that are likely to be bounded at optimality and temporarily remove these variables to reduce the size of the dual problem, namely, we ignore the shrunk elements when choosing working sets. The caching technique refers to storing some recently used K_{ij} in the available memory to reduce the needed number of kernel function evaluations and computation time. This technique is particularly useful for large problems where Q is too large to be fully stored and $Q_{ij} = y_i y_j K_{ij}$ has to be calculated as needed.

Another related line of literature is fair machine learning (ML). Fair ML methods are typically designed by modifying standard ML methods before, during or after the training phase, respectively known as pre-, in-, or post-processing fair ML. Besides the variety in fairness seeking strategies, there is also a large number of fairness definitions, which can be categorized as group versus individual fairness. In the contexts of SVMs, or more generally, classification models, group fairness definitions with respect to the confusion matrix of classification are the most common. We refer to Mehrabi et al. (2019a) for a comprehensive survey of fair ML.

Next, we focus on a specific in-processing approach that is closely related to our study. One can train a fair SVM using the following modified SVM formulation, where $f_l(\theta;b;D)$ are fairness measures with ε_l denoting the respective desirable fairness levels.

$$\min_{\theta;b;\xi} \quad \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n}\xi_i \text{ s.t. } \xi_i \geq 1 - y_i(\langle\theta;\phi(x_i)\rangle + b);\ \xi_i \geq 0\ \forall i;\ f_l(\theta;b;D) \leq \varepsilon_l\ \forall l. \tag{3.6}$$

Olfat and Aswani (2018); Zafar et al. (2017a, 2019) have studied this general model. Zafar et al. (2017a) define a convex proxy for the well-known demographic parity constraint using the covariance associated with the SVM decision boundary. Their constraint ensures (3.6) is a convex program which can be readily solved by commercial solvers in some cases. Donini et al. (2018a) considers SVM as a special case of empirical risk minimization, and applies a similar proxy technique to approximate equalized opportunity constraints. A main motivation for our paper is that even though state-of-the-art solvers are highly efficient for (3.6) with a linear kernel, they tend to suffer noticeable performance drop with a non-linear kernel such as RBF kernel. Specialized fair SVM algorithms, as we demonstrate, could be more robust to kernel choices. Olfat and Aswani (2018) also focus on demographic parity. They note that the convex proxy used in Zafar et al. (2017a) is a relatively weak relaxation of the exact demographic parity definition. They propose a non-convex quadratic constraint as a more accurate proxy, then designed an iterative algorithm to solve (3.6) by solving a sequence of structured convex-concave programs with available solvers. More recently, Zafar et al. (2019) define a continuous proxy for the equalized odds constraints, which require equal false positive rate and false negative rate across groups. The new measure leads (3.6) to the form of a Disciplined Convex-Concave Program, which are still solvable with some existing solvers. These latter two papers are not explicitly comparable with our methods, since they work with a more difficult non-convex fair SVM formulation whereas we aim to design specialized algorithms to solve convex fair SVM formulations.

3.2 Problem Formulation

We use (3.6) as the fair SVM model in this paper. In addition, we focus on a specific form of fairness constraints.

Assumption 1. The fairness measures f_l are functions of $\theta; D$; moreover, they are linear in θ. For notation ease, we denote f_l as: $f_l(\theta; D) := \langle p_l(D), \theta \rangle + c_l$.

Note that $p_l(D)$ is dependent only on the training data D and its definition reflects the desirable notion of fairness. To further simplify notation, we set $p_l = p_l(D)$. Assumption 1 ensures the convexity of (3.6) and makes it convenient to obtain the following Lagrangian dual formulation of fair SVM.

For easy comparison with (3.2), we again use $\mu \in \mathbb{R}^n$ as the dual multipliers for the constraints on ξ_i. Suppose (3.6) contains k fairness constraints, then we let $\gamma \in \mathbb{R}^k$ denote dual multipliers for the fairness constraints, and $d \in \mathbb{R}^k$ represents the constraint parameters with $d_l = \epsilon_l - c_l$ for all $l = 1, \dots, k$. In addition, we use $\bar{Q}$ to denote a matrix defined similarly as Q in the standard SVM case: $\bar{Q} = \begin{bmatrix} Q & -A \\ -A^T & P \end{bmatrix}$, Q is the same as in (3.2) with $Q_{ij} = \langle y_i \phi(x_i), y_j \phi(x_j) \rangle$, A is a $k \times n$ matrix with $A_{li} = \langle p_l, y_i \phi(x_i) \rangle$, and P is a $k \times k$ with $P_{ll'} = \langle p_l, p_{l'} \rangle$.

$$\min_{\mu, \gamma} \; -e^T\mu - d^T\gamma + \frac{1}{2} \begin{bmatrix} \mu \\ \gamma \end{bmatrix}^T \bar{Q} \begin{bmatrix} \mu \\ \gamma \end{bmatrix} \quad \text{s.t} \; \sum_{i=1}^n \mu_i y_i = 0 \,(a); \; 0 \le \mu_i \le C, \; \forall i; \; \gamma_l \ge 0, \; \forall l. \tag{3.7}$$

Proposition 1. Under Assumption 1, (3.7) is the Lagrangian dual of the fair SVM formulation (3.6). In the linear SVM case, namely $\phi(x_i) = x_i$ in (3.6), constraint (a) is not needed in the dual formulation.

Proof. We use $\{\mu_i, \lambda_i, \gamma_l : i \in [n], l \in [k]\}$ to denote the dual multipliers respectively for the three sets of constraints in (3.6). Under Assumption 1, the Lagrangian function for (3.6) is:

$$L(\theta, b, \xi; \mu, \lambda, \gamma) = \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^n \xi_i + \sum_{i=1}^n \mu_i (1 - y_i(\langle \theta, \phi(x_i) \rangle + b) - \xi_i) + \sum_{i=1}^n \lambda_i(-\xi_i) + \sum_{l=1}^k \gamma_l (\langle p_l, \theta \rangle + c_l - \epsilon_l)$$

By definition, the Lagrangian dual is:

$$\max_{\lambda, \mu, \gamma \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{R}^K_+} \; \min_{\theta, b, \xi \in P(\theta, b, \xi)} L(\theta, b, \xi; \mu, \lambda, \gamma);$$

where $P(\theta, b, \xi)$ represents the primal feasible region. We can simplify the Lagrangian dual from the above max-min form to a maximization problem in terms of only the dual variables by taking advantage of the following KKT conditions.

$$\frac{\partial L}{\partial \theta} = 0 \Rightarrow \theta - \sum_{i=1}^{n} \mu_i y_i \phi(x_i) + \sum_{l=1}^{k} \gamma_l \pi_l = 0$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \mu_i y_i = 0$$
$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - \mu_i - \lambda_i = 0 \;\forall i$$

Given a primal optimal solution θ^*, b^*, ξ^* and a dual optimal solution η^*, μ^*, γ^*, they must satisfy these KKT conditions. We use the first condition to replace θ by its equivalent formula in terms of μ, γ, and use the other two conditions to eliminate terms involving b and ξ in the Lagrangian function. In addition, we include the latter two conditions as constraints on μ_i, η_i. The λ_i multipliers can be eliminated by replacing the equations with inequalities. After applying simple algebra to reformulate the objective function, we obtain (3.7) as the Lagrangian dual.

In the case when a linear kernel is used in (3.6), same as the standard linear SVM, we can simplify (3.6) by treating b as an additional dimension in θ, then all the above derivation steps involving b are no longer needed. In particular, we do not need to include the KKT condition corresponding to b to the term, $\sum_{i=1}^{n} \mu_i y_i = 0$, in the dual formulation.

Suppose μ^*, γ^* is an optimal solution to (3.7), the optimal θ^* and b^* for general kernel and linear kernel respectively satisfy (3.8) and (3.9).

$$\theta^* = \sum_{i=1}^{n} \mu_i^* y_i \phi(x_i) - \sum_{l=1}^{k} \gamma_l^* \pi_l;\quad b^* = y_{sv} - \langle \theta^*, \phi(x_{sv}) \rangle: \tag{3.8}$$

$$\theta^* = \sum_{i=1}^{n} \mu_i^* y_i x_i - \sum_{l=1}^{k} \gamma_l^* \pi_l;\quad b^* = \sum_{i=1}^{n} \mu_i^* y_i: \tag{3.9}$$

We observe that (3.7) is a quadratic program (QP) with a very similar structure to (3.1). In addition, the variables μ and γ are separable in the constraints, which plays a key role in the design of our specialized methods. We also note that (3.7) and the optimal formulas for θ^* and b^* simplify to their corresponding forms in standard SVM if all $\gamma = 0$.

3.2.1 Supported Fairness Constraints

We next state two examples of fairness constraints that satisfy Assumption 1. We specify these constraints for binary classification on the training data $D = \{(x_i; y_i; z_i)\}_{i=1}^{n}$. Both examples characterize statistical group fairness notions which seek fairness by eliminating certain disparity among the involved groups. For simplicity of presentation, we consider a single group label $z_i \in \{0; 1\}$ indicating whether i belongs to the protected group of interest, but it is possible to extend the formulations to handle multiple groups by comparing each pair separately.

Among group fairness definitions proposed for fair ML, three well studied notions are *Demographic Parity*, *Equalied Odds* and *Predictive Rate Parity*. In binary classification, exact formulations of these fairness conditions need integer variables to denote the predicted labels $\hat{y}_i \in \{-1, 1\}$. To obtain more tractable constraints, Zafar et al. (2017a) propose the technique of replacing the discrete prediction $\hat{y}_i$ with the continuous decision boundary $\langle \theta, f(x_i) \rangle + b$. They thereby obtain the following *covariance parity* constraint as a convex proxy for demographic parity, i.e., $|P(\hat{y}=1|z=1) - P(\hat{y}=1|z=0)| \le \epsilon$. We denote $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$.

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})(\langle \theta, f(x_i) \rangle + b) &\le \epsilon; \\
\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})(\langle \theta, f(x_i) \rangle + b) &\ge -\epsilon.
\end{aligned}
\tag{3.10}
$$

Both constraints in (3.10) satisfy Assumption 1; more specifically, we have $p_1 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z}) f(x_i)$ and $p_2 = -p_1$.

The same technique applies to *equalized odds*, namely *true positive rate parity*, which requires $|P(\hat{y}=1|z=1, y=1) - P(\hat{y}=1|z=0, y=1)| \le \epsilon$. For notation ease, let $y_i' = \frac{y_i+1}{2}$, $w_i = z_i y_i'$, clearly $y_i', w_i \in \{0, 1\}$. In addition, we define $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i'$, $\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$.

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}(y_i'\bar{w} - \bar{y}w_i)(\langle \theta, f(x_i) \rangle + b) &\le \epsilon; \\
\frac{1}{n}\sum_{i=1}^{n}(y_i'\bar{w} - \bar{y}w_i)(\langle \theta, f(x_i) \rangle + b) &\ge -\epsilon
\end{aligned}
\tag{3.11}
$$

Similar to the covariance parity constraint, we conclude that (3.11) is a pair of constraints, $p_1 = \frac{1}{n}\sum_{i=1}^{n}(y_i'\bar{w} - \bar{y}w_i) f(x_i)$ and $p_2 = -p_1$, satisfying Assumption 1.

We conclude this section by noting two limitations with this substitute technique. First, recent papers Bendekgey and Sudderth (2021); Wu et al. (2019) investigate the theoretical guarantees of these relaxed group fairness constraints and observe that they may be insufficient to provide fairness guarantees for certain ranges of ϵ. Bendekgey and Sudderth (2021) propose an alternative substitute strategy: use a logistic surrogate instead of the simple linear surrogate $\langle \theta, f(x_i) \rangle + b$. With a logistic surrogate, we can no longer derive the exact dual fair SVM formulation, so our dual-based algorithms are not directly applicable. Second, the substitute technique does not generalize to predictive rate parity, which is conditional on the predicted labels instead of the true labels as in demographic parity and equalized odds. Formulating tractable continuous surrogates of predictive rate parity is a challenging open question in literature.

3.3 Fair SVM Training Algorithms

We develop specialized methods to train fair SVM via solving (3.7). As our methods extend dual-based standard SVM algorithms, we begin with a review of specialized algorithms for standard SVM with a general kernel and a linear kernel.

3.3.1 Training Standard SVM

Decomposition Method: General SVM

Decomposition method solves (3.2) by decomposing the complete QP into a series of smaller QP subproblems, and relies on the Karush-Kuhn-Tucker (KKT) conditions to choose working sets, update working set variable values and determine whether an optimal solution is found. We refer to SVM^{light} (Joachims (1998)) as a standard implementation of decomposition method.

SVM^{light} initiates with a feasible solution $\mu^{(0)}$ and maintains a feasible μ throughout the procedure. In iteration t, SVM^{light} examines whether the current solution $\mu^{(t)}$ is optimal based on the KKT conditions of (3.2). The algorithm terminates and returns the optimal solution if one has been found, otherwise it proceeds to the next subproblem. In each subproblem, the complete variable set is split into a *working set*, B, of free variables to be updated, and an *inactive set*, N, of fixed variables to be temporarily treated as constants.

Suppose μ and θ, b are respectively feasible solutions to (3.2) and (3.1), then they are both optimal when they satisfy the following KKT conditions.

$$
\begin{aligned}
\mu_i = 0 \quad &\Rightarrow \quad y_i(\langle \theta, \phi(x_i)\rangle + b) \geq 1 \\
\mu_i \in (0, C) \quad &\Rightarrow \quad y_i(\langle \theta, \phi(x_i)\rangle + b) = 1 \\
\mu_i = C \quad &\Rightarrow \quad y_i(\langle \theta, \phi(x_i)\rangle + b) \leq 1
\end{aligned}
\tag{3.12}
$$

We denote the objective function in (3.2) as $\eta(\mu)$, then the gradient with respect to μ_i is:

$$
\nabla_{\mu_i} \eta(\mu) = \sum_{j=1}^{n} Q_{ij}\mu_j - 1. \tag{3.13}
$$

Using these gradient formulas, an equivalent representation of (3.12) is:

$$
\mu_i > 0 \Rightarrow \nabla_{\mu_i}\eta(\mu) + by_i \leq 0; \quad \mu_i < C \Rightarrow \nabla_i \eta(\mu) + by_i \geq 0 \tag{3.14}
$$

Therefore, a feasible μ is optimal if there exists b such that the KKT conditions (3.14) are satisfied. We can easily check whether a valid b exists by keeping track of the following bounds:

$$
\begin{aligned}
m(\mu) &= \max_{i \in I_{up}(\mu)} -y_i \nabla_{\mu_i}\eta(\mu); \text{ where } I_{up}(\mu) := \{i : \mu_i < C, y_i = 1 \text{ or } \mu_i > 0, y_i = -1\} \\
M(\mu) &= \min_{i \in I_{low}(\mu)} -y_i \nabla_{\mu_i}\eta(\mu); \text{ where } I_{low}(\mu) := \{i : \mu_i < C, y_i = -1 \text{ or } \mu_i > 0, y_i = 1\}
\end{aligned}
\tag{3.15}
$$

Since $y_i \in \{1, -1\}$, (3.14) is satisfied when there exists $m(\mu) \leq b \leq M(\mu)$. In practice, decomposition algorithms typically do not require a perfect fulfillment of (3.14). SVM^{light} considers the following termination criterion allowing up to τ violation to the KKT conditions.

$$
m(\mu) - M(\mu) \leq \tau \tag{3.16}
$$

If (3.16) does not hold at a given m, then KKT conditions (3.14) are violated at some m_i. These indices are the candidates for the next working set, in other words, updating these indices may improve the solution.

In SVM^{light}, working set selection is performed via an optimization problem. (3.17) solves for a feasible direction d that leads to the steepest descent in a first-order approximation of the objective function.

$$\begin{aligned} \min_{d}\ & \nabla_{m} h(m)^T d \quad \text{s.t. } y^T d = 0;\ |\{d_i : d_i \neq 0\}| \leq q \\ & d_i \geq 0 \text{ if } m_i = 0;\ d_i \leq 0 \text{ if } m_i = C;\ -1 \leq d_i \leq 1; \forall i \end{aligned} \tag{3.17}$$

Joachims (1998) show that (3.17) can be solved with a simple iterative procedure. The obtained direction d contains at most q nonzero components. The variables corresponding to these nonzero indices are included in the working set B, namely, $B = \{i : d_i \neq 0\}$ and its size is controlled by the choice of q. After selecting a working set, SVM^{light} proceeds to solve a new subproblem obtained from restricting the changing variables $m_B := \{m_i : i \in B\}$ and fixing $m_N := \{m_j : j \notin B\}$. Note that $\nabla^2_{m_i;m_j} h(m)$ denotes the second order derivative of $h(m)$ with respect to $m_i; m_j$, and we further derive $\nabla^2_{m_i;m_j} h(m) = Q_{ij}$.

$$\begin{aligned} \min_{m_B}\ & \sum_{i \in B} \nabla_{m_i} h(m) m_i + \frac{1}{2} \sum_{i;j \in B} \nabla^2_{m_i;m_j} h(m) m_i m_j \\ \text{s.t. } & m_B^T y_B + m_N^T y_N = 0;\ 0 \leq m_i \leq C; \forall i \in B \end{aligned} \tag{3.18}$$

(3.18) is a convex quadratic program with a small number of variables, so it can be handled by off-the-shelf solvers, for instance, SVM^{light} use a solver based on interior-point method.

To speed up the decomposition algorithm, SVM^{light} implements *caching* to store recently used Q_{ij} in the available memory to reduce the needed number of kernel evaluations, and *shrinking* to reduce the problem size by temporarily neglecting variables that are likely to be bounded (or $m_i = 0$ or C) at optimality. We note that both techniques are broadly applicable to SVM training methods. In addition, SVM^{light} maintains an updated $\nabla_m h(m)$ along with the updates of m to eliminate unnecessary computations. In particular, when an iteration updates the working set variables from m_B to m_B^{new}, we can easily update the gradient at an arbitrary index i:

$$\nabla_{m_i} h(m^{new}) = \nabla_{m_i} h(m) + \sum_{j \in B} (m_j^{new} - m_j) Q_{ij} \tag{3.19}$$

Sequential Minimal Optimization: General SVM

Sequential minimal optimization (SMO) is a special type of decomposition method where all working sets have a fixed size of two. Compared to a generic decomposition method such as SVM^{light}, SMO follows the same main steps but use simpler working set selection and variable update subproblems.

For working set selection in SMO, one approach is to apply (3.17) with $q = 2$. This method is studied in Keerthi et al. (2001) as the *Maximal Violating Pair (MVP)* rule. Recall that a feasible m is not optimal if we cannot find α that satisfies all the KKT conditions in (3.14) at α. A pair of indices $\{i, j : i \in I_{up}(\alpha), j \in I_{low}(\alpha)\}$ leads to violation if $-\nabla_{\alpha_i} h(\alpha) > -\nabla_{\alpha_j} h(\alpha)$. The MVP rule picks the pair with the most violation, namely $i \in I_{up}(\alpha)$ and $j \in I_{low}(\alpha)$ that respectively correspond to $m(\alpha)$ and $M(\alpha)$. Namely, at a given α, the MVP rule selects the following working set:

$$B = \{i, j\}, \text{ where } i = \operatorname{argmax}_{i' \in I_{up}(\alpha)} -y_{i'} \nabla_{\alpha_{i'}} h(\alpha), \; j = \operatorname{argmin}_{j' \in I_{low}(\alpha)} -y_{j'} \nabla_{\alpha_{j'}} h(\alpha). \tag{3.20}$$

After selecting a working set, SMO proceeds to solve the subproblem of the form (3.18), which simplifies to the following QP.

$$\begin{aligned} \min_{\alpha_i, \alpha_j} \; & \begin{bmatrix} 1 - \sum_{l \in N} Q_{il} \alpha_l \\ 1 - \sum_{l \in N} Q_{jl} \alpha_l \end{bmatrix}^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix}^T \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\ \text{s.t. } & \alpha_i y_i + \alpha_j y_j = -\sum_{l \in N} \alpha_l y_l, \; 0 \le \alpha_i, \alpha_j \le C. \end{aligned} \tag{3.21}$$

Since this subproblem is a QP of two variables, its closed form optimal solution is easy to obtain. SMO applies the closed form formulas to update α_i, α_j. The optimal updates

$$\alpha_i^{new} = \begin{cases} 0 & \text{if } \alpha_i < 0 \\ C & \text{if } \alpha_i > C \\ \alpha_i & \text{otherwise} \end{cases}; \quad \alpha_j^{new} = \begin{cases} 0 & \text{if } \alpha_j < 0 \\ C & \text{if } \alpha_j > C \\ \alpha_j & \text{otherwise} \end{cases}$$

are clipped from the following unconstrained optimal solutions α_i, α_j to fit in the $[0, C]$ bounds. Note that $\delta > 0$ is a small constant.

$$\begin{aligned} & \alpha_i = \alpha_i + y_i \frac{b_{ij}}{a_{ij}}; \quad \alpha_j = \alpha_j - y_j \frac{b_{ij}}{a_{ij}}; \\ & \text{where } a_{ij} = \begin{cases} K_{ii} + K_{jj} - 2K_{ij} & \text{if } K_{ii} + K_{jj} - 2K_{ij} > 0 \\ \delta & \text{if } K_{ii} + K_{jj} - 2K_{ij} = 0 \end{cases}; \; b_{ij} = y_j \nabla_{\alpha_j} g(\alpha, \gamma) - y_i \nabla_{\alpha_i} g(\alpha, \gamma). \end{aligned} \tag{3.22}$$

Dual Coordinate Descent Method: Linear SVM

The previous two methods are designed to handle general kernel functions. Among all kernel choices, the linear kernel has been studied separately in literature as the simple structure of linear SVM provides great potentials for large-scale problems. We next review the dual coordinate descent (DCD) algorithm that Hsieh et al. (2008) propose for linear SVMs. Note that this is one of the dual-based algorithms implemented in the LIBLINEAR (Fan et al. (2008)) package.

Coordinate descent is a well-known optimization technique that finds an optimal solution via coordinate-wise optimization, namely, it successively updates one variable at a time via a single-variable subproblem until reaching certain termination criteria. The DCD method in Hsieh et al. (2008) essentially apply this standard technique to solve (3.4). The algorithm initiates with a feasible

$\mu^{(0)}$. An iteration of updating $\mu^{(t-1)}$ to $\mu^{(t)}$ consists of n inner iterations, each of which solves a single-variable optimization model to find the optimal update along each coordinate.

Let $\mu^{(i;t-1)}$ denote the solution obtained in the i-th inner iteration, then $\mu^{(0;t-1)} = \mu^{(t-1)}$, $\mu^{(i;t-1)} = [\mu_1^{(t)}, \dots, \mu_i^{(t)}, \mu_{i+1}^{(t-1)}, \dots, \mu_n^{(t-1)}]$ for $i = 1, \dots, n-1$, and $\mu^{(n;t-1)} = \mu^{(t)}$. The i-th inner iteration solves the following optimization model to update $\mu^{(i;t-1)}$ to $\mu^{(i+1;t-1)}$:

$$\min_d \ \eta(\mu^{(i;t-1)} + d e_i) \ \ s.t.: 0 \le \mu_i^{(t-1)} + d \le C \tag{3.23}$$

Here e_i is the n-dimensional unit vector with $(e_i)_i = 1$. (3.23) thereby finds the optimal feasible update along the direction of μ_i. We further evaluate η to simplify (3.23) to a quadratic program:

$$\min_d \ \frac{1}{2} Q_{ii} d^2 + \nabla_{\mu_i} \eta(\mu^{(i;t-1)}) d \ \ s.t.: 0 \le \mu_i^{(t-1)} + d \le C \tag{3.24}$$

These update subproblems also provide a convenient way to check optimality: $\mu^{(t)}$ is optimal if all n subproblems have an optimal solution $d = 0$, namely, no further improvement is possible along any coordinate. We note that $d = 0$ is an optimal solution to (3.24) if and only if the projected gradient $\nabla^P_{\mu_i} \eta(\mu) = 0$ for all i, where,

$$\nabla^P_{\mu_i} \eta(\mu) = \begin{cases} \nabla_{\mu_i} \eta(\mu) & \text{if } 0 < \mu_i < C \\ \min\{0, \nabla_{\mu_i} \eta(\mu)\} & \text{if } \mu_i = 0 \\ \max\{0, \nabla_{\mu_i} \eta(\mu)\} & \text{if } \mu_i = C \end{cases} \tag{3.25}$$

Similar to decomposition methods, instead of seeking the exact optimality, Hsieh et al. (2008) use a relaxed stopping criteria which requires $\max_{\mu_i}\{\nabla^P_{\mu_i} \eta(\mu)\} - \min_{\mu_i}\{\nabla^P_{\mu_i} \eta(\mu)\} < \tau$ and $\max_{\mu_i}\{|\nabla^P_{\mu_i} \eta(\mu)|\} < \tau$, $\min_{\mu_i}\{|\nabla^P_{\mu_i} \eta(\mu)|\} < \tau$. Another implementation detail is the shrinking procedure: Hsieh et al. (2008) demonstrate that the same shrinking techniques used in decomposition methods are valid for the DCD algorithm.

3.3.2 Training Fair SVM

We next develop algorithms for training fair SVM with the dual formulation (3.7). The key challenge of solving (3.7) is that we need to search the optimal values of two sets of variables, μ and γ. We observe that μ and γ are fully separable in the constraints, and they share the same functional format in the objective function. Given a fixed γ, (3.7) apparently simplifies to the dual formulations (3.2) or (3.4), used in standard SVM. Given a fixed μ, (3.7) becomes a convex quadratic program with only non-negativity constraints on the γ variables: $\min_\gamma \ -d^T\gamma + (-A\mu)\gamma + \frac{1}{2}\gamma^T P\gamma + \text{constant} : \gamma_l \ge 0, \forall l$. Note that a quadratic program of this form is solvable with the coordinate descent method (Luo and Tseng (1993)). In fact, the dual formulation (3.4) of standard linear SVM has the same format, which allows Hsieh et al. (2008) to design the dual coordinate descent method.

Drawing motivation from these observations, we propose to solve (3.7) with separate treatment of μ and γ updates; more explicitly, we apply the update subroutine from a standard SVM algorithm to update μ and we design a coordinate descent based update subroutine for γ variables. Utilizing the new γ update subroutine, we extend SMO and DCD to specialized algorithms, respectively for fair SVMs with a general kernel and a linear kernel. We note that the same extension is also applicable to generic decomposition methods.

Main Steps and Termination Criteria

Same as standard SVM algorithms, our fair SVM algorithms initiate with a feasible solution $(\mu^{(0)};\gamma^{(0)})$. In iteration t, we examine whether the current solution satisfies the termination criteria. If the termination criteria are not reached yet, we first apply an update iteration from a standard SVM algorithm (Section 3.3.2) to update $\mu^{(t-1)}$ to $\mu^{(t)}$ with $\gamma^{(t-1)}$ treated as fix constants, then we update $\gamma^{(t-1)}$ to $\gamma^{(t)}$ with $\mu^{(t)}$ fixed using coordinate descent (Section 3.3.2).

We rely on the KKT conditions to select the stopping criteria. Let $g(\mu;\gamma)$ denote the objective function in (3.7).

Proposition 2. The KKT conditions for fair SVMs defined in (3.6), (3.7) are:

$$\begin{aligned} &\mu_i > 0 \Rightarrow \nabla_{\mu_i} g(\mu;\gamma) + b y_i \le 0; \quad \mu_i < C \Rightarrow \nabla_{\mu_i} g(\mu;\gamma) + b y_i \ge 0; \\ &\gamma_l = 0 \Rightarrow \nabla_{\gamma_l} g(\mu;\gamma) \ge 0; \quad \gamma_l > 0 \Rightarrow \nabla_{\gamma_l} g(\mu;\gamma) = 0; \end{aligned} \tag{3.26}$$

where the gradient formulas are:

$$\nabla_{\mu_i} g(\mu;\gamma) = \sum_{j=1}^{n} Q_{ij}\mu_j + \sum_{l=1}^{k} A_{li}\gamma_l - 1; \quad \nabla_{\gamma_l} g(\mu;\gamma) = \sum_{j=1}^{n} A_{lj}\mu_j + \sum_{l'=1}^{k} P_{ll'}\gamma_{l'} - (c_l - e_l). \tag{3.27}$$

Proof. We first derive the gradient formulas for the dual objective function $g(\mu;\gamma)$. Recall that $g(\mu;\gamma) = -e^T\mu - d^T\gamma + \frac{1}{2}\begin{bmatrix}\mu\\ \gamma\end{bmatrix}^T \bar{Q} \begin{bmatrix}\mu\\ \gamma\end{bmatrix}$, where $\bar{Q} = \begin{bmatrix} Q & A \\ A^T & P\end{bmatrix}$: Q is the same $n \times n$ matrix used in the standard dual SVM with $Q_{ij} = \langle y_i f(x_i); y_j f(x_j)\rangle$, A is a $k \times n$ matrix with $A_{li} = \langle p_l; y_i f(x_i)\rangle$, and P is a $k \times k$ with $P_{ll'} = \langle p_l; p_{l'}\rangle$. It is easy to compute the gradients as follows, which are exactly (3.27).

$$\nabla_{\mu_i} g(\mu;\gamma) = \sum_{j=1}^{n} Q_{ij}\mu_j + \sum_{l=1}^{k} A_{li}\gamma_l - 1; \quad \nabla_{\gamma_l} g(\mu;\gamma) = \sum_{j=1}^{n} A_{lj}\mu_j + \sum_{l'=1}^{k} P_{ll'}\gamma_{l'} - (c_l - e_l).$$

These gradients have the following equivalent representations when we plug in the formulas for the matrix entries.

$$\nabla_{\mu_i} g(\mu,\gamma) = \langle y_i \phi(x_i), \textstyle\sum_{j=1}^{n} \mu_j y_j \phi(x_j)\rangle - \langle y_i \phi(x_i), \textstyle\sum_{l=1}^{k} \gamma_l p_l\rangle - 1;$$
$$\nabla_{\gamma_l} g(\mu,\gamma) = -\langle \textstyle\sum_{j=1}^{n} \mu_j y_j \phi(x_j), p_l\rangle + \langle p_l, \textstyle\sum_{l'=1}^{k} \gamma_{l'} p_{l'}\rangle - (c_l - \epsilon_l).$$

Suppose θ, b and μ, γ are respectively the optimal solution to (3.6) and (3.7), then the KKT conditions of this primal-dual pair are:

$$\mu_i > 0 \Rightarrow y_i(\langle \theta, \phi(x_i)\rangle + b) \le 1; \quad \mu_i < C \Rightarrow y_i(\langle \theta, \phi(x_i)\rangle + b) \ge 1;$$
$$\gamma_l = 0 \Rightarrow \langle p_l, \theta\rangle + c_l < \epsilon_l; \quad \gamma_l > 0 \Rightarrow \langle p_l, \theta\rangle + c_l = \epsilon_l;$$

We recall from (3.8) that the optimal solutions satisfy $\theta = \sum_{i=1}^{n} \mu_i y_i \phi(x_i) - \sum_{l=1}^{k} \gamma_l p_l$. When we replace θ in the above KKT conditions with this optimal format in terms of dual variables, we obtain the format in (3.26).

$$\mu_i > 0 \Rightarrow y_i(\langle \textstyle\sum_{i=1}^{n} \mu_i y_i \phi(x_i) - \sum_{l=1}^{k} \gamma_l p_l, \phi(x_i)\rangle + b) \le 1 \Leftrightarrow \nabla_{\mu_i} g(\mu,\gamma) + b y_i \le 0;$$
$$\mu_i < C \Rightarrow y_i(\langle \textstyle\sum_{i=1}^{n} \mu_i y_i \phi(x_i) - \sum_{l=1}^{k} \gamma_l p_l, \phi(x_i)\rangle + b) \ge 1 \Leftrightarrow \nabla_{\mu_i} g(\mu,\gamma) + b y_i \ge 0;$$
$$\gamma_l = 0 \Rightarrow \langle p_l, \textstyle\sum_{i=1}^{n} \mu_i y_i \phi(x_i) - \sum_{l'=1}^{k} \gamma_{l'} p_{l'}\rangle + c_l < \epsilon_l \Leftrightarrow \nabla_{\gamma_l} g(\mu,\gamma) > 0;$$
$$\gamma_l > 0 \Rightarrow \langle p_l, \textstyle\sum_{i=1}^{n} \mu_i y_i \phi(x_i) - \sum_{l'=1}^{k} \gamma_{l'} p_{l'}\rangle + c_l = \epsilon_l \Leftrightarrow \nabla_{\gamma_l} g(\mu,\gamma) = 0.$$

In fair SVM algorithms, we apply the stopping criteria from standard SVM algorithms with the only change that $\nabla g(\mu,\gamma)$ replaces $\nabla h(\mu)$. Suppose we allow up to τ violation to the KKT conditions. For training a fair SVM with a general kernel, we terminate the fair SMO algorithm when the obtained solution (μ,γ) satisfies $m(\mu,\gamma) - M(\mu,\gamma) \le \tau$ where

$$m(\mu,\gamma) = \max_{i \in I_{up}(\mu)} -y_i \nabla_{\mu_i} g(\mu,\gamma); \quad M(\mu,\gamma) = \max_{i \in I_{low}(\mu)} -y_i \nabla_{\mu_i} g(\mu,\gamma). \tag{3.28}$$

For a linear kernel, we can further simplify (3.26) to eliminate the terms including b through the augmentation trick, that is, we add an additional element equal to 1 to x_i for all i and an additional zero element to p_l for all l. Fair linear SVM algorithm thereby uses a similar stopping condition as the DCD method in Section 3.3.1, that is, $\max_i\{\nabla^P_{\mu_i} g(\mu,\gamma)\} - \min_i\{\nabla^P_{\mu_i} g(\mu,\gamma)\} \le \tau$, $\max_i\{|\nabla^P_{\mu_i} g(\mu,\gamma)|\} < \tau$, $\min_i\{|\nabla^P_{\mu_i} g(\mu,\gamma)|\} \le \tau$, where the projected gradient $\nabla^P_{\mu_i} g(\mu,\gamma)$ has an analogous definition to $\nabla^P_i h(\mu)$.

Update Dual Variables for Standard SVM Constraints

In iteration t, we first update $\alpha^{(t)}$ to $\alpha^{(t+1)}$ with $\gamma^{(t)}$ fixed. The update requires solving a subproblem containing only the α variables. Similar to the stopping criteria, compared to the subproblems used in standard SVM algorithms, the only change needed is to substitute the gradient definitions with $\nabla g(\alpha, \gamma^{(t-1)})$.

For fair SVMs with a general kernel, we modify the update step in the SMO algorithm. In each iteration, we choose the working set B from α variables by solving (3.17) with $\nabla_i h(\alpha)$ replaced by $\nabla_{\alpha_i} g(\alpha, \gamma)$. It suffices to use the following modified maximal violation pair rule.

$$B = \{i, j\}, \text{ where } i = \operatorname{argmax}_{i' \in I_{up}(\alpha)} -y_{i'} \nabla_{\alpha_{i'}} g(\alpha, \gamma), \; j = \operatorname{argmin}_{j' \in I_{low}(\alpha)} -y_{j'} \nabla_{\alpha_{j'}} g(\alpha, \gamma). \tag{3.29}$$

Then we solve for the optimal update with the following subproblem.

$$\begin{aligned} \min_{\alpha_B} \; & \sum_{i \in B} \nabla_{\alpha_i} g(\alpha, \gamma) \alpha_i + \frac{1}{2} \sum_{i,j \in B} \nabla^2_{\alpha_i, \alpha_j} g(\alpha, \gamma) \alpha_i \alpha_j \\ \text{s.t. } & \alpha_B^T y_B + \alpha_N^T y_N = 0, \; 0 \le \alpha_i \le C, \forall i \in B \end{aligned} \tag{3.30}$$

For fair linear SVMs, we utilize the DCD method: the outer iteration updating $\alpha^{(t-1)}$ to $\alpha^{(t)}$ consists of n coordinate-wise optimization subproblems. The i-th subproblem is stated below.

$$\min_d \frac{1}{2} Q_{ii} d^2 + \nabla_{\alpha_i} g(\alpha^{(i,t-1)}, \gamma^{(t-1)}) d \;\; \text{s.t.} \; 0 \le \alpha_i^{(t-1)} + d \le C. \tag{3.31}$$

Update Dual Variables for Fairness Constraints

We now introduce the coordinate descent updates for γ variables. Recall that there are k fairness constraints, thus k dual variables. The outer iteration of computing $\gamma^{(t)}$ consists of k inner iterations. For $l = 0, \dots, k$, we use $\gamma^{(l,t-1)}$ to denote the solution obtained from the l-th inner iteration, then $\gamma^{(0,t-1)} = \gamma^{(t-1)}$, $\gamma^{(l,t-1)} = [\gamma_1^{(t)}, \dots, \gamma_l^{(t)}, \gamma_{l+1}^{(t-1)}, \dots, \gamma_k^{(t-1)}]$ for $l = 1, \dots, k-1$, and $\gamma^{(k,t-1)} = \gamma^{(t)}$. The l-th inner iteration optimizes the objective function $g(\alpha^{(t)}, \gamma)$ along the direction of γ_l with the following optimization problem where e_l is the k-dimensional unit vector with the l-th element equal to 1.

$$\min_d g(\alpha^{(t)}, \gamma^{(l-1,t-1)} + d e_l) \;\; \text{s.t.} \; \gamma_l^{(t-1)} + d \ge 0 \tag{3.32}$$

The closed form solution to (3.32) is stated in the following proposition.

Proposition 3. Suppose $\alpha^{(t)}$ is a given feasible solution in iteration t, for $l = 1, \dots, k$, the optimal solution to (3.32) is $d_l^* = \max\{d_l, -\gamma_l^{(t-1)}\}$, where

$$d_l := \frac{\sum_{i=1}^n A_{li} \alpha_i^{(t)} + \sum_{l'=1}^k P_{ll'} \gamma_{l'}^{(l-1,t-1)} - (c_l - \epsilon_l)}{P_{ll}} = \frac{\nabla_{\gamma_l} g(\alpha^{(t)}, \gamma^{(l-1,t-1)})}{P_{ll}}.$$

Therefore $g_l^{(t)} = \max\{g_l^{(t-1)} - \frac{\nabla_{g_l} g(m^{(t)};g^{(l-1;t-1)})}{P_{ll}}; 0\}$ for all l.

Proof. The objective function of (3.32) simplifies to the following format:

$$g(m^{(t)};g^{(l-1;t-1)} + de_l) = \frac{1}{2}\left\| \sum_{i=1}^{n} m_i^{(t)} y_i f(x_i) - \sum_{l'=1}^{k} g_{l'}^{(l-1;t-1)} p_{l'} - dp_l \right\|^2 - (c_l - e_l)d$$

We observe that the objective function is a convex function of d. In addition, the gradient has the following formula:

$$\begin{aligned}
\nabla_d g(m^{(t)};g^{(l-1;t-1)} + de_l) &= -\langle p_l; \textstyle\sum_{i=1}^{n} m_i^{(t)} y_i f(x_i) - \sum_{l'=1}^{k} g_{l'}^{(l-1;t-1)} p_{l'} - dp_l \rangle - (c_l - e_l) \\
&= dP_{ll} + \textstyle\sum_{i=1}^{n} A_{li} m_i^{(t)} + \sum_{l'=1}^{k} P_{ll'} g_{l'}^{(l-1;t-1)} - (c_l - e_l) \\
&= dP_{ll} + \nabla_{g_l} g(m^{(t)};g^{(l-1;t-1)}) = dP_{ll} - d_l P_{ll}
\end{aligned}$$

The last two equations follow directly from the definition of $\nabla_{g_l} g(m^{(t)};g^{(l-1;t-1)})$ and d_l. Since (3.32) contains a single bound constraint, if the stationary point of the objective function satisfies the constraint, then the stationary point is the optimal solution. Namely,

$$d_l = d_l \text{ when } g_l^{(t-1)} + d_l \geq 0:$$

Otherwise, if $g_l^{(t-1)} + d_l < 0$, which further implies that $\nabla_{g_l} g(m^{(t)};g^{(l-1;t-1)}) > g_l^{(t-1)} P_{ll} \geq 0$, therefore the objective function is minimized at the smallest d_l feasible. Namely,

$$d_l = -g_l^{(t-1)} \text{ when } g_l^{(t-1)} + d_l < 0:$$

The optimal formula for d_l implies that $g_l^{(t)} = \max\{g_l^{(t-1)} + d_l; 0\}$.

An alternative interpretation of the subproblem (3.32) is that in the t-th outer iteration with a given feasible $m^{(t)}$, in each inner iteration for $l = 1;\ldots;k$, we update g_l to fix the violated g_l-related KKT condition at $m^{(t)}$ and the latest updated $g^{(l-1;t-1)}$. This is formalized in the next proposition.

Proposition 4. Suppose $m^{(t)}$ is a given feasible solution iteration t, and $g^{(1;t-1)};\ldots;g^{(k;t-1)}$ is the sequence of updates generated by coordinate descent method with subproblems (3.32), then for all $l = 1;\ldots;k$, $(m^{(t)};g^{(l;t-1)})$ satisfies the following conditions where $g_l^{(t)} = g_l^{(l;t-1)}$ is the update from the l-th inner iteration:

$$g_l^{(t)} = 0 \Rightarrow \nabla_{g_l} g(m^{(t)};g^{(l;t-1)}) \geq 0; \; g_l^{(t)} > 0 \Rightarrow \nabla_{g_l} g(m^{(t)};g^{(l;t-1)}) = 0:$$

Proof. We have derived that the gradient of g with respect to g_l is:

$$\nabla_{g_l} g(m;g) = \sum_{j=1}^{n} A_{lj} m_j + \sum_{l'=1}^{k} P_{ll'} g_{l'} - (c_l - e_l):$$

In iteration t, given a feasible $\mu^{(t)}$, the gradient $\nabla_\gamma g$ after the l-th inner iteration satisfies:

$$\nabla_\gamma g(\mu^{(t)};\gamma^{(l;t-1)}) = \nabla_\gamma g(\mu^{(t)};\gamma^{(l-1;t-1)}) + P_{ll}(\gamma_l^{(t)} - \gamma_l^{(t-1)}).$$

We next discuss in two separate cases. First, $\gamma_l^{(t)} > 0$, by the derived γ update formula in Proposition 3, this happens when $\gamma_l^{(t)} = \gamma_l^{(t-1)} - \frac{\nabla_\gamma g(\mu^{(t)};\gamma^{(l-1;t-1)})}{P_{ll}}$, therefore we can conclude

$$\nabla_\gamma g(\mu^{(t)};\gamma^{(l;t-1)}) = \nabla_\gamma g(\mu^{(t)};\gamma^{(l-1;t-1)}) + P_{ll}(\gamma_l^{(t)} - \gamma_l^{(t-1)}) = 0.$$

The second case is $\gamma_l^{(t)} = 0$, which happens when $0 \geq \gamma_l^{(t-1)} - \frac{\nabla_\gamma g(\mu^{(t)};\gamma^{(l-1;t-1)})}{P_{ll}}$. This further implies

$$\nabla_\gamma g(\mu^{(t)};\gamma^{(l;t-1)}) = \nabla_\gamma g(\mu^{(t)};\gamma^{(l-1;t-1)}) + P_{ll}(\gamma_l^{(t)} - \gamma_l^{(t-1)}) \geq 0.$$

Implementation techniques

We use caching to reduce the needed kernel evaluations in both fair SMO and fair DCD. In addition, both algorithms maintain an efficient update of the gradient $\nabla g(\mu,\gamma)$ by updating the gradients along with the variable updates. In Fair SMO, after computing $\mu^{(t)}$ in iteration t, we update all gradients once with the following formulas:

$$\begin{aligned}
\nabla_{\mu_i} g(\mu^{(t)};\gamma^{(t-1)}) &= \nabla_{\mu_i} g(\mu^{(t-1)};\gamma^{(t-1)}) + \textstyle\sum_{j=1}^{n} Q_{ij}(\mu_j^{(t)} - \mu_j^{(t-1)}) \ \forall i \\
\nabla_{\gamma_l} g(\mu^{(t)};\gamma^{(t-1)}) &= \nabla_{\gamma_l} g(\mu^{(t-1)};\gamma^{(t-1)}) + \textstyle\sum_{j=1}^{n} A_{lj}(\mu_j^{(t)} - \mu_j^{(t-1)}) \ \forall l
\end{aligned} \tag{3.33}$$

Then during the coordinate descent steps updating $\gamma^{(t-1)}$ to $\gamma^{(t)}$, we update all gradients after each inner iteration. Namely, after the l-th iteration computes $\gamma_l^{(t)}$, we update new gradients as below:

$$\begin{aligned}
\nabla_{\mu_i} g(\mu^{(t)};\gamma^{(l;t-1)}) &= \nabla_{\mu_i} g(\mu^{(t)};\gamma^{(l-1;t-1)}) + A_{li}(\gamma_l^{(t)} - \gamma_l^{(t-1)}) \ \forall i \\
\nabla_{\gamma_{l'}} g(\mu^{(t)};\gamma^{(l;t-1)}) &= \nabla_{\gamma_{l'}} g(\mu^{(t)};\gamma^{(l-1;t-1)}) + P_{ll'}(\gamma_l^{(t)} - \gamma_l^{(t-1)}) \ \forall l'.
\end{aligned} \tag{3.34}$$

In fair DCD, an iteration t updates all gradients in each inner iteration of μ update with (3.33) and in each inner iteration of γ update with (3.34).

3.4 Theoretical Properties of Fair SVM Algorithms

We prove that both specialized algorithms for fair SVM asymptotically converge to a optimal solution of (3.7). In particular, we establish the stronger linear convergence property for fair DCD, which implies that fair DCD returns a solution with up to ε violation to the KKT conditions, namely reaches the termination criteria, in infinitely many iterations. Moreover, we show that the shrinking heuristic for standard DCD is applicable to fair DCD and guarantees a finite termination.

3.4.1 Asymptotic convergence of fair SMO for fair general SVMs

We observe that fair SMO algorithm is a block coordinate descent method, as each iteration finds a new feasible solution by either updating a block of size two among the m variables or a block of size one among the g variables. Tseng and Yun (2010) prove the asymptotic convergence of a block coordinate descent method for linearly constrained smooth optimization when the block is chosen to provide sufficient descent in the objective value. For completeness, we state the convergence result from Tseng and Yun (2010).

Theorem 13 (Theorem 4.1 Tseng and Yun (2010)). Given a linearly constrained smooth optimization problem

$$\min_x f(x) \text{ s.t. } Ax = b, l \le x \le u, x \in \mathbb{R}^n.$$

In a block coordinate descent algorithm, iteration t consists of the following steps:

1. select a nonempty block $B^{(t)}$ from all x coordinates and a symmetric matrix $H^{(t)} \in \mathbb{R}^{n \times n}$ with $B^T_{B^{(t)}} H^{(t)}_{B^{(t)}B^{(t)}} B_{B^{(t)}} \succ 0$, where $B_{B^{(t)}}$ is a matrix whose columns form an orthonormal basis for $\mathrm{Null}(A_{B^{(t)}})$.
2. update $x^{(t+1)} = x^{(t)} + a^{(t)} d^{(t)}$, with $a^{(t)} > 0$ and $d^{(t)} = \operatorname{argmin}_{d \in \mathbb{R}^n} \{\nabla f(x)^T d + \frac{1}{2} d^T H^{(t)} d : A(x^{(t)} + d) = b, l \le x^{(t)} + d \le u, d_i = 0 \, \forall i \notin B^{(t)}\}$.

Every limit point of $\{x^{(t)}\}$ is an optimal solution if the following conditions are satisfied:

(a) $f(x^{(t)} + a^{(t)} d^{(t)}) \le f(x^{(t)} + \tilde{a}^{(t)} d^{(t)})$ for all t, where $\{\tilde{a}^{(t)}\}$ is chosen by the Armijo rule.

(b) There exists $0 < v \le 1$ such that for all t,

$$\begin{aligned} &\min\{\nabla f(x)^T d + \frac{1}{2} d^T \mathrm{diag}(H^{(t)}) d : A(x^{(t)} + d) = b, l \le x^{(t)} + d \le u, d_i = 0 \, \forall i \notin B^{(t)}\} \\ \le\, & v \min\{\nabla f(x)^T d + \frac{1}{2} d^T \mathrm{diag}(H^{(t)}) d : A(x^{(t)} + d) = b, l \le x^{(t)} + d \le u\}. \end{aligned} \tag{3.35}$$

In addition, there exist $0 < \underline{l} \le \bar{l}$ such that for all t, $\bar{l} \succeq \mathrm{diag}(H^{(t)}) \succeq \underline{l}$.

To prove the asymptotic convergence of fair SMO algorithm, it suffices to show that fair SMO satisfies all the needed conditions in (Tseng and Yun, 2010, Theorem 4.1), thus the convergence result applies. We first prove the following lemma which states the degree of improvement achieved with each update.

Lemma 1. Suppose $\{(m^{(t)}, g^{(t)})\}$ is the sequence generated by fair SMO. In the outer iteration t, suppose $\{g^{(l,t)}\}_{l \in [k]}$ is the sequence generated by the inner iterations updating g. Then

$$g(m^{(t+1)}, g^{(t)}) - g(m^{(t)}, g^{(t)}) \le -\frac{1}{4} a_{i_t j_t} \left\| m^{(t+1)} - m^{(t)} \right\|^2, \tag{3.36}$$

$$g(\mu^{(t+1)};\gamma^{(l;t)}) - g(\mu^{(t+1)};\gamma^{(l-1;t)}) \geq \frac{1}{2}P_l(\gamma_l^{l;t} - \gamma_l^{l-1;t})^2; \tag{3.37}$$

where $\{i_t;j_t\}$ is the working set used in iteration t and $a_{i_tj_t} = \|f(x_{i_t}) - f(x_{j_t})\|^2$. Moreover,

$$g(\mu^{(t+1)};\gamma^{(t+1)}) - g(\mu^{(t)};\gamma^{(t)}) \geq \frac{1}{4}a_{i_tj_t}\|\mu^{(t+1)} - \mu^{(t)}\|^2 + \frac{1}{2}\sum_{l=1}^{k} P_l(\gamma_l^{(t+1)} - \gamma_l^{(t)})^2. \tag{3.38}$$

Proof. Suppose fair SMO chooses $\{i_t;j_t\}$ as the working set in iteration t. For notation ease, we denote $(i;j) = (i_t;j_t)$, then the optimal updates without the bound constraints are:

$$\mu_i = \mu_i^{(t)} + y_i d = \mu_i^{(t)} + y_i\frac{b_{ij}}{a_{ij}}; \quad \mu_j = \mu_j^{(t)} - y_j d = \mu_j^{(t)} - y_j\frac{b_{ij}}{a_{ij}}$$

With the additional constraints $0 \leq \mu_i^{(t)} + y_i d \leq C$ and $0 \leq \mu_j^{(t)} + y_j d \leq C$, we suppose $L \leq d \leq U$ is the feasible domain of d. Then the optimal feasible updates are:

$$\mu_i^{(t+1)} = \mu_i^{(t)} + y_i\bar{d}; \quad \mu_j^{(t+1)} = \mu_j^{(t)} - y_j\bar{d}; \quad \text{where } \bar{d} := \min\left\{\frac{b_{ij}}{a_{ij}};U\right\}.$$

Recall from the working set selection rule $b_{ij} \geq 0$, so we have $\bar{d} \geq 0$. Since the other indices are not modified in iteration t, we conclude $\|\mu^{(t+1)} - \mu^{(t)}\|^2 = 2\bar{d}^2$. Using the definition of $g(\mu;\gamma)$, we have:

$$\begin{aligned} g(\mu^{(t+1)};\gamma^{(t)}) - g(\mu^{(t)};\gamma^{(t)}) &= -\left(\frac{1}{2}K_{ii} + \frac{1}{2}K_{jj} - K_{ij}\right)\bar{d}^2 + \left(-y_i\nabla_{\mu_i}g(\mu^{(t)};\gamma^{(t)}) + y_j\nabla_{\mu_j}g(\mu^{(t)};\gamma^{(t)})\right)\bar{d} \\ &= -\frac{1}{2}a_{ij}\bar{d}^2 + b_{ij}\bar{d} = -(a_{ij}\bar{d} - b_{ij})\bar{d} + \frac{1}{2}a_{ij}\bar{d}^2 \\ &\geq \frac{1}{2}a_{ij}\bar{d}^2 = \frac{a_{ij}}{4}\|\mu^{(t+1)} - \mu^{(t)}\|^2. \end{aligned}$$

The last inequality follows from $a_{ij}\bar{d} - b_{ij} \leq 0$ and $\bar{d} \geq 0$.

Now, to prove (3.37), we again begin with the definition of g to simplify the function difference formula. Recall from Proposition 3, $d_l = \max\left\{\frac{\nabla_{\gamma_l} g(\mu^{(t+1)};\gamma^{(l-1;t)})}{P_l}; -\gamma_l^{(t)}\right\}$.

$$\begin{aligned} g(\mu^{(t+1)};\gamma^{(l;t)}) - g(\mu^{(t+1)};\gamma^{(l-1;t)}) &= -\frac{1}{2}P_l(d_l)^2 - \left(\epsilon_l - c_l + \sum_{i=1}^{n}\mu_i A_{li} + \sum_{l'=1}^{k}\gamma_{l'}P_{l'l}\right)d_l \\ &= (-P_l d_l + \nabla_{\gamma_l} g(\mu^{(t)};\gamma^{(l-1;t)}))d_l + \frac{1}{2}P_l(d_l)^2 \end{aligned}$$

In the case when $d_l = \frac{\nabla_{\gamma_l} g(\mu^{(t+1)};\gamma^{(l-1;t)})}{P_l}$, the RHS in the above formula simplifies to $\frac{1}{2}P_l(d_l)^2$. In the other case when $d_l = -\gamma_l^{(t)} \leq 0$, we have $d_l \geq \frac{\nabla_{\gamma_l} g(\mu^{(t+1)};\gamma^{(l-1;t)})}{P_l}$, equivalently $(-P_l d_l + \nabla_{\gamma_l} g(\mu^{(t+1)};\gamma^{(l-1;t)})) \leq 0$. Together with $d_l \leq 0$, we conclude that $g(\mu^{(t+1)};\gamma^{(l;t)}) - g(\mu^{(t+1)};\gamma^{(l-1;t)}) \geq \frac{1}{2}P_l(d_l)^2$ in this case.

After all k inner iterations, we obtain $g^{(t+1)}$. As we have shown, for each $l = 1,\ldots,k$, updating the l-th coordinate of g reduces the value of g by at least $\frac{1}{2}P_{ll}(d_l)^2$. Therefore, considering all the updates,

$$
\begin{aligned}
& g(m^{(t+1)};g^{(t+1)}) - g(m^{(t)};g^{(t)}) \\
= \; & g(m^{(t+1)};g^{(k;t)}) - g(m^{(t+1)};g^{(k-1;t)}) + \cdots + g(m^{(t+1)};g^{(1;t)}) - g(m^{(t+1)};g^{(0;t)}) + g(m^{(t+1)};g^{(t)}) - g(m^{(t)};g^{(t)}) \\
& -\frac{1}{2}P_{kk}(d_l)^2 - \frac{1}{2}P_{k-1;k-1}(d_l)^2 - \cdots - \frac{1}{2}P_{11}(d_l)^2 - \frac{1}{4}a_{i_t j_t}\left\| m^{(t+1)} - m^{(t)} \right\|^2 \\
= \; & -\frac{1}{4}a_{i_t j_t}\left\| m^{(t+1)} - m^{(t)} \right\|^2 - \frac{1}{2}\sum_{l=1}^{k} P_{ll}(g^{l;t} - g^{l-1;t})^2.
\end{aligned}
$$

Theorem 14. Suppose $f(m^{(t)};g^{(t)})g$ is the infinite sequence generated by fair SMO with tolerance $t = 0$, a limit point $(\bar{m};\bar{g})$ of $f(m^{(t)};g^{(t)})g$ an optimal solution to (3.7).

Proof. Fair SMO solves the dual fair SVM formulation (3.7), where the objective function $g(m;g)$ is smooth, and the constraint matrix is $1 \times (n+k)$ matrix with y_i in the first n indices and 0 in the remaining k indices. In iteration t, the working block is $B^{(t)} = f\,m_{i_t};m_{j_t}\,g$ or $B^{(t)} = f\,g\,g$. Then along the descent direction $d^{(t)}$, fair SMO solves the subproblem (3.30) or (3.32) to find the optimal step size $a^{(t)}$. In addition, fair SMO uses $H^{(t)} = \bar{Q} = \begin{bmatrix} Q & A \\ A^T & P \end{bmatrix}$, the Hessian of the objective function g, for all t.

In iteration t, if $B^{(t)} = f\,m_{i_t};m_{j_t}\,g$, then $H^{(t)}_{B^{(t)}B^{(t)}} = \begin{bmatrix} Q_{i_t i_t} & Q_{i_t j_t} \\ Q_{i_t j_t} & Q_{j_t j_t} \end{bmatrix}$ and $B_{B^{(t)}} = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}$. If $B^{(t)} = f\,g\,g$, then $H^{(t)}_{B^{(t)}B^{(t)}} = P_{ll}$ and $B_{B^{(t)}} = 1$. In both cases, we derive that $B^T_{B^{(t)}} H^{(t)}_{B^{(t)}B^{(t)}} B_{B^{(t)}} \succ 0$. Together with the definitions of $d^{(t)}$, we have verified that fair SMO is a block coordinate descent algorithm as stated in Theorem 13.

We next show that $f(m^{(t)};g^{(t)});B^{(t)};H^{(t)};d^{(t)};a^{(t)}g$ from fair SMO satisfies Theorem 13(a) and (b). Since fair SMO applies a minimization rule to find $a^{(t)}$, which is at least as well as the Armijo rule, we immediately conclude (a) is satisfied. To verify (b), we observe that

$$
g(m^{(t+1)};g^{(t)}) - g(m^{(t)};g^{(t)}) = \nabla_{m_{B^{(t)}}} g(m^{(t+1)};g^{(t)})^T d^{(t)} + \frac{1}{2}{d^{(t)}}^T \operatorname{diag}(H^{(t)}) d^{(t)};
$$

$$
g(m^{(t+1)};g^{(l;t)}) - g(m^{(t+1)};g^{(l-1;t)}) = \nabla_g\, g(m^{(t+1)};g^{(l-1;t)})^T d^{(t)} + \frac{1}{2}{d^{(t)}}^T \operatorname{diag}(H^{(t)}) d^{(t)}.
$$

Therefore, Lemma 1 implies that for all t the LHS of (3.35) is negative. Moreover, the RHS of (3.35) is no smaller than the LHS, as the maximal objective descent when allowing all indices to change is at least as large as the maximal descent when only $B^{(t)}$ indices can change. Therefore, there exists $0 < v \leq 1$ such that (3.35) holds for all t. Lastly, by definition of $H^{(t)}$, we easily conclude that $\bar{l} \succeq \operatorname{diag}(H^{(t)}) \succeq \underline{l}$ with $\bar{l} = \max f\,Q_{ii};P_{ll}\,g$ and $\underline{l} = \min f\,Q_{ii};P_{ll}\,g$.

Therefore, Theorem 13 implies that a limit point $(\bar{m}, \bar{g})$ of $\{(m^{(t)}; g^{(t)})\}$ is an optimal solution of to (3.7).

Another useful property is to show the finite termination of fair SMO in this general case. Extending the finite termination proofs of standard SMO turns out to be insufficient for this purpose: Chen et al. (2006) prove that standard SMO terminates in finite iterations by using the continuity of $\nabla_m h(m)$ and a counting argument to show that SMO will run out of m indices to update. In fair SMO, the counting technique is invalid because the gradients $\nabla_m g(m, g)$ are updated with both m and g value changes. In Tseng and Yun (2010), besides the asymptotic convergence result used above to establish the asymptotic convergence of fair SMO, they also proved linear convergence for a working set selection utilizing second order information $\nabla_{m_i; m_j} h(m^{(t)})$, which is computationally more costly than the maximal violating pair rule we use but is more efficient in reaching the optimal solution. One possible future step is to investigate fair SMO with this alternative working set selection rule, and utilize the linear convergence result in Tseng and Yun (2010) to establish stronger convergence guarantees, including finite termination guarantees.

3.4.2 Linear convergence of coordinate descent for fair linear SVMs

Our algorithm for fair linear SVM is a standard coordinate descent method, with each iteration updating one variable in the steepest descent direction. In fact, our algorithm can be viewed as a direct extension of the coordinate descent method that Hsieh et al. (2008) proposed for training standard linear SVMs. We therefore follow the proof techniques used in Hsieh et al. (2008) to establish theoretical properties of our algorithm.

Theorem 15. Suppose $\{m^{(t)}; g^{(t)}\}$ is an infinite sequence generated by the coordinate descent algorithm for fair linear SVMs, then the sequence globally converges to an optimal solution $(m^*; g^*)$ with a convergence rate that is at least linear. Namely, there exists $0 \le a < 1$ and an iteration t_0 such that $g(m^{(t+1)}; g^{(t+1)}) - g(m^*; g^*) \le a\,(g(m^{(t)}; g^{(t)}) - g(m^*; g^*))$ for all $t \ge t_0$.

Proof. In Luo and Tseng (1992), they proved the linear convergence of coordinate descent method for the following convex optimization problems with $L_i \in [-\infty; \infty)$ and $U_i \in (-\infty; \infty]$:

$$\min_x f(Ex) + b^T x \text{ s.t. } L_i \le x_i \le U_i. \tag{3.39}$$

The linear convergence holds when the following conditions are satisfied: (a) E has no zero columns; (b) The set of optimal solutions to (3.39) is nonempty; (c) f is strongly convex and twice differentiable everywhere. We next show that the dual fair linear SVM problem satisfies all these conditions. Recall from (3.7), fair linear SVM has the following dual formulation:

$$\min_{m,g} \frac{1}{2} \begin{bmatrix} m \\ g \end{bmatrix}^T \begin{bmatrix} Q & A \\ A^T & P \end{bmatrix} \begin{bmatrix} m \\ g \end{bmatrix} - \begin{bmatrix} e \\ d \end{bmatrix}^T \begin{bmatrix} m \\ g \end{bmatrix} \text{ s.t. } 0 \le m_i \le C;\ \forall i \in [n];\ g_l \ge 0;\ \forall l \in [k].$$

In the notation of (3.39), $E = \begin{bmatrix} y_1 x_1 & \cdots & y_n x_n & p_1 & \cdots & p_l \end{bmatrix}$ and $f(E\begin{bmatrix}\mu & \gamma\end{bmatrix}^T) = \frac{1}{2}\left\|E\begin{bmatrix}\mu & \gamma\end{bmatrix}\right\|^2$.

We observe that (a) holds by our assumption that there is no all-zero x_i or p_l as both are trivial cases that can be eliminated from consideration. (c) holds by the format of f. In addition, (b) can be shown with strong duality. In the primal problem $\min_{\theta,b,\xi}\{\frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^n \xi_i : \xi_i \geq 1 - y_i(\theta^T x_i + b); \xi_i \geq 0\ \forall i; \theta^T p_l \leq \epsilon_l\ \forall l\}$, we can find θ, b and $\xi > 0$ that satisfies all constraints as strict inequalities, namely, the Slater's condition is satisfied. Therefore, strong duality holds for fair SVM with a linear kernel, implying that the dual program has a nonempty optimal solution set and the optimal dual value is equal to the optimal primal value.

Therefore, by (Luo and Tseng, 1992, Theorem 2.1), coordinate descent method on fair linear SVMs converges linearly to an optimal solution.

3.4.3 Finite termination of coordinate descent with shrinking for fair linear SVMs

The standard DCD from Hsieh et al. (2008) adopt the same shrinking technique used in standard SMO, that is, remove i from the current set of active indices if $\mu_i^{(t)} = 0$ or $\mu_i^{(t)} = C$. In fair DCD, since we have both μ and γ variables, we can shrink the variables at bounds, namely $\mu_i = 0$ or $\mu_i = C$ and $\gamma_l = 0$, from both sets. We will show that fair DCD with this shrinking heuristic reaches the KKT based termination criteria in finite iterations. We begin by observing some properties of the optimal solutions to fair linear SVM, and the objective descent in each iteration.

Theorem 16. Let μ^*, γ^* be the limit point of $\{\mu^{(i,t)}, \gamma^{(l,t)}\}$ generated by coordinate descent algorithm.

(a) If $\mu_i^* = 0$ and $\nabla_{\mu_i} g(\mu^*, \gamma^*) > 0$, then there exists t_i such that $\forall t \geq t_i$, $\forall s$, $\mu_i^{s,t} = 0$.

(b) If $\mu_i^* = C$ and $\nabla_{\mu_i} g(\mu^*, \gamma^*) < 0$, then there exists t_i such that $\forall t \geq t_i$, $\forall s$, $\mu_i^{s,t} = C$.

(c) If $\gamma_l^* = 0$ and $\nabla_{\gamma_l} g(\mu^*, \gamma^*) > 0$, then there exists t_l such that $\forall t \geq t_l$, $\forall m$, $\gamma_l^{m,t} = 0$.

(d) $\lim_{t\to\infty} \max\{\max_i \nabla^P_{\mu_i} g(\mu^{(i,t)}, \gamma^{(0,t)}); \max_l \nabla^P_{\gamma_l} g(\mu^{(n,t)}, \gamma^{(l,t)})\}$
$= \lim_{t\to\infty} \min\{\min_i \nabla^P_{\mu_i} g(\mu^{(i,t)}, \gamma^{(0,t)}); \min_l \nabla^P_{\gamma_l} g(\mu^{(n,t)}, \gamma^{(l,t)})\} = 0$.

Proof. From the global convergence proved in Theorem 15, μ^*, γ^* is an optimal solution to dual fair linear SVM. By strong duality and optimality conditions, $\theta^* = \sum_{i=1}^n \mu_i^* y_i \begin{bmatrix} x_i \\ 1 \end{bmatrix} - \sum_{l=1}^k \gamma_l^* \begin{bmatrix} p_l \\ 0 \end{bmatrix}$ is an optimal primal solution. Moreover, θ^* is the unique primal optimal solution since the primal problem has a strictly convex objective function. For the sake of this proof, we re-assign the indices of the infinite sequence $\{\mu^{(i,t)}, \gamma^{(l,t)}\}$ as $\{\mu^{(t)}, \gamma^{(t)}\}$, where updating $\mu^{(t)}, \gamma^{(t)}$ to $\mu^{(t+1)}, \gamma^{(t+1)}$ corresponds to solving the sub-problem at one coordinate (either μ or γ). Let $\theta^{(t)} = \sum_{i=1}^n \mu_i^{(t)} y_i \begin{bmatrix} x_i \\ 1 \end{bmatrix} - \sum_{l=1}^k \gamma_l^{(t)} \begin{bmatrix} p_l \\ 0 \end{bmatrix}$.

From the global convergence to optimality, we have

$$\lim_{t\to\infty} \sum_{i=1}^n \mu_i^{(t)} y_i \begin{bmatrix} x_i \\ 1 \end{bmatrix} - \sum_{l=1}^k \gamma_l^{(t)} \begin{bmatrix} p_l \\ 0 \end{bmatrix} = \lim_{t\to\infty} \theta^{(t)} = \theta^* .$$

This further implies

$$\lim_{t\to\infty} \nabla_{\mu_i} g(\mu^{(t)},\gamma^{(t)}) = \lim_{t\to\infty} y_i\left\langle \theta^{(t)}, \begin{bmatrix}\xi_i\\1\end{bmatrix}\right\rangle - 1 = y_i\left\langle \theta^*, \begin{bmatrix}\xi_i\\1\end{bmatrix}\right\rangle - 1 = \nabla_{\mu_i} g(\mu^*,\gamma^*); \tag{3.40}$$

$$\lim_{t\to\infty} \nabla_{\gamma_l} g(\mu^{(t)},\gamma^{(t)}) = \lim_{t\to\infty} \left\langle \theta^{(t)}, \begin{bmatrix}\pi_l\\0\end{bmatrix}\right\rangle + \delta_l = \left\langle \theta^*, \begin{bmatrix}\pi_l\\0\end{bmatrix}\right\rangle + \delta_l = \nabla_{\gamma_l} g(\mu^*,\gamma^*). \tag{3.41}$$

There exists t_i such that for all $t \geq t_i$,

$$\nabla_{\mu_i} g(\mu^*,\gamma^*) > 0 \to \nabla_{\mu_i} g(\mu^{(t)},\gamma^{(t)}) > 0,\ \nabla_{\mu_i} g(\mu^*,\gamma^*) < 0 \to \nabla_{\mu_i} g(\mu^{(t)},\gamma^{(t)}) < 0. \tag{3.42}$$

Similarly, there exists t_l such that for all $t \geq t_l$,

$$\nabla_{\gamma_l} g(\mu^*,\gamma^*) > 0 \to \nabla_{\gamma_l} g(\mu^{(t)},\gamma^{(t)}) > 0,\ \nabla_{\gamma_l} g(\mu^*,\gamma^*) < 0 \to \nabla_{\gamma_l} g(\mu^{(t)},\gamma^{(t)}) < 0. \tag{3.43}$$

When we update $\mu^{(t)},\gamma^{(t)}$ to $\mu^{(t+1)},\gamma^{(t+1)}$ by changing μ_i, the KKT condition of the subproblem implies:

$$\nabla_{\mu_i} g(\mu^{(t+1)},\gamma^{(t+1)}) > 0 \to \mu_i^{(t+1)} = 0,\ \nabla_{\mu_i} g(\mu^{(t+1)},\gamma^{(t+1)}) < 0 \to \mu_i^{(t+1)} = C. \tag{3.44}$$

These relations imply, after $t \geq t_i$, μ_i is always 0 or always C. We also note that with $\nabla_{\mu_i} g(\mu^{(t+1)},\gamma^{(t+1)}) \neq 0$ for all $t \geq t_i$, it is impossible that $\mu_i \in (0,C)$, because it will contradict the optimality condition. Therefore, (3.42) and (3.44) imply (a) and (b). To obtain the exact statements, we need to switch back to the original indices $\{\mu_i^{(s,t)}\}$ where $s \in \{1,\dots,n\}$.

Similarly, when updating $\mu^{(t)},\gamma^{(t)}$ to $\mu^{(t+1)},\gamma^{(t+1)}$ changes γ_l, the KKT condition of the subproblem implies:

$$\nabla_{\gamma_l} g(\mu^{(t+1)},\gamma^{(t+1)}) > 0 \to \mu_l^{(t+1)} = 0. \tag{3.45}$$

After $t \geq t_l$, γ_l is always 0. We now conclude (c) from (3.43) and (3.45), and switching back to the index $\gamma_l^{(m,t)}$ where $m \in \{1,\dots,k\}$.

Moreover, (3.42) and (3.43) imply that for $t \geq t_i$, if $\nabla_{\mu_i} g(\mu^*,\gamma^*) \neq 0$, then $\nabla^P_{\mu_i} g(\mu^{(t)},\gamma^{(t)}) = 0$. The other case $\nabla_{\mu_i} g(\mu^*,\gamma^*) = 0$ directly implies $\nabla^P_{\mu_i} g(\mu^{(t)},\gamma^{(t)}) = 0$. Similarly, for $t \geq t_l$, we conclude that $\nabla^P_{\gamma_l} g(\mu^{(t)},\gamma^{(t)}) = 0$. Changing back to the original indices, we have shown (d).

Lemma 2. Suppose $\{(\mu^{(t)},\gamma^{(t)})\}$ is the sequence generated by coordinate descent algorithm. In the outer iteration t, $\{\mu^{(i,t)},\gamma^{(l,t)}\}$ is the sequence generated by the inner iterations. Then

$$g(\mu^{(i,t)},\gamma^{(0,t)}) - g(\mu^{(i-1,t)},\gamma^{(0,t)}) \leq -\frac{1}{2}Q_{ii}(\mu_i^{(i,t)} - \mu_i^{(i-1,t)})^2; \tag{3.46}$$

$$g(m^{(n;t)}, g^{(l;t)}) - g(m^{(n;t)}, g^{(l-1;t)}) \le -\frac{1}{2}P_{ll}(g_l^{(l;t)} - g_l^{(l-1;t)})^2. \tag{3.47}$$

Proof. When updating $(m^{(i-1;t)}, g^{(0;t)})$ to $(m^{(i;t)}, g^{(0;t)})$, we solve the subproblem (3.31). From the KKT conditions of the subproblem, we have

$$\nabla_{m_i} g(m^{(i;t)}, g^{(0;t)}) > 0 \Rightarrow m_i^{(i-1;t)} - m_i^{(i;t)} \ge 0;\quad \nabla_{m_i} g(m^{(i;t)}, g^{(0;t)}) < 0 \Rightarrow m_i^{(i-1;t)} - m_i^{(i;t)} \le 0.$$

Therefore,

$$\begin{aligned} g(m^{(i;t)}, g^{(0;t)}) - g(m^{(i-1;t)}, g^{(0;t)}) &= \nabla_{m_i} g(m^{(i;t)}, g^{(0;t)})(m_i^{(i;t)} - m_i^{(i-1;t)}) - \frac{1}{2}Q_{ii}(m_i^{(i;t)} - m_i^{(i-1;t)})^2 \\ &\le -\frac{1}{2}Q_{ii}(m_i^{(i;t)} - m_i^{(i-1;t)})^2. \end{aligned}$$

Next, when updating $(m^{(n;t)}, g^{(l-1;t)})$ to $(m^{(n;t)}, g^{(l;t)})$, we solve the subproblem (3.32). From the KKT conditions of the subproblem, we have

$$\nabla_{g_l} g(m^{(n;t)}, g^{(l;t)}) > 0 \Rightarrow g_l^{(l;t)} = 0;\quad \nabla_{g_l} g(m^{(n;t)}, g^{(l;t)}) = 0 \Rightarrow g_l^{(l;t)} \ge 0.$$

Therefore,

$$\begin{aligned} g(m^{(n;t)}, g^{(l;t)}) - g(m^{(n;t)}, g^{(l-1;t)}) &= \nabla_{g_l} g(m^{(n;t)}, g^{(l;t)})(g_l^{(l;t)} - g_l^{(l-1;t)}) - \frac{1}{2}P_{ll}(g_l^{(l;t)} - g_l^{(l-1;t)})^2 \\ &\le -\frac{1}{2}P_{ll}(g_l^{(l;t)} - g_l^{(l-1;t)})^2. \end{aligned}$$

We next prove the finite termination of coordinate descent with the selected shrinking heuristic among m variables on fair linear SVM.

Theorem 17. *Algorithm 5 terminates in finite iterations.*

Proof. For the sake of contradiction, we assume Algorithm 5 does not terminate in finite iterations. Then the algorithm will generate an infinite sequence $\{m^{(t)}, g^{(t)} : t = 1, 2, \ldots\}$. Throughout the algorithm, at an iteration, if the active m indices in I and all the g indices satisfy the stopping criteria $M - m \le \epsilon_t, |M| \le \epsilon_t, |m| \le \epsilon_t$, we will un-shrink I back to all m indices and go through all m_i in the next outer iteration (with $n + k$ inner iterations). We collect all these outer iterations with un-shrinking as a subsequence $\{m^{(t)}, g^{(t)}\}_R$.

We first claim that $\{m^{(t)}, g^{(t)}\}_R$ must be infinite. Assume $\{m^{(t)}, g^{(t)}\}_R$ is finite, then we consider the iterations after the last iteration in this set. Theorem 16(d) implies that these future iterations will reach the stopping conditions $M - m \le \epsilon_t, |M| \le \epsilon_t, |m| \le \epsilon_t$ at some point. Thus, $\{m^{(t)}, g^{(t)}\}$ will be a finite sequence, contradicting our assumption that Algorithm 5 does not terminate.

Next, consider a subsequence $\bar{R} \subseteq R$ such that $\{m^{(t)}, g^{(t)} : t \in \bar{R}\}$ converges to (m^*, g^*). Lemma 2 has shown that $\{g(m^{(i;t)}, g^{(l;t)})\}$ with $i \in \{1, \ldots, n\}$ and $l \in \{1, \ldots, k\}$ is decreasing. From the proof of Theorem 13, we have that $\{g(m^{(i;t)}; g^{(l;t)})\}$ is lower bounded by the optimal dual value. Therefore, we conclude

$$\lim_{t \in R, t \to \infty} g(m^{(i;t)}; g^{(0;t)}) - g(m^{(i-1;t)}; g^{(0;t)}) = 0; \quad \lim_{t \in R, t \to \infty} g(m^{(n;t)}; g^{(l;t)}) - g(m^{(n;t)}; g^{(l-1;t)}) = 0$$

Moreover, taking the limits of (3.46) and (3.47), we have

$$\lim_{t \in R, t \to \infty} (m^{(l;t)}; g^{(0;t)}) = \lim_{t \in R, t \to \infty} (m^{(l-1;t)}; g^{(0;t)}) = \lim_{t \in R, t \to \infty} (m^{(n;t)}; g^{(l;t)}) = \lim_{t \in R, t \to \infty} (m^{(n;t)}; g^{(l-1;t)})$$
$$= \lim_{t \in R, t \to \infty} (m^{(t+1)}; g^{(t+1)}) = (m; g):$$

We claim that the limit point $(m; g)$ must be an optimal dual solution. Assume it is not optimal, then there exists an index s where we can solve the corresponding subproblem and obtain an optimal descent d such that $g((m; g) + de_s) < g(m; g)$. Note that the variable at index s can be either m_i or g_l. In each outer iteration t, the inner iteration at this index s solves the subproblem (3.31) or (3.32) to optimality. Suppose the variable at index s is m_i, then $g((m^{(i;t)}; g^{(0;t)}) + de_s) \geq g(m^{(i+1;t)}; g^{(0;t)})$. Taking the limit on both sides gives $g((m; g) + de_s) \geq g(m; g)$. We can derive the same inequality when g_l is the variable at index s. Therefore, we have reached a contradiction. So the claim $(m; g)$ is an optimal dual solution must be true.

We have shown that any limit point of $\{m^{(t)}; g^{(t)}\}_R$ is a dual optimal solution. By Lemma 2, the objective values are decreasing throughout all iterations, so $\{m^{(i;t)}; g^{(0;t)}\}$ for any i and $\{m^{(n;t)}; g^{(l;t)}\}$ for any l also converge to a dual optimal solution. We can apply the proof techniques used in Theorem 16 to conclude that $\tilde{\nabla}^P g(m^{(i;t)}; g^{(0;t)})$ and $\tilde{\nabla}^P g(m^{(n;t)}; g^{(l;t)})$ globally converges to 0. Therefore, the stopping conditions $M - m \leq t$; $|M| \leq t$; $|m| \leq t$ will be reached by Algorithm 5 in finite iterations.

3.5 Numerical Experiments

We implement fair SMO and fair DCD to train SVMs constrained with covariance parity (3.10) and/or true positive rate parity (3.11) constraints. We apply fair SMO to nonlinear SVMs with a Gaussian RBF kernel, and fair DCD to linear SVMs. Our main goal is to demonstrate the practical potentials of these specialized fair SVM algorithms. To this end, we compare the runtime of fair SVM algorithms against their corresponding standard algorithms without fairness constraints. In addition, we also compare the runtime of both fair SVM algorithms with an off-the-shelf quadratic program solver.

Both algorithms are coded in C++ using Visual Studio 16.11.10. In addition, we use the CVXOPT QP solver with the default setting in Python as the off-the-shelf solver. All experiments are conducted in Windows 10 Pro 64-bit on a machine with Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz processors and 24 GB of RAM.

3.5.1 Datasets

We run experiments on two real-world datasets that are suitable for training and testing fair classification algorithms: the German credit dataset Hofmann (1994) and the Adult income dataset adu (1996). As our implementations are based on LIBSVM/LIBLINEAR, we use both datasets in LIBSVM format provided in Chang and Lin (2011). The German credit dataset consists of 1000 data points with 24 features (transformed from 20 attributes), and the label y indicates whether a person has a good ($y = 1$) or bad ($y = -1$) credit risk. For this dataset, we choose the features of housing status as the group attribute: $z = 1$ represents 'rent', and $z = 0$ represents 'own' or 'for free'; we respectively refer to them as 'renters' and 'homeowners'. In the Adult income dataset, each point has 123 features (transformed from 24 attributes), and the label y represents whether a person has income above $50{,}000$ dollars ($y = 1$) or not ($y = -1$). We choose gender as the sensitive feature, and assign $z = 1$ for women and $z = 0$ for men. From this dataset, we use the 'a4a' instance with 4781 data points.

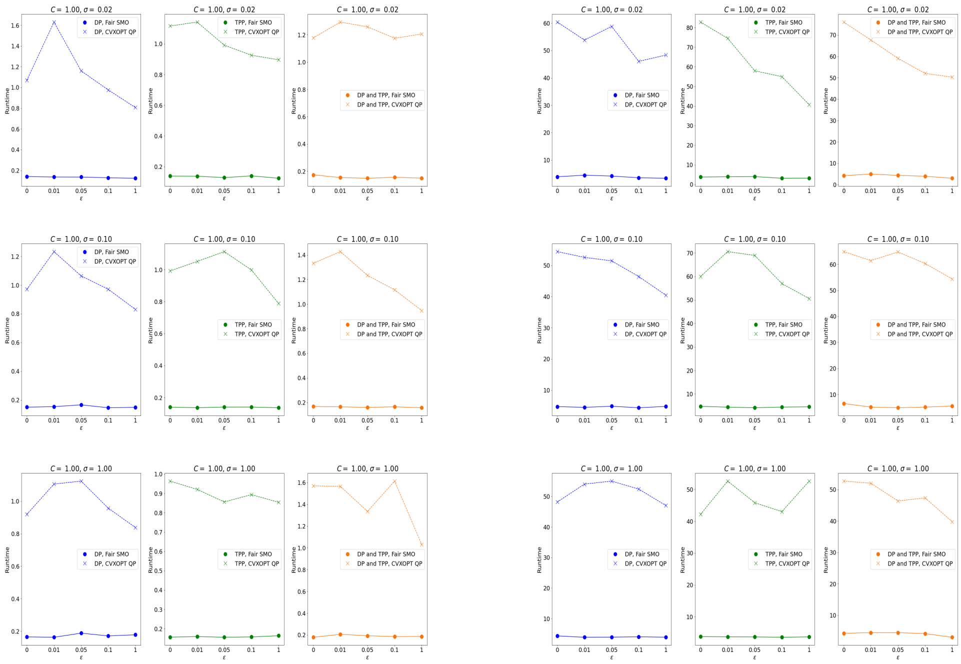
3.5.2 Implementation Details

We implement fair SMO algorithm by modifying the SMO implementation in LIBSVM (Chang and Lin (2011)), and fair DCD by modifying the standard DCD implemented in LIBLINEAR (Fan et al. (2008)). For both algorithms, the main modifications include defining functions to compute $\bar{Q}$ (standard SVM only requires computing the submatrix Q), adding the g update subroutine and replacing all gradient formulas used in the implementation. In addition, we keep the same caching operations provided in the standard algorithms, and extend the shrinking steps in LIBLINEAR to handle both m and g.

On both datasets, we test fair SMO for a RBF kernel and fair DCD for a linear kernel. We consider three fairness settings: (a) demographic parity with covariance parity constraints (3.10), (b) equalized odds with true positive parity constraints (3.11), and (c) both conditions with (3.10) and (3.11). For simplicity, we use the same threshold c for all fairness constraints in one instance. To compare performances at fairness constraints of different strengths, we test $c \in \{0.01, 0.05, 0.1, 1\}$ in each set of experiments. Since it is well known that the training performance of SVM is influenced by hyper-parameters, for linear SVM instances, we test different values for the penalty parameter $C \in \{0.1, 1, 10\}$. For nonlinear SVMs with a RBF kernel, we $C = 1$ and use different values γ in the kernel functions $\gamma \in \{0.02, 0.1, 1\}$. In each experiment instance, we use 5-fold cross validation with a 80/20 train-test split and report the average statistics for accuracy and fairness.

3.5.3 Experiment Results

We focus on the training time, namely the runtime of an algorithm before returning an optimal solution, as the main performance measure to compare the computational costs of specialized algorithms with and without fairness constraints, and off-the-shelf QP solver. For completeness, we also report accuracy and fairness statistics to verify that the tested fair SVM models provide reasonable



(a) German credit dataset (b) Adult income dataset

Fig. 3.1 Runtime comparison between Fair SMO and CVXOPT: SVMs with a RBF kernel

performances. We provide brief discussions about accuracy and fairness performances on both datasets in the captions of Tables 3.1-3.12.

Our Algorithms vs. Solver

In Figs. 3.1a and 3.1b, we compare the runtime of fair SMO versus the CVXOPT solver for training a fair non-linear SVM with a RBF kernel on both datasets. Figs. 3.2a and 3.2b provide the same comparisons between fair DCD versus the CVXOPT solver for training fair linear SVMs. We show the runtime separately for different fairness settings and thresholds. We observe that fair SMO noticeably outperforms the solver on all instances. In fact, when using the CVXOPT solver, we pre-compute and input the complete kernel matrix Q to the solver, whereas we only input the feature vectors to fair SMO; therefore, the effective advantages of fair SMO are greater than those shown in illustration. For training linear SVMs, we observe that fair DCD has a shorter runtime than the CVXOPT solver on all but the German dataset instances at $C = 10$ in Fig. 3.2a. Moreover, even in these instances, the runtime gaps are smaller than the time needed to pre-generate the kernel matrix for the solver, so fair DCD still has a shorter total runtime.

Hyperparam.	Fair. Const.	No Fairness		e = 0		e = 0:01		e = 0:05		e = 0:1		e = 1	
C : 1:0;s : 0:02	Only DP.	0.74	0.01	0.73	0.00	0.78	0.01	0.71	0.01	0.73	0.00	0.76	0.00
	Only TPP.	0.74	0.01	0.73	0.00	0.77	0.00	0.72	0.00	0.73	0.00	0.77	0.00
	DP. and TPP.	0.74	0.01	0.73	0.00	0.78	0.01	0.71	0.01	0.73	0.00	0.76	0.00
C : 1:0;s : 0:10	Only DP.	0.76	0.01	0.76	0.00	0.78	0.00	0.73	0.00	0.75	0.00	0.78	0.00
	Only TPP.	0.76	0.01	0.76	0.00	0.78	0.00	0.74	0.00	0.75	0.00	0.77	0.00
	DP. and TPP.	0.76	0.01	0.75	0.00	0.78	0.00	0.73	0.00	0.75	0.00	0.78	0.00
C : 1:0;s : 1:00	Only DP.	0.71	0.01	0.71	0.00	0.76	0.00	0.69	0.00	0.69	0.00	0.70	0.00
	Only TPP.	0.71	0.01	0.71	0.00	0.76	0.00	0.69	0.00	0.69	0.00	0.69	0.00
	DP. and TPP.	0.71	0.01	0.71	0.00	0.76	0.00	0.68	0.00	0.69	0.00	0.70	0.00

Table 3.1 Predictive accuracy on test data using non-linear SVMs with a RBF kernel on German credit dataset

Hyperparam.	Fair. Const.	No Fairness		e = 0		e = 0:01		e = 0:05		e = 0:1		e = 1	
C : 1:0;s : 0:02	Only DP.	0.15	0.01	0.08	0.03	0.13	0.02	0.09	0.02	0.11	0.03	0.11	0.02
	Only TPP.	0.15	0.01	0.13	0.02	0.16	0.02	0.11	0.01	0.14	0.03	0.14	0.02
	DP. and TPP.	0.15	0.01	0.09	0.03	0.13	0.02	0.09	0.02	0.10	0.03	0.11	0.02
C : 1:0;s : 0:10	Only DP.	0.14	0.03	0.08	0.02	0.12	0.02	0.08	0.00	0.07	0.02	0.19	0.02
	Only TPP.	0.14	0.03	0.11	0.01	0.16	0.01	0.08	0.00	0.08	0.01	0.22	0.01
	DP. and TPP.	0.14	0.03	0.09	0.02	0.12	0.02	0.08	0.00	0.07	0.01	0.19	0.02
C : 1:0;s : 1:00	Only DP.	0.03	0.01	0.01	0.00	0.01	0.00	0.02	0.01	0.04	0.00	0.04	0.01
	Only TPP.	0.03	0.01	0.01	0.00	0.01	0.00	0.02	0.00	0.04	0.00	0.06	0.00
	DP. and TPP.	0.03	0.01	0.01	0.00	0.01	0.00	0.02	0.00	0.04	0.00	0.04	0.01

Table 3.2 Demographic parity between renters vs. homeowners on test data using non-linear SVMs with a RBF kernel on German credit dataset

Hyperparam.	Fair. Const.	No Fairness		e = 0		e = 0:01		e = 0:05		e = 0:1		e = 1	
C : 1:0;s : 0:02	Only DP.	0.19	0.07	0.31	0.06	0.07	0.03	0.15	0.02	0.09	0.01	0.16	0.02
	Only TPP.	0.19	0.07	0.40	0.03	0.03	0.01	0.18	0.01	0.10	0.02	0.18	0.02
	DP. and TPP.	0.19	0.07	0.32	0.05	0.07	0.03	0.15	0.02	0.10	0.01	0.17	0.02
C : 1:0;s : 0:10	Only DP.	0.16	0.07	0.33	0.04	0.07	0.03	0.06	0.01	0.18	0.05	0.20	0.01
	Only TPP.	0.16	0.07	0.39	0.02	0.04	0.02	0.06	0.00	0.12	0.04	0.20	0.00
	DP. and TPP.	0.16	0.07	0.34	0.04	0.07	0.03	0.06	0.01	0.18	0.05	0.20	0.01
C : 1:0;s : 1:00	Only DP.	0.07	0.02	0.05	0.00	0.13	0.00	0.09	0.01	0.09	0.00	0.03	0.01
	Only TPP.	0.07	0.02	0.05	0.00	0.13	0.00	0.07	0.00	0.09	0.00	0.01	0.00
	DP. and TPP.	0.07	0.02	0.05	0.00	0.13	0.00	0.08	0.01	0.09	0.01	0.02	0.01

Table 3.3 True positive rate parity between renters vs. homeowners on test data using non-linear SVMs with a RBF kernel on German credit dataset

Hyperparam.	Fair. Const.	No Fairness		e = 0		e = 0:01		e = 0:05		e = 0:1		e = 1	
C : 1:0;s : 0:02	Only DP.	0.83	0.00	0.83	0.01	0.83	0.01	0.83	0.00	0.84	0.01	0.84	0.01
	Only TPP.	0.83	0.00	0.84	0.01	0.84	0.01	0.84	0.01	0.84	0.01	0.84	0.01
	DP. and TPP.	0.83	0.00	0.83	0.00	0.83	0.01	0.83	0.00	0.67	0.17	0.84	0.01
C : 1:0;s : 0:10	Only DP.	0.82	0.00	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01
	Only TPP.	0.82	0.00	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01
	DP. and TPP.	0.82	0.00	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01	0.83	0.01
C : 1:0;s : 1:00	Only DP.	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00
	Only TPP.	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00	0.76	0.00
	DP. and TPP.	0.76	0.00	0.76	0.00	0.76	0.00	0.61	0.15	0.76	0.00	0.76	0.00

Table 3.4 Predictive accuracy on test data using non-linear SVMs with a RBF kernel on Adult income dataset

Hyperparam.	Fair. Const.	No Fairness		e = 0		e = 0:01		e = 0:05		e = 0:1		e = 1	
C : 1:0;s : 0:02	Only DP.	0.19	0.00	0.11	0.01	0.11	0.01	0.14	0.01	0.15	0.01	0.18	0.01
	Only TPP.	0.19	0.00	0.13	0.01	0.18	0.01	0.18	0.01	0.18	0.01	0.18	0.01
	DP. and TPP.	0.19	0.00	0.12	0.01	0.11	0.01	0.14	0.01	0.12	0.03	0.18	0.01
C : 1:0;s : 0:10	Only DP.	0.21	0.00	0.15	0.01	0.16	0.01	0.17	0.01	0.18	0.01	0.19	0.01
	Only TPP.	0.21	0.00	0.16	0.01	0.19	0.01	0.19	0.01	0.19	0.01	0.19	0.01
	DP. and TPP.	0.21	0.00	0.15	0.01	0.16	0.01	0.17	0.01	0.18	0.01	0.19	0.01
C : 1:0;s : 1:00	Only DP.	0.07	0.00	0.02	0.00	0.02	0.00	0.05	0.00	0.06	0.00	0.06	0.00
	Only TPP.	0.07	0.00	0.06	0.00	0.06	0.00	0.06	0.00	0.06	0.00	0.06	0.00
	DP. and TPP.	0.07	0.00	0.02	0.00	0.02	0.00	0.04	0.01	0.06	0.00	0.06	0.00

Table 3.5 Demographic parity between female vs. male test data using non-linear SVMs with a RBF kernel on Adult income dataset

setting with true labels $\{y_i\}$ and predicted labels $\{\hat{y}_i\}$. They define the utility function as a function of $y_i, \hat{y}_i$, and the specific format is chosen to reflect whether i is risk averse, neutral or seeking, and how close the predicted outcome $\hat{y}_i$ is to i's desirable outcome. They then define a utilitarian sum of these individual utilities as the social welfare measure, and propose to add a constraint on this social welfare value to standard ML models as an in-processing fair ML approach. Hu and Chen (2020) study a similar utility definition without the risk component in a classification setup. They evaluate the overall welfare associated with classification decisions through comparing a vector of welfare values, which measure the utilitarian welfare by group. Also in a classification setting, Corbett-Davies and Goel (2018) suppose each group has fixed benefits and costs associated with classification outcomes, and these values are used as parameters in the utility functions. A group's utility aggregates the benefits and costs that individuals of the group incur from their classification outcomes. A more refined view of utility is studied in Heidari et al. (2019): they partition one's actual utility into an effort-based component and an advantage component. Utilizing this partition, they group individuals by effort-based utilities and propose a fairness measure equivalent to the expected advantage utility of the worst-off group.

4.2 The Basic Optimization Problem

The general problem of maximizing social welfare can be stated

$$\max_{\mathbf{x}} \; W\big(\mathbf{U}(\mathbf{x})\big) \quad \mathbf{x} \in S_{\mathbf{x}} \tag{4.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector of resources distributed across stakeholders $1, \ldots, n$, and $S_{\mathbf{x}}$ is the set of feasible values of $\mathbf{x}$ permitted by resource limits and other constraints. $\mathbf{U} = (U_1, \ldots, U_n)$ is a vector of utility functions, where $U_i(\mathbf{x})$ defines the utility experienced by stakeholder i as a result of the resource distribution $\mathbf{x}$. We can normally write $U_i(\mathbf{x})$ as $U_i(x_i)$, since a stakeholder's utility typically depends only on the resources allotted to that stakeholder. Finally, $W(\mathbf{u})$ is a social welfare function that measures the desirability of a vector $\mathbf{u}$ of utilities. Problem (4.1) maximizes social welfare over all feasible resource allocations.

In practice, it is often convenient to model the utility functions $\mathbf{U}$ using constraints, because this results in problems better suited for optimization solvers. One therefore writes (4.1) as

$$\max_{\mathbf{x}, \mathbf{u}} \; W'(\mathbf{u}) \quad (\mathbf{x}, \mathbf{u}) \in S_{\mathbf{xu}} \tag{4.2}$$

where $\mathbf{u}$ is a vector of utilities, and $S_{\mathbf{xu}}$ is defined so that $(\mathbf{x}, \mathbf{u}) \in S_{\mathbf{xu}}$ implies $\mathbf{x} \in S_{\mathbf{x}}$ and $\mathbf{u} = \mathbf{U}(\mathbf{x})$. The function W' is a possibly simplified version of W that yields an equivalent optimization problem due to constraints defining $S_{\mathbf{xu}}$.

To simplify exposition, we assume that the original problem constraints that define $S_{\mathbf{x}}$ consist of (or can be approximated by) a system of linear inequalities and equations. Thus, for example, when we say that (4.1) is a linear programming (LP) problem for a given W, we mean that (4.1) can be formulated as an LP problem (4.2) when $S_{\mathbf{x}}$ is defined by a linear system. The linearity assumption actually allows a great deal of modeling flexibility, because $S_{\mathbf{x}\mathbf{u}}$ can be approximated by linear constraints whenever $S_{\mathbf{x}}$ is convex and $\mathbf{U}(\mathbf{x})$ is a concave function of $\mathbf{x}$. The latter occurs in the common situation where $\mathbf{U}$ is linear or represents decreasing returns to scale.

All of the SWFs considered here can be formulated as linear, nonlinear, or mixed integer programming problems for which advanced solution technology exists. An LP model optimizes a linear function over continuous variables, subject to linear inequality constraints. The problem is extremely well solved. Nonlinear programming (NLP) models optimize a nonlinear function over continuous variables, subject to linear or nonlinear inequality constraints. All the NLP models considered here are relatively easy to solve. Mixed integer/linear programming (MILP) models are LP problems except that some variables must take integer values. They are combinatorial in nature, but state-of-the-art software frequently solves industrial instances with thousands of discrete variables.

If some of the original problem variables $\mathbf{x}$ are discrete, an otherwise LP problem becomes an MILP problem, and an NLP problem becomes a mixed integer/nonlinear programming (MINLP) problem. The latter can be quite hard to solve. An MILP problem of course remains an MILP problem.

4.2.1 Advantages of Optimization

The optimization of social welfare functions offers several advantages as a framework for incorporating fairness into AI.

- Social welfare functions provide a broader perspective on fairness than can be achieved by focusing exclusively on bias and concepts of parity across groups. They not only have the flexibility to represent a wide range of fairness concepts, but they encourage modelers to take into account the overall welfare of those affected. While AI-based decision making already strives to maximize predictive accuracy, a welfare perspective allows it to consider explicitly the more general benefits that accurate predictions can deliver, as well as whether the benefits are distributed justly.

- Social welfare functions allow one to balance equity and efficiency in a principled way. Where equity is an issue, there is often a desire for efficiency as well. A social welfare approach obliges one to consider how equity and utilitarian goals should be represented and balanced when one chooses the function to be maximized. One can of course maximize efficiency subject to a constraint on some measure of inequity, but this provides no principled way of regulating the trade-off between the two.

- Optimization models allow one to harness powerful optimization methods, which have been developed and refined over a period of 80 years or more. A wide variety of social welfare functions can be formulated for solution by highly advanced linear, nonlinear, and mixed integer programming solvers. We provide examples in Section 4.5.

- Optimization models offer enormous flexibility to include constraints on the problem. Decisions are normally made in the context of resource constraints or other limitations on possible options. These can be represented as constraints in the optimization problem, as nearly all state-of-the-art optimization methods are designed for constrained optimization. Also, a complex social welfare function can often be simplified by adding constraints to the optimization problem, resulting in a problem that is easier to solve.

4.3 Example: Mortgage Loan Processing

We use mortgage loan processing as a running example, as it is a much-discussed application of AI-based decision making. Issues of fairness arise when an AI system is more likely to deny loans to members of certain groups, perhaps reflecting minority status or gender. A frequently used remedy is to apply statistical bias metrics to detect the problem and adjust the decision algorithms in an attempt to solve it.

Yet bias is only one element of a broader decision-making context. For one thing, there is a clear utilitarian imperative. The reason for automating mortgage decisions in the first place is to predict more accurately who will default, because defaults are costly for the bank and devastating to home buyers. The desire for accurate prediction is, at root, a desire to maximize utility. Furthermore, bias is regarded as unfair in large part because it reduces the welfare of a segment of society that is already disadvantaged. An aversion to bias is, to a great degree, grounded in a desire for distributive justice in general. All this suggests that loan decisions should be designed to achieve what we really want: efficiency and distributive justice, rather than focusing exclusively on predictive accuracy and group parity.

The social welfare function W in (4.1) should be selected to balance efficiency and equity in a suitable fashion; we consider some candidate SWFs in Section 4.5. The stakeholders $1, \ldots, n$ might include the loan applicants, the bank, the bank's stockholders, and the community at large. For simplicity, we focus on the loan applicants as stakeholders. The utility function $\mathbf{U}$ converts a given set of loan decisions $\mathbf{d} = (d_1, \ldots, d_n)$ to a vector of expected utilities $\mathbf{u} = (u_1, \ldots, u_n) = \mathbf{U}(\mathbf{d})$ that the stakeholders experience as a result. Since granting a loan is a yes-or-no decision, we can define d_i to be a binary variable with $d_i = 1$ if applicant i receives a loan. The utility measure $u_i = U_i(d_i)$ for applicant i could depend on the applicant's financial situation as well as the amount of the loan, as for example when the marginal value of a loan dollar is greater for an applicant who is less well-off. The SWF can reflect a preference for granting loans to disadvantaged applicants even when they have a somewhat higher probability of default, so as to ensure a more just distribution of utility. This could have the effect of avoiding bias against minority groups, but as part of a more comprehensive assessment of social welfare.

A bank can rely solely on a fully specified social welfare optimization model to determine the optimal loan decisions; this is the standard approach in the optimization literature. The welfare optimization model can also be integrated with other AI methods to support loan decision-making. We specify several possible integration scenarios.

The loan decisions $\mathbf{d}$ and the associated utilities $\mathbf{u}$ may depend on predictions from machine learning. Utilizing historical data on loan applications, approvals and defaults, the bank can train machine learning models to predict whether an applicant is qualified for a loan or the likelihood that an applicant will default if a loan were granted, which respectively require classification and regression models. In the former case, suppose applicant i has true label $y_i = 1$ if he/she is qualified and $y_i = -1$ otherwise. The applicant's predicted label is $\hat{y}_i = 1$ if he/she is classified as a qualified applicant, and $\hat{y}_i = -1$ otherwise. One way to use these classification outcomes is to consider granting loans only for those labeled as 'qualified', namely the welfare optimization model requires $d_i = 0$ if $\hat{y}_i = -1$ and $d_i \in \{0,1\}$ if $\hat{y}_i = 1$. These constraints on $\mathbf{d}$ also affect the set of feasible utilities $\mathbf{u}$, thus the corresponding welfare-optimizing decisions. In the latter case, we denote applicant i's true default probability as p_i and the predicted probability as $\hat{p}_i$. Suppose the bank associates different utilities for when an applicant repays or defaults a loan, then the utility can be defined as $u_i = \hat{p}_i u_i^1 + (1-\hat{p}_i)u_i^0$, where u_i^1 is the utility that results if i repays the loan and u_i^0 if i defaults. As shown, the ML-based predictions are used to fully specify the bank's social welfare optimization problem. When confronted with a batch of loan decisions, the bank will use machine learning models to generate the needed predictions, then maximize $W(\mathbf{U}(\mathbf{d}))$ subject to feasibility constraints, such as, a constraint $\sum_i c_i d_i \leq B$ on the funds available (where c_i is the requested loan amount).

Another option is for the bank to solve the optimization problem in advance, before particular applicants are considered. It would maximize $W(\mathbf{U}(\mathbf{d}))$ over a set of hypothetical applicants corresponding to various financial profiles, again using ML-based default probabilities as input. In this case, the utility $U_k(d_k)$ that accrues to a potential applicant type would depend in part on the estimated number of applicants in the population that have the corresponding profile. Then when someone with financial profile k applies for a loan, the bank would award the loan if $d_k = 1$ in the optimal solution of the welfare maximization problem.

We will later refer to these mentioned options as examples of *post-processing* integration of social welfare optimization and AI, because the welfare-based fairness in injected after the learning phase with AI tools. An alternative integration perspective is to incorporate welfare-based fairness considerations into the learning components, and we refer to these as *in-processing* integration. We specify an example of in-processing integration for the case where the bank uses a classification model to predict whether an applicant is qualified, namely whether a loan application should be approved. The standard training algorithm minimizes the predictive loss, or equivalently maximizes the classification accuracy. To incorporate welfare-based fairness, utilities need to be defined with respect to classification outcomes, namely, $\mathbf{u} = \mathbf{U}(\hat{\mathbf{y}})$. We consider a linear utility format $u_i(\hat{y}_i) = b_i + g_i \hat{y}_i$, where b_i, g_i respectively denote i's starting utility and additional utility gain/loss from classification. As we later illustrate in our case study, we can assign b_i, g_i based on a person's true label y_i and other features including i's group membership. Note that b_i, g_i values should fit the classification contexts.

For example, we expect a true positive outcome to have a higher utility than a false negative outcome, that is, when $y_i = 1$, the selected $\mathbf{u}_i; g_i$ should satisfy $u_i(1) > u_i(-1)$.

4.4 Welfare-based Fairness: A General Framework

Drawing motivation from the running example, we now formalize a general framework for designing AI systems with welfare-based fairness guarantees.

4.4.1 Step 1: Specify decision problem

We begin by specifying the needed components of the decision problem. This step is critical for the success of later steps as it ensures we have a precise understanding of the problem scope and context. We highlight some key components that commonly exist in problem instances. Note that additional factors may be needed in specific problems.

- Task: the task specifies the downstream actions and the involved resources to allocate. In our running example, the bank's task is to decide whether to grant loans to applicants.
- Stakeholders: stakeholders are individuals or groups directly or indirectly affected by the decisions, namely, they are the utility recipients in the problem.
- Goals: we characterize the desirable outcomes as goals of the decision problem. These goals serve as the guiding principles for defining the social welfare objective $W(\mathbf{u})$.
- Constraints: these are restrictions in the problem context that limit which actions are feasible, namely, we specify constraints to define the domain S_d. A main source of restriction is the scarcity of resources, for example, the bank is subject to a budget constraint. In addition, the decision contexts may impose constraints on actions, for instance, the loan allocated to an applicant should not exceed the requested amount.

4.4.2 Step 2: Define utility and social welfare functions

With a clear problem statement, we continue to define utility functions and social welfare functions. As we mention in the running example, it is useful to distinguish the definitions for different components of an integrated decision-making framework. For the prediction component relying on machine learning, we define utility and social welfare with respect to the prediction results. For the decision component utilizing optimization, utility and social welfare should reflect the involved stakeholders' well-beings from the decision outcomes.

4.4.3 Step 3: Develop decision models

Depending on decision contexts, we can integrate social welfare optimization with rule-based AI systems and machine learning.

Full Information: Integration with Rule-based AI

When there is full information on the social welfare optimization problem, it is sufficient to solve the fully specified optimization model for the optimal decisions. The needed information may be available from past data, or provided by experts utilizing their domain knowledge. This full information context is suitable for integration by means of rule-based AI, which utilizes a set of rules to encode knowledge relevant to the decision and to produce pre-defined outcomes. Rule-based systems are increasingly recognized for their capacity to support principled and transparent AI in various application domains. For instance, Brandom (2018) observes the trend in autonomous vehicle industry whereby "companies have shifted to rule-based AI, an older technique that lets engineers hard-code specific behaviors or logic into an otherwise self-directed system." Moreover, Kim et al. (2021) demonstrate that ethical principles can be precisely represented as rules to include in an AI system. In fact, they suggest that a rule-based formulation is necessary for making ethical decisions.

We highlight two possible schemes to implement the integration of rule-based AI and optimization. The first method is to use the optimization problem to guide the selection of rules to encode into the AI system, then rely on the rule-based system to make decisions. As we have illustrated in the mortgage example, the bank may pre-specify applicant classes and determine decision-rules for these classes using a social welfare optimization model. Such a rule-based system is straightforward to use: for a new loan applicant, the bank would first identify which class the applicant belongs to, then approve the loan if the corresponding rule for the class says so and reject otherwise.

Alternatively, in an AI rule base, we can include rules that provide instructions for formulating the optimization problem and for choosing actions based on the optimal solution. This is consistent with the proposal from Bringsjord et al. (2006) that one could constrain AI systems with ethical principles formalized as logic statements, such as if-then statements. For example, the bank may consider rules that require applicants with certain features to receive reasonable prioritization, and these rules can be captured as constraints or incorporated into the objective function in the optimization model. Furthermore, when making the final loan decisions, the bank may define rules about implementing the allocation solution obtained from the optimization problem.

Partial Information: Integration with Machine Learning

A second type of context arises when a limited amount of information is required to formulate the relevant optimization problem. The partial information case motivates a natural integration of optimization with machine learning. In particular, we focus on supervised learning methods that train predictive models from labelled data. Suppose a training data set is $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i$ is the feature vector and y_i is the true label, then a supervised learning method trains a predictor function h with the accuracy in the predicted labels $\{h(\mathbf{x}_i)\}$ as the primary goal. The ML literature has studied a large number of formats for h, ranging from a simple functional form in logistic regression and support vector machine to more complex structures like decision tree and neural network. Optimization, as a technique, is broadly used to train ML models, but our emphasis is to integrate optimization as the fairness-seeking strategy.

We formalize two integration approaches that differ in the usage of social welfare optimization. First, in a *post-processing* view, ML models are trained to predict the information needed to formulate the welfare optimization model, then the social welfare optimization model is solved for decisions. As shown below, the prediction step focuses solely on accuracy through the standard loss minimization, then the decision step incorporates fairness considerations via the social welfare objective. It is notable that all supervised learning methods are suitable for such post-processing integration, and the decision maker has the flexibility to choose the ML methods fitting for the problem context and computational requirement.

$$\text{Prediction step: } h = \arg\min_h L(h; D);$$
$$\text{Decision step: } \mathbf{d} = \arg\max_{\mathbf{d}} \{ W(\mathbf{U}(\mathbf{d})) : d_i = d(\mathbf{x}_i; h(\mathbf{x}_i)) \}.$$

The other approach is in-processing integration where social welfare optimization is directly embedded into a machine learning model. We can consider this approach as a type of in-processing fair ML method, and the key distinction with the majority of literature is that we encode fairness in a social welfare function. More precisely, we implement the integration by modifying the standard accuracy objective in a training model with a social welfare function. As we display below, one convenient modification is to use a weighted sum of the training loss and the negative social welfare. Note that the choice of W clearly affects the complexity of the learning model, hence the success of this in-processing integration is contingent on whether the resulting training model could be solved efficiently.

$$\text{Prediction step: } h = \arg\min_h \{ L(h; D) - \lambda W(\mathbf{U}(h)) : u_i = U(h(\mathbf{x}_i); \mathbf{x}_i; y_i) \};$$
$$\text{Decision step: } \mathbf{d} = \arg\max_{\mathbf{d}} \{ W(\mathbf{U}(\mathbf{d})) : d_i = d(\mathbf{x}_i; h(\mathbf{x}_i)) \}.$$

Remark 1. We briefly discuss the integration potentials with two other core machine learning methods, unsupervised learning and reinforcement learning (RL). Fairness has been studied in both methods, but the progress is much more limited compared to fair supervised learning. Within unsupervised learning, we focus on clustering methods. We can easily apply post-processing integration to clustering methods and utilize the trained clusters as input to specify the optimization problem. For instance, in the loan example, the bank can use clustering algorithms to decide a categorization of financial profiles that will play a role in the optimization formulation. Recent works in fair clustering, e.g. Abraham et al. (2019); Deepak and Abraham (2020), have explored an in-processing strategy to extend K-means clustering to include fairness considerations by adding a fairness component to the usual K-means objective function. This indicates the potentials of in-processing integration, that is, we can define social welfare based fairness component to modify the usual clustering objective functions. In reinforcement learning, the goal is to search for a reward-maximizing policy in a dynamic environment that is typically modelled as a Markov Decision Process. Defining and achieving fairness in RL is more challenging due to the sequential and dynamic structure. Siddique et al. (2020); Weng (2019) propose a novel framework for fair multi-objective reinforcement learning based on welfare optimization. The key component of their proposal is to replace the standard reward objective with a particular social welfare function on the reward distribution. This exactly captures the perspective of in-processing integration, hence demonstrates the potentials of social welfare optimization for seeking fairness in RL.

4.5 A Sampling of Social Welfare Functions

We briefly review a collection of SWFs to illustrate how they can embody various conceptions of equity. For each, we indicate the type of optimization model it yields, and whether it is appropriate for our running example of mortgage loan processing. We classify the SWFs as pure fairness metrics, functions that combine fairness and efficiency, and statistical fairness metrics.

4.5.1 Pure fairness measures

Social welfare functions that measure fairness alone, without an element of efficiency, are of two basic types: inequality metrics and fairness for the disadvantaged.

Inequality metrics abound in the economics literature. Some simple ones are represented by the following SWFs (which negate the inequality measure):

$$W(\mathbf{u}) = \begin{cases} -(1/\bar{u})(u_{max} - u_{min}) & \text{for the relative range} \\ -(1/\bar{u})\sum_i |u_i - \bar{u}| & \text{for the relative mean deviation} \\ -(1/\bar{u})\left[(1/n)\sum_i (u_i - \bar{u})^2\right]^{\frac{1}{2}} & \text{for the coefficient of variation} \end{cases}$$

There is also the well-known Gini coefficient, which is proportional to the area between the Lorenz curve and a diagonal line representing perfect equality. It corresponds to the SWF

$$W(\mathbf{u}) = 1 - \frac{1}{2\bar{u}n^2}\sum_{i,j} |u_i - u_j|$$

Although these SWFs are nonlinear, all but the coefficient of variation have LP models. The coefficient of variation has a convex quadratic programming model with linear constraints, for which there are very efficient specialized solvers.

Other fairness-based SWFs are concerned with the lot of the disadvantaged. The Hoover index measures the fraction of total utility that would have to be transferred from the richer half of the population to the poorer half to achieve perfect equality. The SWF is

$$W(\mathbf{u}) = -\frac{1}{2n\bar{u}}\sum_i |u_i - \bar{u}|$$

The Hoover index is proportional to the relative mean deviation and can therefore be optimized using the same LP model.

The *McLoone index* compares the total utility of individuals at or below the median utility to the utility they would enjoy if all were brought up to the median utility. The index is 1 if nobody's utility is strictly below the median and approaches 0 if there is a long lower tail. The SWF is

$$W(\mathbf{u}) = \frac{1}{|I(\mathbf{u})|\tilde{u}} \sum_{i \in I(\mathbf{u})} u_i$$

where $\tilde{u}$ is the median of utilities in $\mathbf{u}$ and $I(\mathbf{u})$ is the set of indices of utilities at or below the median. The McLoone index can be optimized in an MILP model.

The Hoover and McLoone indices measure only the relative welfare of disadvantaged parties, and not their absolute welfare. The *maximin* criterion addresses both. It is based on the Difference Principle of John Rawls, which states that inequality should exist only to the extent it is necessary to improve the lot of the worst-off (Freeman (2003); Rawls (1999); Richardson and Weithman (1999)). It can be plausibly extended to a lexicographic maximum principle. The SWF is simply

$$W(\mathbf{u}) = \min_i \{u_i\}$$

and has an LP model.

Purely fairness-oriented SWFs can be used when equity is truly the only issue of concern. In particular, they are unsuitable for the mortgage problem, where overall utility is a prime consideration.

4.5.2 Combining fairness and efficiency

Several SWFs combine equity and efficiency, sometimes with a parameter that regulates the relative importance of each. Perhaps the best known is *alpha fairness*, for which the SWF is

$$W_\alpha(\mathbf{u}) = \begin{cases} \dfrac{1}{1-\alpha} \sum_i u_i^{1-\alpha} & \text{for } \alpha \geq 0;\ \alpha \neq 1 \\ \sum_i \log(u_i) & \text{for } \alpha = 1 \end{cases}$$

Larger values of α imply a greater emphasis on equity, with $\alpha = 0$ corresponding to a pure utilitarian criterion $\sum_i u_i$, and $\alpha = \infty$ to a pure maximin criterion. An important special case is $\alpha = 1$, which corresponds to *proportional fairness*, also known as the *Nash bargaining solution*. It is widely used in telecommunications and other engineering applications. Both proportional fairness and alpha fairness have been given axiomatic and bargaining justifications (Binmore et al. (1986); Harsanyi (1977); Lan et al. (2010); Nash (1950); Rubinstein (1982)). The alpha fairness SWF is irreducibly nonlinear, but because it is concave for all α, it can be maximized with reasonable efficiency by NLP methods.

Alpha fairness is conceptually a reasonable choice for the mortgage problem, because the bank can obtain any desired balance between utility and fairness by adjusting α. While it is difficult to justify to stakeholders any particular choice for the value of α, a perceived bias against minorities can always be addressed by increasing α. On the other hand, the presence of 0–1 variables x_i produces an MINLP model, which can be hard to solve. Thus alpha fairness may be practical only for problems with at most a few hundred applicants.

The *Kalai-Smorodinsky* (K–S) bargaining solution, proposed as an alternative to the Nash bargaining solution, minimizes each person's relative concession. That is, it provides everyone the largest possible utility relative to the maximum one could obtain if other players are disregarded, subject to the condition that all persons receive the same fraction β of their maximum. In addition to the bargaining justification of Kalai and Smorodinsky (1975), this approach has been defended by Thompson (1994) and is implied by the contractarian philosophy of Gauthier (1983). The SWF can be formulated

$$W(\mathbf{u}) = \begin{cases} \sum_i u_i, & \text{if } \mathbf{u} = \beta \mathbf{u}^{\max} \text{ for some } \beta \text{ with } 0 \le \beta \le 1 \\ 0, & \text{otherwise} \end{cases}$$

where $u_i^{\max} = \max_{(\mathbf{x},\mathbf{u}) \in S_{x\mathbf{u}}} u_i$ for each i. It can be optimized by maximizing β subject to $\mathbf{u} = \beta \mathbf{u}^{\max}$ and $\beta \le 1$, an easy LP problem.

The K–S criterion cannot be used for the mortgage problem, because the role of 0–1 variables in the problem almost ensure that the optimization model will be infeasible. Since $u_i^{\max} = \bar{u}_i^1$, we must have $U_i(x_i) = \beta u_i^1$ for all i. But $U_i(x_i) = \bar{v}_i + (\bar{u}_i - \bar{v}_i)x_i$, which means that there must be a ratio β that, for each i, is equal to either $\bar{v}_i/u_i^1$ or $\bar{u}_i/u_i^1$ (which correspond to setting $x_i = 0$ or $x_i = 1$, respectively). It is very unlikely that the problem data will have this property.

Williams and Cookson (2000) suggest two *threshold* criteria for combining maximin and utilitarian objectives in a 2-person context. One uses maximin until the cost of fairness becomes too great, whereupon it switches to utilitarianism, and the other does the opposite. Hooker and Williams (2012) generalize the former to n persons by proposing the following SWF:

$$W_\Delta(\mathbf{u}) = (n-1)\Delta + \sum_{i=1}^{n} \max\{u_i - \Delta, u_{\min}\}$$

where $u_{\min} = \min_i\{u_i\}$. The parameter Δ regulates the equity/efficiency trade-off in a way that may be easier to interpret in practice than the α parameter: parties whose utility is within Δ of the lowest utility receive special priority. Thus the disadvantaged are favored, and Δ defines who is disadvantaged. As with the α parameter, $\Delta = 0$ corresponds to a purely utilitarian criterion and $\Delta = \infty$ to a maximin criterion. Hooker and Williams provide an MILP model of the SWF and show that it is sharp (i.e., its continuous relaxation describes the convex hull of its feasible set). Partly for this reason, they found that the model solves rapidly in computational tests.

This threshold approach is a reasonable choice for the mortgage problem. Since the problem has discrete variables regardless of the SWF used, the MILP-based threshold formulation adds relatively little complexity to the problem. In addition, loan officers can specify in a meaningful way when an applicant is to be considered disadvantaged, by selecting an appropriate value of Δ.

One weakness of the model is that the actual utility levels of disadvantaged parties other than the very worst-off have no effect on the measurement of social welfare, as long as those utilities are within Δ of the lowest. As a result, the socially optimal solution may not be as sensitive to equity as one might desire. Chen and Hooker (2020a,b) address this issue by combining utilitarianism with a *leximax* rather than a maximin criterion. A leximax (lexicographic maximum) solution is found by first maximizing the lowest utility, then while holding it fixed, maximizing the second lowest utility, and so forth. Chen and Hooker combine leximax and utilitarian criteria by maximizing a sequence of threshold SWFs that have tractable MILP models. Their approach may yield more satisfactory solutions of the mortgage problem.

4.5.3 Statistical bias metrics

While we argue that bias metrics afford an overly narrow perspective on fairness, they nonetheless can be expressed as SWFs if desired. The utility vector $\mathbf{u}$ becomes simply a binary vector in which $u_i = 1$ if individual i is selected for some benefit, and $u_i = 0$ otherwise. In the mortgage example, the benefit is a mortgage loan. We set constant $a_i = 1$ when person i actually qualifies for selection (as for example when person i in the mortgage training set repaid the loan), and $a_i = 0$ otherwise. Two groups are compared, respectively indexed by N and N'. One is a protected group, such as a minority subpopulation, and the other consists of the rest of the population.

For example, *demographic parity* has the SWF

$$W(\mathbf{u}) = 1 - \left| \frac{1}{|N|}\sum_{i\in N} u_i - \frac{1}{|N'|}\sum_{i\in N'} u_i \right|$$

Equalized odds can be measured in two ways, one of which is *equality of opportunity*:

$$W(\mathbf{u}) = 1 - \left| \frac{\sum_{i\in N} a_i u_i}{\sum_{i\in N} a_i} - \frac{\sum_{i\in N'} a_i u_i}{\sum_{i\in N'} a_i} \right|$$

Another SWF represents *accuracy parity*:

$$W(\mathbf{u}) = 1 - \left| \frac{1}{|N|}\sum_{i\in N} \big(a_i u_i + (1-a_i)(1-u_i)\big) - \frac{1}{|N'|}\sum_{i\in N'} \big(a_i u_i + (1-a_i)(1-u_i)\big) \right|$$

and still another *predictive rate parity*:

$$W(\mathbf{u}) = 1 - \left| \frac{\sum_{i\in N} a_i u_i}{\sum_{i\in N} u_i} - \frac{\sum_{i\in N'} a_i u_i}{\sum_{i\in N'} u_i} \right|$$

The computational challenge varies widely across the various bias-oriented SWFs. The first three SWFs above give rise to linear models (which become MILP models due to the 0–1 restriction on u_i), while the last produces an extremely difficult nonconvex MINLP model.

Bias measures are inappropriate as social welfare objectives for the mortgage problem, because they take no account of ef ciency. One can, of course, maximize predictive accuracy subject to constraints on the amount of bias, but this has a number of drawbacks:

- As previously argued, it provides a very limited perspective on the utility actually created by decisions. Indeed, the utility vector consists only of 0–1 choices.
- There is no consensus on which bias measure is suitable in a given context, if any. Bias measures were developed by statisticians to measure predictive accuracy, not to assess fairness.
- There is no principle for balancing equity and ef ciency. If equity is one of the objectives, it should be part of the objective function. The choice of that function obliges one to justify the equity/ef ciency trade-off mechanism in a transparent manner.
- Bias measurement forces one to identify *a priori* which individuals in a training set should be selected for bene ts (as indicated by a_i). In a social welfare approach, no prior decisions of this kind are necessary.
- Bias measurement forces one to designate "protected groups" (as indicated by the index set N). There is no clear principle for selecting which groups should be protected, unless one is content simply to recognize those mandated by law.

4.6 Case Study: Mortgage Loan Approval

4.7 Conclusion

We formalize a general framework for using optimization to incorporate welfare-based fairness into AI applications. The framework provides a guideline for formulating a decision task into a social welfare optimization problem. In particular, we illustrate how optimization can be integrated with rule-based AI systems and machine learning models. By expanding the fairness problem to the optimization of social welfare functions, one can achieve a broader perspective on fairness that are driven by the well-beings of stakeholders and characterize the broader fairness concepts in a principled way. Optimization models also provide the exibility of adding constraints on resources and other problem elements, while harnessing the power of highly advanced optimization solvers.

We conclude the paper by outlining a research program to explore some key questions related to the framework.

- There is a wide gap between the presented general formalization of integration strategies and practical implementations of integrated methods. For integration with rule-based AI, one important direction is to investigate how to build ethics-sensitive rule bases to t into different social welfare optimization scenarios. Previous works on formulating ethics principles into rules, e.g. Bringsjord et al. (2006); Kim et al. (2021), may provide guidance for this direction. For integration with machine learning, future research could explore the in-processing perspective and study how to de ne social welfare functions to use as the objective in machine learning models. The modi ed objective functions need to have a format that can be ef ciently trained, and the trained models need to provide the desirable fairness and welfare guarantees.

- Although optimization solvers have been developed over decades, not all classes of optimization models are readily solvable by state-of-the-art software. Among all classes, linear programming and convex programming problems can be considered tractable up to reasonably large sizes, but non-convex formulations including some mixed integer programming problems are more restricted. For practical use of social welfare optimization models, one may need to apply available computational strategies or design problem-speci c heuristics to speed up solving the optimization problems.
- The social welfare functions we consider are of a static nature, that is, a SWF does not attempt to capture potential dynamics in the utilities. A SWF takes utility values as the input, and the function values characterize the associated static utility distributions. While such a static view is often suf cient and reasonable for a one-shot decision problem, a dynamic perspective may be required in sequential decision problems where decisions need to be made repeatedly and the selected actions have incremental impacts on the long term social welfare. Future research could explore how to extend the presented optimization based framework to t a dynamic view of welfare and fairness. Although this is not a trivial task, there are many well-developed techniques to utilize, such as, stochastic optimization, Markov decision process, etc.

Chapter 5

Online Convex Optimization Perspective for Learning from Dynamically Revealed Preferences

5.1 Introduction

Preferences of an agent implicitly dictates his/her actions, and in uence for example what a company should offer as its products or how a company should personalize recommendations to an individual customer (agent). This creates incentives for the company/central decision maker to learn the preferences of their agents. Nevertheless, in reality, the true preferences of the agents are often private to the individual agents and are only implicitly revealed in the form of their behaviors/actions to the central decision maker. Such typical interactions for example include a streaming platform suggesting a number of videos to a user and tracking whether the user watches or likes them. As evident from such scenarios, inferring the agents' preference information through agent interactions and observations of their behaviors is a critical task for the decision makers in such settings.

A common assumption adopted to formalize the problem of learning from revealed preferences is that rational agents are utility maximizers, that is, they choose actions to maximize their utility functions subject to a set of restrictions. The central decision maker interacting with the agents is the learner. An important learner-centric goal is to design schemes for the learner to extract useful information on the agents' utility functions. This learning point of view of revealed preferences has been explored in a broad range of literature from economics (e.g., Beigman and Vohra (2006); Varian (2006)), machine learning (e.g., Balcan et al. (2014); Dong and Zeng (2020); Dong et al. (2018b)) and operations research (e.g., Ahmadi et al. (2020); Bärmann et al. (2017); Mohajerin Esfahani et al. (2018)). Based on the type of learner-agent interactions and information feedback, such preference learning schemes vary in information requirement, preference elicitation objective and learning complexity.

In this chapter, we focus on a specific setup where the learner seeks to learn the utility function of a non-strategic agent while receiving information about the agent's actions in an online fashion. This setup fits naturally in applications where the agents benefit from effective learning of their true preferences. For example this is the case when a streaming platform interacts with its users to learn their preferences. In this example, in a typical interaction, the platform recommends videos to a user and the user takes actions based on the recommendations. User actions, e.g., clicks, movie streaming, etc., are fully observable to the platform and they reflect the user's true preferences.

5.1.1 Our Approach and Results

We propose a novel modeling framework for learning from dynamically revealed preferences via online data-driven inverse optimization. In our setup, the learner monitors a sequence of data signals and observes the respective rational decisions of a non-strategic agent without noise over a finite time horizon of T time steps. The learner operates and receives information in an online fashion, and updates an estimate θ_t of θ_{true} using newly available information at each time step.

Our framework is enabled by the identification of ℓ^{sim}, a loss function that is both simple in structure and has practical connections with conventional loss functions from the inverse optimization literature. Online inverse optimization based on ℓ^{sim} is a online convex optimization (OCO) problem, which enjoys the flexibility to be handled by any deterministic OCO algorithm.

- In Section 5.2, we present a formal description of our problem setting. Section 5.2.1 introduces the agent's problem and discusses a rather broad decomposable structure assumption on the agent's utility functions that is capable of representing all of the utility functions studied in the data-driven inverse optimization literature as well as other key utility functions. Section 5.2.2 describes the learner's inverse optimization problem that minimizes a given loss function $\ell(\cdot)$ to obtain an accurate estimate θ of the hidden parameter θ_{true}. We then state the online inverse optimization framework in Section 5.2.3: we describe the sequence of events and define regret as the performance measure.
- In Section 5.3, we utilize our structural assumption on the agent's utility function to design a new convex loss function, namely simple loss ℓ^{sim}.
- We establish in Section 5.4 that in the noiseless setting, a bounded regret with respect to ℓ^{sim} also guarantees a bounded regret with respect to all the other loss functions; see Proposition 5 and Corollary 1. We also briefly discuss the noisy setting.
- Convexity and simplicity of ℓ^{sim} enables us to use an online convex optimization (OCO) framework (see Section 5.5) that offers the flexibility to use different OL algorithms, such as, online Mirror Descent (MD) utilizing a first-order oracle (Section 5.5.1) and implicit OL based on a solution oracle (Section 5.5.2). In the noiseless setup, our framework equipped with online MD covers all of the problem classes studied in the online data-driven inverse optimization

literature, and matches the corresponding state-of-the-art regret bounds with respect to all of the loss functions in a unified manner. In particular, our results immediately generalize the customized algorithms from Bärmann et al. (2017) and completely bypass the requirement to verify the rather technical assumptions of Dong et al. (2018a) and the need to use their expensive MISOCP-based solution oracle; see Section 5.5.3 for a detailed comparison discussion.

- Our numerical study in Section 5.6 highlights that when compared to the ℓ^{pre}-based implicit OL with an MISOCP solution oracle approach of Dong et al. (2018a), ℓ^{sim}-based OL algorithms equipped with a first-order oracle or a solution oracle, particularly online MD, demonstrate significant advantages in terms of both the learning performance (i.e., regret bounds) and the computation time. This is directly in line with our theoretical results. Moreover, these results seem to be fairly robust with respect to changes in the structure of the agent's domain as well as the noise in observations.

5.1.2 Related Literature

Varian (2006) is one of the earliest and most celebrated work for learning from revealed preferences in the economics literature. They study constructing utility functions of the agent to explain a sequence of her/his observed actions. Nevertheless, this approach has a main shortcoming—a utility function capable of explaining past actions not necessarily also guarantees accurate predictions of the future actions. Consequently, Beigman and Vohra (2006) have initiated a new line of research to learn utility functions capable of predicting future actions with statistical performance guarantees. Beigman and Vohra (2006) examine a statistical setup where the learning algorithm takes as input a batch of observations and is evaluated by its sample complexity guarantees. Zadimoghaddam and Roth (2012) focus on the setting where the agent has a linear or linearly separable concave utility function, and propose learning algorithms with polynomially bounded sample complexity. Balcan et al. (2014) identify a connection between the problem of learning a utility function and the structured prediction problem of D-dimensional linear classes. Through this connection, Balcan et al. (2014) suggest an algorithm for learning utility functions that is superior (in terms of sample complexity) than the method from Zadimoghaddam and Roth (2012) in the case of linear utility functions and is also applicable for learning separable piecewise-linear concave functions and CES functions with explicit sample complexity bounds.

As an alternative to this statistical view, Balcan et al. (2014) study a query-based learning model, where the learner aims to recover the exact utility function by querying an oracle for the agent's optimal actions. The query-based models consider an online feedback mechanism where the learner receives one observation of the agent's action at a time. When the learner has the power to choose which observation to receive from the query oracle, Balcan et al. (2014) give exact learning algorithms for several classes of utility functions. There is a recent research stream on learning to optimize the learner's objective function based on information from revealed preferences of the agents. In this stream it is often assumed that the learner has similar power on the selection of the observations. For example, Amin et al. (2015) and Ji et al. (2018) propose algorithms for finding the profit-maximizing prices for a seller, who has price controlling power and learns buyer preferences by observing the buying behavior at different price levels. Roth et al. (2016) and Dong et al. (2018b) consider the learning task as Stackelberg games, where the leader player is the learner and a follower player is a strategicagent with incentive to manipulate actions and hide information.

We note two restrictions with the problem setup in these fore-mentioned papers. First, the assumption that the learner can choose observations is not always achievable in practice. A more realistic setup is accommodated by the data-driven inverse optimization view where the learner does not control the sequence of observations. Second, when the learner is optimizing an objective function that does not explicitly measure how well s/he is learning about the agent, the approaches that are effective for choosing the learner's objective-optimizing action provide no guarantees on the quality of the learned agent information.

Inverse optimization generalizes the query-based view and offers a natural abstraction of learning from revealed preferences. This approach is typically used in settings with non-strategic agents, in which the agents have no incentive to hide information from the learner, and thus an agent's decisions reveal her/his true preferences. In this setting, the learner's goal is to recover unknown parameters of an agent's utility function from the observations of her/his true optimal solution. Chan et al. (2021) provides a comprehensive review of inverse optimization. We next summarize key developments in the literature, with an emphasis on two topics that are more relevant to this paper: data-driven inverse optimization and online inverse learning.

Early studies on inverse optimization examine the setting where the agent's optimization problem is fixed, see e.g., Ahuja and Orlin (2001); Heuberger (2004); Iyengar and Kang (2005); Schaefer (2009). Unfortunately, this classical setup is limited in its practical applicability as it ignores uncertainty in the environment. A new thread of research on data-driven inverse optimization studies a flexible setup, where the learner observes the agent's optimal or sub-optimal decisions corresponding to varying external data signals. In the noiseless case, that is, when observations of optimal solutions/agent actions are available, Keshavarz et al. (2011) show that data-driven inverse optimization of convex programs is polynomial time solvable. In the case of noisy observations, Aswani et al. (2018) proves that such problems are NP-hard in general.

Data-driven inverse optimization is further categorized based on whether observations are given as a batch upfront or in an online manner. In the batch setup, Keshavarz et al. (2011) study the inverse optimization of identifying the unknown affine weights in a convex objective function that is an affine combination of pre-selected basis convex functions. Recent work of Aswani et al. (2018) and Mohajerin Esfahani et al. (2018) in the batch setup investigates the inverse optimization of general convex programs without the basis function structure. Aswani et al. (2018) adopt the prediction loss ℓ^{pre}, which measures the difference between the observed agent action and the predicted agent action through squared norm distance, as the inverse optimization objective. They formulate the inverse problem into a bilevel program using Lagrangian duality, and present two heuristic algorithms with approximation guarantees for solving the bilevel formulation. Mohajerin Esfahani et al. (2018) use suboptimality loss ℓ^{sub}, which is defined as the difference between objective values at the observed agent action and the predicted action, as their loss function and provide a distributionally robust formulation of the inverse problem. Batch setup requires that the learner receives observations of the agent's actions all at once. However, obtaining a large batch of observations all at once as well as learning from such a batch often presents operational and computational challenges. In practice, such strong batch feedback is rare as the learner often interacts with the agent repetitively in a dynamic environment.

A recent stream of research Bärmann et al. (2017); Dong et al. (2018a) adopts a dynamic information acquisition setup and studies the online data-driven inverse optimization where the learner observes a stream of the agent's actions one by one in an online fashion. Both Bärmann et al. (2017) and Dong et al. (2018a) suggest OL algorithms and measure their performance via the regret, i.e., the difference between the losses incurred from online estimates of the unknown parameters in the agent's utility function and the offline optimal estimate. Bärmann et al. (2017) consider the problem of learning the linear utility function of an agent given the noiseless online observations of the agent's actions in a dynamic environment. They propose two specialized OL algorithms with first-order oracles that both achieve a bound of $O(\sqrt{T})$ on the sum of the suboptimality loss ℓ^{sub} and the estimate loss ℓ^{est} after T periods but lacks regret guarantees with respect to the prediction loss ℓ^{pre}. Dong et al. (2018a) consider the setup, where the learner wishes to learn an unknown linear component of an agent's quadratic objective function from noisy observations. By utilizing the implicit OL framework of Kulis and Bartlett (2010) equipped with a Mixed Integer Second Order Cone Program (MISOCP)-based solution oracle, they provide a regret bound of $O(\sqrt{T})$ with respect to the prediction loss ℓ^{pre} after T periods whenever ℓ^{pre} is convex. Dong et al. (2018a) present a number of rather technical assumptions that guarantee the convexity of ℓ^{pre}, however these assumptions are not only difficult to verify but also quite restrictive. In fact, in Dong et al. (2018a), these were shown to hold only for a specific class of convex quadratic problem.

Notation. We let $\mathbb{R}^n_+$ be the set of nonnegative n-dimensional vectors. For a given vector v, we use v_i to denote its i-th element. We let $[n] := \{1,\dots,n\}$, and we use $\{a_i\}_{i\in[n]}$ to represent a collection of entries, such as vectors, functions, etc., indexed with $i \in [n]$. For a differentiable function f, we use $\nabla f(x)$ to denote the gradient of f at x. For a nondifferentiable function f, we use $\partial f(x)$ to denote the subdifferential of f at x.

5.2 Problem Setting

In our setting, the learner monitors a sequence of external signals $\{u_t\}_{t\in[T]} \subseteq \mathbb{R}^k$ and observations $\{y_t\}_{t\in[T]} \subseteq \mathbb{R}^n$ of the agent's respective optimal decisions over a finite time horizon of T time steps.

the agent's problem is given by $\min\{\min_x\{x : x \in [-1,1], x \neq 0\}, 0+\theta\cdot(-1)\} = \min\{-1, -\theta\}$. Then, we deduce that the agent's optimal solution will satisfy the following:

- When $\theta < 0$, agent's optimal solution is $x(\theta) = -1$;
- When $\theta > 0$, $x(\theta) = -1$ if $\theta < 1$, and $x(\theta) = 0$ if $\theta \geq 1$ (note that in the case of alternative optima, we assume that the solver will break ties by selecting the solution with smaller norm);
- When $\theta = 0$, $x(\theta) = -1$.

To summarize, in the given domain Θ, when $\theta \in [-3,1)$, $x(\theta) = -1$ and $c(x(\theta)) = 0$; when $\theta \in [1,3]$, $x(\theta) = 0$ and $c(x(\theta)) = -1$.

We next show that implicit OL based on ℓ^{pre} with a solution oracle may lead to an unbounded regret. Suppose we choose $\eta_t = \frac{1}{t}$ for all t. Then, at time step t, based on the implicit OL based on ℓ^{pre}, we update θ_{t+1} by solving the following optimization problem: in this example, $\ell_t^{pre}(\theta) = \|x(\theta_{true}) - x(\theta)\|^2 = (-1-x(\theta))^2$, hence

$$\theta_{t+1} = \arg\min_{\theta \in [-3,3]} \frac{1}{2}(\theta-\theta_t)^2 + \frac{1}{t}(-1-x(\theta))^2.$$

If we initialize $\theta_1 = 3$, then the above update will generate $\theta_2 = \arg\min_{\theta \in [-3,3]} \frac{1}{2}(\theta-3)^2 + \frac{1}{t}(-1-x(\theta))^2$. To decide the optimal solution, we need to compare three scenarios: when $\theta = \theta_1$, the objective value is $0 + \frac{1}{t}(-1-x(\theta_1))^2 = \frac{1}{t}(-1-0)^2 = \frac{1}{t}$; when $\theta < 1$, the objective value is $\frac{1}{2}(\theta-\theta_1)^2 + \frac{1}{t}(-1-x(\theta))^2 = \frac{1}{2}(\theta-3)^2 + 0 \geq \frac{1}{2}(1-3)^2 = 2 > \frac{1}{t}$ (where we used $x(\theta) = -1$ for $\theta < 1$); when $1 \leq \theta < \theta_1$, the objective value is $\frac{1}{2}(\theta-\theta_1)^2 + \frac{1}{t}(-1-x(\theta))^2 = \frac{1}{2}(\theta-3)^2 + \frac{1}{t}(-1-0)^2 > \frac{1}{t}$. Therefore, we have $\theta_2 = \theta_1$, and by the same derivation, later iterations will always stay at the same estimate $\theta_t = \theta_1$. This means the implicit OL algorithm will generate $\theta_t = 3$ for all t, and each iteration the learner incurs prediction loss as $\ell_t^{pre}(\theta_t) = \|x(\theta_{true}) - x(\theta_t)\|^2 = \|x(0) - x(3)\|^2 = 1$. Therefore, the associated regret with respect to ℓ^{pre} is unbounded as $T \to \infty$:

$$R_T(\{\ell_t^{pre}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]}) = \sum_{t\in[T]} \ell_t^{pre}(\theta_t) - \sum_{t\in[T]} \ell_t^{pre}(\theta_{true}) = T.$$

We note that this example does not invalidate the regret convergence in Theorem 20. With the contrived definition of $c(x)$, the loss function ℓ^{pre} does not satisfy the Lipschitz continuity assumption needed for regret convergence guarantees given in Theorem 20. To be more specific, $\ell_t^{pre}(\theta) = (x(\theta_{true}) - x(\theta))^2$: consider $\epsilon > 0$, $\theta_1 = 1$; $\theta_2 = 1+\epsilon > 1$, we conclude $\ell_t^{pre}(\theta_1) = (-1-(-1))^2 = 0$ and $\ell_t^{pre}(\theta_2) = (-1-0)^2 = 1$. As $\epsilon \to 0$, there is no finite G as a valid Lipschitz constant for ℓ_t^{pre}.

We also examine the use of online Mirror Descent (MD) based on ℓ^{sim} in the same setup. Let Euclidean distance be the distance generating function in Bregman distance, then online MD simplifies to projected gradient descent. We again choose $\eta_t = \frac{1}{t}$ for all t, then at time step t we update θ_{t+1} via

$$\theta_{t+1} = \text{proj}_{[-3,3]}\left(\theta_t - \frac{1}{t}\left(c(x(\theta_{true})) - c(x(\theta_t))\right)\right).$$

frameworks for solving robust convex optimization in Ben-Tal et al. (2015); Ho-Nguyen and Kılınç-Karzan (2018); Ho-Nguyen and Kılınç-Karzan (2019). For our particular application, in Section 5.3, we design linear functions $\{\ell_t^{sim}(\theta)\}_{t \in [T]}$ to which we apply OCO algorithms. While using non-arbitrary classes of loss functions does not invalidate any guarantees from deterministic OCO methods such as online MD, more caution is needed for OCO algorithms which involve randomness and provide guarantees on *expected regret* such as stochastic gradient descent. This is due to the fact that the design of the loss functions in specific applications may create undesirable dependence among random variables and invalidate certain steps used in the analysis of stochastic OCO algorithms. Hence, here we will focus on deterministic OCO algorithms.

5.3 Loss Functions

Loss function $\ell(\theta)$ plays a key role in the formulation of the inverse problem (5.3). Since the learner's goal is to mimic the agent's true action with the estimated preference θ, the appropriate loss functions should reflect how close the prediction $x(\theta; u)$ is to $x(\theta_{true}; u)$ at a given signal u. The following are common loss functions used in inverse optimization context (recall that f is the agent's forward objective in (5.1) and $x(\theta; u_t)$ is the optimal solution to (5.2) for given θ, u_t):

- Prediction loss: $\ell^{pre}(\theta; x(\theta; u_t); y_t; u_t) := \|y_t - x(\theta; u_t)\|^2$,
- Suboptimality loss: $\ell^{sub}(\theta; x(\theta; u_t); y_t; u_t) := f(y_t; \theta; u_t) - f(x(\theta; u_t); \theta; u_t)$, and
- Estimate loss: $\ell^{est}(\theta; x(\theta; u_t); y_t; u_t) := f(x(\theta; u_t); \theta_{true}; u_t) - f(y_t; \theta_{true}; u_t)$.

These functions use the observation y_t as a proxy of the true action $x(\theta_{true}; u_t)$. Under perfect information with $y_t = x(\theta_{true}; u_t)$, ℓ^{pre} directly compares the distance between the true action and the predicted action. ℓ^{sub} and ℓ^{est} utilize the agent's objective function. ℓ^{sub} measures how much y_t would affect the agent's optimal objective value at estimate θ; ℓ^{est} measures how much $x(\theta; u_t)$ would change the observed agent's objective value. Under imperfect information, due to the noises present in y_t, the loss values can be split into two components, one reflects the difference between $x(\theta; u_t)$ and $x(\theta_{true}; u_t)$, and the other one is dependent on the noise shifting $x(\theta_{true}; u_t)$ to y_t.

Within data-driven inverse optimization in a batch setup, ℓ^{pre} is used in Aswani et al. (2018) and ℓ^{sub} is used in Mohajerin Esfahani et al. (2018) (see Section 5.1.2). In the online inverse optimization setup, ℓ^{sub} and ℓ^{est} are studied by Bärmann et al. (2017) under the assumption that f is linear in x, and ℓ^{pre} by Dong et al. (2018a) when the forward problem is a quadratic program with a special structure. The OL algorithms from these latter two papers are customized for the chosen loss functions and forward problem structure, indicating the lack of a unified general framework.

We introduce the following shorthand notation.

$$\ell_t^{pre}(\theta) := \ell^{pre}(\theta; x(\theta; u_t); y_t; u_t); \quad \ell_t^{sub}(\theta) := \ell^{sub}(\theta; x(\theta; u_t); y_t; u_t); \quad \ell_t^{est}(\theta) := \ell^{est}(\theta; x(\theta; u_t); y_t; u_t):$$

Assumption 2 ensures that ℓ_t^{sim} is a convex function of θ, and thus any deterministic OCO algorithm will be applicable for regret minimization with respect to $\{\ell_t^{sim}\}_{t\in[T]}$. Then, as a consequence of Proposition 5 (and Corollary 1), such algorithms will also be minimizing regret with respect to the loss functions $\ell_t^{sub}, \ell_t^{est}$ (and ℓ_t^{pre}), as well.

Remark 4. The regret bounds with respect to these loss functions have the following implications under perfect information. A sublinear regret bound with respect to ℓ^{sim} implies that the average loss incurred by the estimates $\{\theta_t\}$ approaches the offline optimal loss over time. Sublinear regret bounds with respect to ℓ^{sub} and ℓ^{est} indicate that the learner is able to generate estimates $\{\theta_t\}$ that lead to vanishing errors in the predicted agent's objective function values. A sublinear regret bound with respect to ℓ^{pre} additionally indicates that the average $\|\cdot\|_2$-norm distance between the predicted agent's action and her/his true action decreases to zero over time. Note that none of these regret guarantees in particular ensures that the $\{\theta_t\}$ generated from the online learning process are good approximations of θ_{true}. In general, this is an overly ambitious task as (Bärmann et al., 2017, Example 3.2) has shown a simple case where the exact recovery of θ_{true} cannot be guaranteed. We note that stronger performance guarantees, such as $\frac{1}{T}\sum_{t\in[T]}\|\theta_t - \theta_{true}\| \to 0$ may be possible for special cases, for example, when $x(\theta; u_t)$ has a closed form expression as a continuous function of θ. In addition, in certain cases the optimal actions from the forward problem may be non-unique, our framework is not aiming to predict the chosen action $x(\theta_{true}; u_t)$, instead, we measure the learning performance with regret values based on the objective value of the agent.

5.4.2 Imperfect Information

The case when the learner has access to only imperfect information about the agent's actions is of natural interest as well. Mohajerin Esfahani et al. (2018) identify two types of noisy information as of interest: (i) *measurement noise*, that is, for all $t \in [T]$, the learner observes $y_t = x(\theta_{true}; u_t) + \epsilon_t$ with ϵ_t denoting a random noise, and (ii) *bounded rationality*, which means for all $t \in [T]$, the agent may choose a sub-optimal action instead of $x(\theta_{true}; u_t)$. Such imperfect information does not affect the convexity property of loss functions, and so both ℓ^{sub} and ℓ^{sim} remain convex (see Lemma 3 and 4) still enabling the use of OCO algorithms for regret minimization with respect to these loss functions. For instance, we can still apply a ℓ^{sim}-based OCO algorithm to minimize regret with respect to ℓ^{sim}.

However, since y_t is no longer guaranteed to be a minimizer of (5.1) with $u = u_t$, Proposition 5 does not hold in general, and consequently $R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]})$ is not guaranteed to bound the regrets with respect to the other loss functions. In addition, due to the noise in y_t, for any of the four loss definitions, its associated regret $R_T(\{\ell_t\}_{t\in[T]}; \{\theta_t\}_{t\in[T]})$ can no longer accurately measure learning performance with respect to the agent's true actions and objective values.

- ω-center: $\theta_\omega := \arg\min_{\theta \in \Theta} \omega(\theta)$.
- Set width: $\Omega := \max_{\theta \in \Theta} V_{\theta_\omega}(\theta) \le \max_{\theta \in \Theta} \omega(\theta) - \min_{\theta \in \Theta} \omega(\theta)$.

When functions $\{\ell_t^{sim}(\theta)\}_{t \in [T]}$ are convex in θ, online MD as stated in (Ho-Nguyen and Kılınç-Karzan, 2019, Algorithm 1) is applicable to guarantee a sublinear regret bound on $R_T(\{\ell_t^{sim}\}_{t \in [T]}; \{\theta_t\}_{t \in [T]})$, which further bounds regrets with respect to the other loss functions as discussed in Section 5.4.

Theorem 18. (Ho-Nguyen and Kılınç-Karzan, 2019, Theorem 1) Suppose Θ is convex and $\ell_t : \Theta \mapsto \mathbb{R}$ is a convex function for $t \in [T]$. Suppose there exists $G \in (0, \infty)$ such that all the subgradients s_t of ℓ_t are bounded, i.e., $\max_{s_t \in \partial \ell_t(\theta)} \|s_t\|_* \le G$ for all $\theta \in \Theta$ and $t \in [T]$. Let the step size η_t be chosen as $\eta_t = \sqrt{\frac{2\Omega}{G^2 T}}$. At time step t, using the online Mirror Descent algorithm, we generate θ_{t+1} as

$$\theta_{t+1} := \mathrm{Prox}_{\theta_t}(\eta_t s_t) = \arg\min_{\theta \in \Theta} \{\langle \eta_t s_t, \theta \rangle + V_{\theta_t}(\theta)\}, \tag{5.5}$$

where $s_t \in \partial \ell_t(\theta_t)$. Then the sequence $\{\theta_t\}_{t \in [T]}$ satisfies $R_T(\{\ell_t\}_{t \in [T]}; \{\theta_t\}_{t \in [T]}) \le \sqrt{2\Omega G^2 T}$.

In applying Theorem 18 to the loss functions $\{\ell_t^{sim}\}_{t \in [T]}$, we have the subgradient $s_t = c(y_t) - c(x(\theta_t; u_t))$. So, it suffices to set $G \ge 2\max\{\|c(x)\|_* : x \in X\}$. The set width Ω depends on Θ only and can be computed for a given Θ and Bregman distance explicitly.

5.5.2 Implicit Online Learning with a Solution Oracle

We next review the implicit OL with a solution oracle from Dong et al. (2018a). This algorithm was first introduced in its general form in Kulis and Bartlett (2010).

The *implicit online learning* algorithm computes

$$\theta_{t+1} := \arg\min_{\theta \in \Theta} L_t(\theta), \tag{5.6}$$

where $L_t(\theta) = V_{\theta_t}(\theta) + \eta_t \ell_t(\theta)$ and $V_\theta(\theta')$ is the Bregman distance, η_t is a step size. This approach does not rely on the first-order oracle $\nabla \ell_t$ but rather assumes the existence of a *solution oracle* to solve (5.6). Kulis and Bartlett (2010) establish the following regret bound on the OL using implicit update (5.6).

Theorem 19. (Kulis and Bartlett, 2010, Theorem 3.2.) Suppose Θ is convex, and $\ell_t : \Theta \mapsto \mathbb{R}$ is a convex and differentiable function for $t \in [T]$. Let θ^* be the offline optimal solution to $\min_{\theta \in \Theta} \sum_{t \in [T]} \ell_t(\theta)$. For any $0 < \alpha_t \le \frac{L_t(\theta_{t+1})}{L_t(\theta_t)}$ for $t \in [T]$, for any step size $\eta_t > 0$, an implicit OL algorithm with the update rule (5.6) attains

$$R_T(\{\ell_t\}_{t \in [T]}; \{\theta_t\}_{t \in [T]}) \le \sum_{t \in [T]} \frac{1}{\eta_t} \left((1 - \alpha_t)\eta_t \ell_t(\theta_t) + V_{\theta_t}(\theta^*) - V_{\theta_{t+1}}(\theta^*) \right). \tag{5.7}$$

When ℓ_t is a convex and Lipschitz continuous function of θ and the domain Θ has a finite width with respect to the selected Bregman divergence, the regret bound (5.7) further results in a $O(\sqrt{T})$ bound on $R_T(\{\ell_t\}_{t\in[T]},\{\theta_t\}_{t\in[T]})$.

Theorem 20. Suppose Θ is convex, and for each $t\in[T]$, $\ell_t:\Theta\mapsto\mathbb{R}$ is a convex function of θ that is uniformly Lipschitz continuous with parameter G, and suppose $\max_{\theta_1,\theta_2\in\Theta} V_{\theta_1}(\theta_2)\le\hat{W}$. Then, by choosing $\eta_t=\frac{\sqrt{\hat{W}}}{G}\frac{1}{\sqrt{t}}$ for $t\in[T]$, an implicit OL algorithm with the update rule (5.6) attains

$$R_T(\{\ell_t\}_{t\in[T]},\{\theta_t\}_{t\in[T]})\le 2\sqrt{\hat{W}G^2T}. \tag{5.8}$$

To apply Theorem 20, we can choose the Lipschitz parameter G by definition. For instance, with the loss functions $\{\ell_t^{sim}\}_{t\in[T]}$, we have $|\ell_t^{sim}(\theta)-\ell_t^{sim}(\theta')|=|\langle\theta-\theta', c(y_t)-c(x(\theta_t,u_t))\rangle|\le\|\theta-\theta'\|\|c(y_t)-c(x(\theta_t,u_t))\|$, hence $G\ge 2\max\{\|c(x)\|: x\in X\}$ suffices. Alternatively, with $\{\ell_t^{pre}\}_{t\in[T]}$, $|\ell_t^{pre}(\theta)-\ell_t^{pre}(\theta')|=|\langle x(\theta',u_t)-x(\theta,u_t), 2y_t-x(\theta',u_t)-x(\theta,u_t)\rangle|\le\|x(\theta',u_t)-x(\theta,u_t)\|\|2y_t-x(\theta',u_t)-x(\theta,u_t)\|$, so we need additional information about $x(\theta,u_t)$ to decide a suitable G. The set width $\hat{W}$ only depends on Θ and the Bregman distance definition.

5.5.3 Comparison with the Existing Approaches

Bärmann et al. (2017) study online inverse optimization under perfect information where the agent's objective f is a bilinear function of θ and x, i.e., $f(x,\theta)=\langle\theta,x\rangle$. They suggest using the online gradient descent and the Multiplicative Weights Update (MWU) algorithms to generate $\{\theta_t\}_{t\in[T]}$ estimates and show via separate analysis that the resulting estimates have vanishing average losses with respect to ℓ^{est} and ℓ^{sub} (at the rate $O(1/\sqrt{T})$) but do not present their regret bounds or analyze ℓ^{pre} loss. Note that both online gradient descent and MWU algorithm are simply special cases of the online MD algorithm customized to the geometry of the problem domain. Moreover, the setting studied in Bärmann et al. (2017) clearly satisfies our Assumption 2 and the perfect information assumption, hence we can utilize our ℓ^{sim}-based OL framework equipped with online MD and directly derive average regret bounds of $O(1/\sqrt{T})$ on ℓ^{sim}, ℓ^{sub} and ℓ^{est}. In addition, as opposed to the simple bilinear form of f considered in Bärmann et al. (2017), our framework can handle more general functions f in the forward problem when $f_1(x,u)$ and/or $f_2(\theta,u)$ are nontrivial. In this respect, it is of interest to study the case of strongly convex $f_1(x,u)$, where through Corollary 1, our framework also leads to regret bound with respect to ℓ^{pre}.

Dong et al. (2018a) study the following problem where f is linear in θ and strongly convex in x

$$\min_x\left\{\tfrac{1}{2}x^\top Px-\langle\theta_{true},x\rangle : x\in X(u)\right\}. \tag{5.9}$$

Here, P is a positive definite matrix and $X(u)$ is the agent's feasible domain determined by the external signal fixed as u. In this setting, Dong et al. (2018a) propose a regret minimization algorithm utilizing the implicit OL method (Kulis and Bartlett (2010)) with a nonconvex MISOCP oracle. They focus on the prediction loss ℓ^{pre}, and establish an $O(\sqrt{T})$ bound on $R_T(\{\ell_t^{pre}\}_{t\in[T]};\{\theta_t\}_{t\in[T]})$ whenever $\ell_t^{pre}(\theta)$ is a convex function of θ. A main limitation of their approach is that the convexity of ℓ^{pre} does not hold in general. Although they identify a technical sufficient condition (Dong et al., 2018a, Assumption 3.3) that can guarantee convexity of ℓ^{pre}, they also remark that this condition is restrictive and very hard to verify in practice even for the simplest form of problem classes. In fact, the only example they identify as satisfying their assumption is when the agent's optimization problem is (5.9) and the set $\mathcal{X}(u)$ must *always* contain the minimizer of the unrestricted objective minimization problem, i.e., $-P^{-1}\theta_{true} \in \mathcal{X}(u)$ for all possible u.

When the agent's problem has the specific form of (5.9), the algorithm from Dong et al. (2018a) updates θ_{t+1} as the optimal solution of the following bilevel program:

$$\theta_{t+1} := \arg\min_{\theta\in\Theta}\left\{\frac{1}{2}\|\theta-\theta_t\|^2+\eta_t\|y_t-x(\theta;u_t)\|^2 : x(\theta;u_t)\in\arg\min_x\left\{\frac{1}{2}x^\top Px-\langle\theta,x\rangle : x\in\mathcal{X}(u_t)\right\}\right\}.$$

It was shown in Dong et al. (2018a) that when the feasible domain $\mathcal{X}(u_t)$ is polyhedral, this bilevel program can be represented as a MISOCP. Consequently, the implicit OL algorithm of Dong et al. (2018a) utilizes an MISOCP based solution oracle to generate $\{\theta_t\}_{t\in[T]}$. The main convergence result (Dong et al., 2018a, Theorem 3.2) proves that under their assumptions by choosing the step size $\eta_t \propto 1/\sqrt{t}$, the sequence of estimates $\{\theta_t\}_{t\in[T]}$ generated with the above update yields a $O(\sqrt{T})$ bound on the regret $R_T(\{\ell_t^{pre}\}_{t\in[T]};\{\theta_t\}_{t\in[T]})$.

Note that the format of in (5.9) satisfies our Assumption 2, and consequently ℓ_t^{sim} is guaranteed to be convex for any $\mathcal{X}(u)$. Therefore, our OCO framework based on minimizing regret for loss functions $\{\ell_t^{sim}\}_{t\in[T]}$ is applicable to (5.9). In addition, in the perfect information setting, through Proposition 5 and Corollary 1, our framework can provide regret bounds with respect to all of ℓ^{sim}, ℓ^{est}, ℓ^{sub} and ℓ^{pre}, without further structural assumptions on the agent's domain. In contrast, the implicit OL approach of Dong et al. (2018a) for minimizing regret with respect to ℓ^{pre} requires additional conditions on the agent's domain (see (Dong et al., 2018a, Assumptions 3.1, 3.2, 3.3)) in order to guarantee a regret bound. In particular, it is specifically focused on ℓ^{pre} and provides no insight on other performance measures of interest captured by ℓ^{est} and ℓ^{sub} either. Moreover, online MD in our framework uses a much simpler (and computationally faster) first-order oracle in contrast to the expensive MISOCP oracle in the implicit OL approach of Dong et al. (2018a). One aspect that Dong et al. (2018a) emphasize is the noise in observations: they prove theoretical expected regret bounds with respect to ℓ^{pre} when y_t is a noisy observation of $x(\theta_{true};u_t)$, but the theoretical guarantees of our framework based on ℓ^{sim} do not readily extend to the imperfect information setup. We caution that their analysis for the noisy setup is subject to the restrictive conditions needed to ensure convexity of $\ell_t^{pre}(\theta)$, so it might not provide meaningful performance guarantees in general.

5.6 Computational Study

We perform numerical experiments on a practical application that is motivated by a company (learner) seeking to learn about its customer's (agent's) preferences in a changing market. We assume the customer is a rational decision maker, and in any given market situation, her/his action reflects accurately her/his optimal preferences. These experiments do not aim to provide structural insights on specific instances, rather, our main purpose is to demonstrate the performance of ℓ^{sim}-based OCO algorithms from various aspects and the comparison with an alternative ℓ^{pre}-based approach in Dong et al. (2018a).

We first focus on the case when perfect information is available, i.e., there is no noise in learner's observations of the agent's optimal actions, and address three main questions. First, are there notable performance differences among OL algorithms based on different oracles? Second, how do the algorithm performances vary in terms of different loss functions? Third, does the structure of the agent's feasible region affect complexity of the learning problem and the algorithm performances? While discussing these questions, we also compare against existing algorithms from the literature.

In the second part of our numerical study, we examine the robustness of these OL algorithms under imperfect information, i.e., when there is random noise to the learner's observations of the agent's optimal actions. Recall that in the imperfect information setup, our OL based approach is not guaranteed to provide low regret guarantees, so these experiments essentially shed light to their empirical performance in the noisy setup. Despite the lack of theoretical bounds, we can analyze the OL algorithms to explain the observed empirical trends in losses and regrets (see Appendix B.4).

All algorithms are coded in Python 3.8, and Gurobi 8.1.1 with default settings is used to solve the mathematical programs needed for the subproblems associated with the corresponding oracles. We limit the solution time of each mathematical program to be at most 3600 seconds. We have not hit this imposed time limit in any of our experiments. All experiments are conducted on a server with 2.8 GHz processor and 64GB memory.

5.6.1 Problem Instances

We consider a market with n products that evolves over a finite time horizon T, e.g., the product prices change. These changes consequently impact the agent's feasible actions; in this case, agents are customers interested in purchasing the products. For each $t \in [T]$, we let u_t denote the market parameters relevant to the agent's decisions at period t. When constraint parameters are fixed as u_t, an agent's action $x(q_{true}; u_t)$ is an optimal solution to an optimization problem parametrized by u_t and q_{true}, where q_{true} captures the agent's preferences over the products. We model the agent's optimization problem as a maximization of her/his utility function subject to feasibility constraints. The learner knows the agent's decision problem up to the parameter vector q_{true}, and the learner's goal is to estimate q_{true} using observations of the agent's actions y_t in response to the market conditions u_t at each period $t \in [T]$.

We study two different forms for the agent's utility function.

(a) For direct comparison with Dong et al. (2018a), we examine the case where the agent's utility function has the quadratic form (5.9), i.e., the agent's action $x(q_{true}; u_t)$ is given by

$$x(q_{true}; u_t) := \arg\max_x \left\{ -\frac{1}{2} x^\top P x + \langle q_{true}, x \rangle : x \in X(u_t) \right\}, \tag{5.10}$$

where $P \in S^n_{++}$ is a fixed positive definite matrix known by both the learner and the agent and $X(u_t)$ represents the domain for the agent's feasible actions determined by the market parameters u_t.

(b) We also examine a second setup where the agent has a CES utility function with $r = 2$. Hence, in period t, the agent's action $x(q_{true}; u_t)$ is given by

$$x(q_{true}; u_t) := \arg\max_x \left\{ \sum_{i \in [n]} (q_{true})_i x_i^2 : x \in X(u_t) \right\}. \tag{5.11}$$

Note that this setup with a CES utility has not been previously studied in an OL framework.

These particular forms of utility functions in (5.10) and (5.11) imply that the dimensions of q and x are the same, i.e., $p = n$. Moreover, observe that both of the objective functions in (5.10) and (5.11) satisfy Assumption 2, and thus in both cases $\ell_t^{sim}(q)$ is convex in q.

To identify the impact of agent's feasible region on the complexity of the problem and on the performance of the learning algorithms, we experiment on a variety of settings for $X(u_t)$.

(i) Continuous knapsack domain: in this setting, we impose only a budget constraint on the agent: $X(u_t) = X^{ck}(p_t, b_t) := \{x \in \mathbb{R}^n_+ : \langle p_t, x \rangle \leq b_t\}$, where the parameters $p_t \in \mathbb{R}^n_+$ correspond to the product prices and $b_t \in \mathbb{R}_+$ is the budget available to the customer during time period t. Note that both p_t and b_t can vary in each time period $t \in [T]$.

(ii) Continuous polytope domain: here, we generalize the continuous knapsack domain and model general resource constraints resulting in a polytope as the feasible region: $X(u_t) = X^{cp}(A_t, c_t) = \{x \in \mathbb{R}^n_+ : A_t x \leq c_t\}$, where all the parameters are nonnegative.

(iii) Binary knapsack domain: in this case, we again impose a single budget constraint, but also require that the agent's action is a binary vector: $X(u_t) = X^{bk}(p_t, b_t) := \{x \in \{0, 1\}^n : \langle p_t, x \rangle \leq b_t\}$.

(iv) Equality constrained knapsack domain: that is, $X(u_t) = X^{eck}(p_t, b_t) := \{x \in \mathbb{R}^n_+ : \langle p_t, x \rangle = b_t\}$.

We ran experiments with the utility function (5.10) where we choose the matrix P to be a positive definite diagonal matrix and generate each of its diagonal entries P_{ii} by first drawing a number from $[1, 21]$ uniformly and then normalizing the drawn vector $(P_{11}, \ldots, P_{nn})$ to have a unit ℓ_1-norm, and we also set the domain to be $X^{ck}(p_t;b_t)$, $X^{cp}(A_t;c_t)$, or $X^{bk}(p_t;b_t)$. In the case of CES utility function (5.11), for implementation simplicity, we use instances with the domain $X^{eck}(u_t)$.

In all of our experiments, we consider a market with $n = 50$ goods. We compare OL algorithms by running $T = 500$ iterations on a batch of 50 randomly generated instances for each setting. The domain Θ is set be a unit simplex, i.e., $\Theta = \{\theta \in \mathbb{R}^n_+ : \sum_{i\in[n]} \theta_i = 1\}$. We follow the same instance generation methodology used in (Bärmann et al., 2017, Section 4.1) for generating the true parameter θ_{true} and the agent's domain $X(u_t)$. In each instance θ_{true} is obtained by drawing a random sample from a uniform distribution over $[1,1000]^n$ and then normalizing the sampled vector to have a unit ℓ_1-norm. In the case of $X^{ck}(p_t;b_t)$, $X^{bk}(p_t;b_t)$, and $X^{eck}(p_t;b_t)$, for all $t \in [T]$, the constraint parameters p_t, b_t are generated randomly as follows: p_t is set as $\theta_{true} + 100 \cdot 1_n + r$, where r is an integer vector sampled from discrete uniform distribution over the collection of integer vectors in $[-10,10]^n$ (numpy.random.randint function is used). The budget b_t is selected uniformly random from the range $[1, \sum_{i=1}^n (p_t)_i]$. In the case of continuous polytope domain $X^{cp}(A_t;c_t)$, we choose A_t as an $m \times n$ matrix with $m = 10$, where each row of A_t is generated in the same way as p_t, and each coordinate i in the vector c_t is drawn uniformly random from $[1, \sum_{j=1}^m (A_t)_{ji}]$.

In the OL setup, at time step t, the learner observes the signal u_t and the agent's action, and uses the information revealed so far to construct the estimate θ_{t+1}. Under perfect information, we have $y_t = x(\theta_{true};u_t)$ for all t; under imperfect information, we assume $y_t = x(\theta_{true};u_t) + \epsilon_t$, where ϵ_t denotes the random noise. In the noisy setup, each coordinate of ϵ_t is randomly drawn from a uniform distribution over a given range. We consider two ranges to simulate small and large noises, and we choose the range bounds based on the average $\delta = \frac{1}{T}\sum_{t\in[T]} \|x(\theta_{true};u_t)\|$: the small noises are generated with the range $[-\delta/n, \delta/n]$, and the large noises are generated with $[-\delta, \delta]$.

5.6.2 Implementation Details

In order to compute the estimates $\{\theta_t\}_{t\in[T]}$, we implement three OL algorithms and compare their performances. By taking advantage of the convexity of ℓ^{sim}, we design two OL algorithms minimizing regret with respect to ℓ^{sim}: one equipped with a first-order oracle and one with a solution oracle. In addition, for comparison with the literature, we implemented another implicit OL algorithm with a solution oracle aimed to minimize the regret associated with ℓ^{pre}, i.e., the one from Dong et al. (2018a) that utilizes an MISOCP solution oracle. We provided precisely the same dynamic observations, i.e., the realizations of signals u_t along with the agent's optimum action $x(\theta_{true};u_t)$ in each iteration $t \in [T]$ to all of these OL algorithms.

In the case of OL with the first-order oracle, because Θ is a unit simplex, we use the negative entropy function $\omega(\theta) = \sum_{i=1}^n \theta_i \ln(\theta_i)$ as the distance generating function in the definition of Bregman distance $V_{\theta_t}(\theta)$. Then, the update rule (5.5) for the OL with first-order oracle is given explicitly by the following formula, where $s_t(\theta_t)$ is a subgradient of $\ell_t^{sim}(\theta)$ at θ_t.

$$(\theta_{t+1})_i = \frac{(\theta_t)_i \exp(-\eta_t (s_t(\theta_t))_i)}{\sum_{j=1}^n (\theta_t)_j \exp(-\eta_t (s_t(\theta_t))_j)}, \quad \text{for all } i \in [n].$$

The main challenge in the implementation of implicit OL algorithm with a solution oracle is whether one can design a computationally tractable solution oracle. When the loss function $\ell_t(\theta)$ used in the implicit OL involves $x(\theta; u_t)$, as is the case in all loss functions from Section 5.3 except $\ell_t^{\text{sim}}(\theta)$, (5.6) is a bilevel program. Bilevel programs are difficult to solve in general, but can be reformulated into a single level problem using KKT conditions of the inner level problem whenever the inner level is a convex problem. In contrast to this, when ℓ_t^{sim} is used as the loss function in an implicit OL algorithm, (5.6) becomes a single level optimization problem in θ and thus the solution oracle becomes much simpler. Consequently, we study two variants of the implicit OL algorithm based on ℓ^{sim} and ℓ^{pre} that are necessarily equipped with different solution oracles.

In the first variant, we design an implicit OL algorithm to minimize the regret with respect to ℓ^{sim}. Using the squared Euclidean norm as the d.g.f., we arrive at the implicit OL algorithm with a solution oracle that updates θ_{t+1} as the optimal solution to

$$\theta_{t+1} \coloneqq \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + h_t \ell_t^{\text{sim}}(\theta).$$

Under Assumption 2, $\ell_t^{\text{sim}}(\theta)$ is convex in θ, and when the domain Θ is convex, the above problem is a convex program. Therefore, the implementation requires only a convex solution oracle; see Appendix B.3 for the explicit formulations of these oracles.

For comparison purposes, we implement a second variant of the implicit OL algorithm minimizing regret with respect to the loss function ℓ^{pre}. By following the same approach taken in Dong et al. (2018a), we use the squared Euclidean norm as the d.g.f., and the resulting solution oracle updates θ_{t+1} with the following bilevel program, where the inner level computes $x(\theta; u_t)$ used in $\ell_t^{\text{pre}}(\theta)$:

$$\theta_{t+1} \coloneqq \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + h_t \ell_t^{\text{pre}}(\theta).$$

When the agent is maximizing a concave objective function over a polyhedral domain $X(u_t)$, we can reformulate the above bilevel program into a mixed integer program (MIP).

Consequently, at time t, this ℓ^{pre}-based implicit OL algorithm requires a nonconvex oracle given by the MIP formulation to obtain θ_{t+1}. In the case of (5.10), it was demonstrated in Dong et al. (2018a) that when the domain $X(u_t)$ is polyhedral, the MIP reformulation admits a nice MISOCP structure due to the quadratic objective. For completeness, we provide the MISOCP reformulation of this solution oracle in Appendix B.3. Note that due to the advanced capabilities of modern MIP solvers, the resulting MISOCP still remains computationally tractable whenever the scale of the agent's problem is relatively small.

On the other hand, when the domain of the inner problem $X(u_t)$ is nonconvex, e.g., when we consider $X^{\text{bk}}(p_t; b_t)$ that involves binary variables, or when the agent maximizes a nonconcave function over a convex domain $X(u_t)$ as in the case of (5.11) for $\theta \notin \mathbb{R}^n_+$, we no longer have access to KKT based optimality certificates for the inner problem. Consequently, in such cases, we do not know how to design a computationally tractable solution oracle, and this is an open question. Therefore, we did not experiment with the ℓ^{pre}-based implicit OL algorithm in these cases.

5.6.3 Perfect Information Experiments

In this section, we discuss our numerical results along with plots that highlight our key observations pertinent to the questions of interest to the perfect information case listed at the beginning of Section 5.6.

Learning a Quadratic Utility Function

In this case, we assume that the agent's utility function is of form (5.10). We first compare the performance of the three OL approaches in terms of both average regret performance and the solution time. Figures 5.1 and 5.2 display the means of average (expected) regret performance of the iterates $\{q_t\}_{t\in[T]}$ returned by the OL algorithms with respect to all five loss functions of interest for the instances where the agent's domain is of continuous knapsack and polytope type, respectively; the shaded areas indicate 95% confidence interval for the means. These means are computed based on all 50 random instances generated in the experiment. In Figures 5.1 and 5.2 (and all the later ones as well), the scale of loss functions naturally differ because the associated regret and loss values are evaluated with respect to different terms present in their corresponding loss definitions. In terms of the rate at which the average regret converges, in the case of the continuous knapsack instances, Figure 5.1 shows that regardless of the loss function used to evaluate the performance, all three OL algorithms have quite similar performances. This empirical observation is in line with the theoretical regret guarantees given in Section 5.5; recall that this particular domain type was the focus of Dong et al. (2018a), and their analysis presents some restrictive assumptions guaranteeing convergence of their approach on this type of instances. For the continuous polytope instances, Figure 5.2 demonstrates similar performances from the two OL algorithms based on ℓ^{sim}, but highlights the drastically different performance of the implicit OL with the ℓ^{pre}-minimizing solution oracle, which now leads to average regrets converging to non-zero values. Recall from Section 5.5.2, the regret convergence of an implicit OL algorithm with a solution oracle requires the convexity of the selected loss function. In fact, Dong et al. (2018a) adopt further strong assumptions on $X(u_t)$ to guarantee that ℓ^{pre} is a convex function of q when the agent's problem is of form (5.10) with $X(u_t) = X^{ck}(p_t; b_t)$. Our empirical results indicate that these assumptions are indeed hard to satisfy in general and our randomly generated continuous polytope instances do not necessarily satisfy their required assumption. In contrast, since ℓ^{sim} is guaranteed to be a convex function of q when the agent's problem is of form (5.10) regardless of the structure of the agent's domain $X(u_t)$, the average regrets of the ℓ^{sim}-based implicit OL algorithm with the solution oracle converge to zero for instances with polytope domain as well. Furthermore, we note that the regret convergence of the ℓ^{sim}-based implicit OL algorithm with the solution oracle is slightly better than the OL with the first-order oracle in both types of instances.

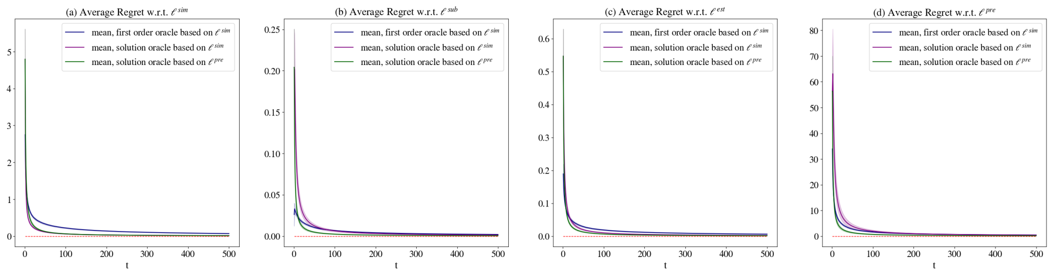


Fig. 5.1 Means of average regret with respect to different loss functions over $T = 500$ iterations for continuous knapsack instances; the shaded region is 95% confidence interval for the means.

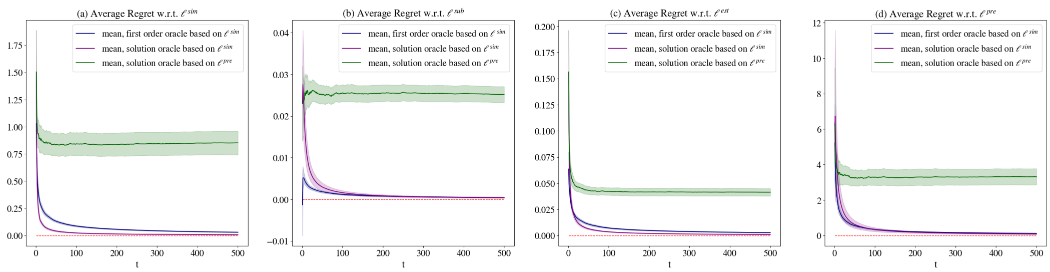


Fig. 5.2 Means of average regrets with respect to different loss functions over $T = 500$ iterations for continuous polytope instances; the shaded region is 95% confidence interval for the means.

In our numerical study, we observe almost no variation in terms of the solution time of the OL algorithms across different random instances generated from the same setting. We thereby report the time spent by all three OL algorithms on a randomly selected instance from our problem set. When computing the solution time at iteration t, we always ignore the time taken to compute $x(\theta_{true}; u_t)$. In iteration t of the OL with the first-order oracle, we account for the time to compute $x(\theta_t; u_t)$ and generate θ_{t+1} using the first-order oracle. Lastly, in each iteration of both of the ℓ^{sim}- and ℓ^{pre}-based implicit OL algorithms with a solution oracle, we account for the time used by the corresponding solution oracles in updating θ_{t+1}. For an arbitrary instance with the continuous knapsack domain, OL with the first-order oracle finishes in about 0.08 seconds, ℓ^{sim}-based implicit OL with the solution oracle takes 2.03 seconds, and ℓ^{pre}-based implicit OL with the solution oracle takes 146 seconds. These highlight that, by a significant margin, our OL algorithms minimizing regret with respect to the loss function ℓ^{sim} and utilizing the first-order oracle and the solution oracle are much more computationally efficient than the ℓ^{pre}-based implicit OL with the MISOCP solution oracle one from Dong et al. (2018a).

We next analyze whether the agent's domain structure has any visible effect on the overall regret performance of the OL with the first-order oracle. From Figures 5.1 and 5.2, we observe that the superiority of the OL with first-order oracle in terms of the average regret is slightly more obvious in the continuous knapsack setting than in the polytope setting. In Figure 5.3, we compare the means of average regrets for the continuous knapsack instances versus the binary knapsack instances. The regret performances with respect to the loss functions ℓ^{sub} and ℓ^{est} seem to vary only slightly when

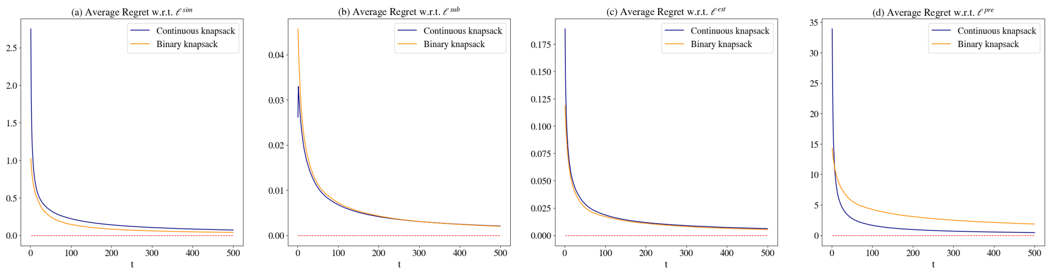


Fig. 5.3 Means of average regret with respect to different loss functions contrasting continuous knapsack instances with binary knapsack instances, when OL with the first-order oracle is used.

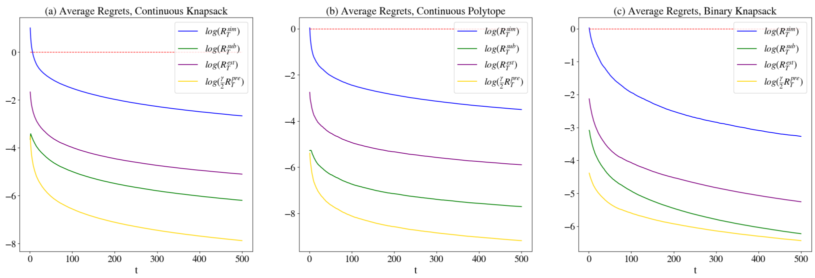


Fig. 5.4 Means of average regret (on a logarithmic scale) with respect to different loss functions over $T = 500$ iterations for (a) continuous knapsack instances, (b) continuous polytope instances, (c) binary knapsack instances, when OL with first-order oracle is used.

the agent's domain type changes from continuous knapsack to binary knapsack; yet these differences are slightly more noticeable in the case of loss functions ℓ^{pre} and ℓ^{sim}.

Lastly, we examine the regret performance of OL with the first-order oracle with respect to different loss functions $\ell(q)$. From the experiment results from continuous knapsack and continuous polytope instances (respectively Figures 5.1 and 5.2), we observe that the average regret with respect to any of the four loss functions convergences roughly at the same rate, but the corresponding regret bounds differ in their scales. This is not surprising, as the corresponding regrets are based on different terms, e.g., norms of solutions or objective function values, etc. Moreover, recall from Section 5.4 that in the perfect information case the following relationship among the regret bounds with respect to different loss functions (here for simplicity in notation, we denote $R_T^{sim} := R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{q_t\}_{t\in[T]})$, etc.) holds: $R_T^{sim} \geq R_T^{sub} + R_T^{est} \geq \frac{\gamma}{2} R_T^{pre} \geq 0$, where γ is the strong convexity parameter of the quadratic objective function in (5.10). Recall that our instance generation guarantees $P \in S^n_{++}$, i.e., its smallest eigenvalue $\lambda_{min}(P) > 0$, and then by the definition of strong convexity, we deduce $\gamma = \lambda_{min}(P)$. Figure 5.4 displays (on a logarithm scale) the means of the average regrets for different loss functions for estimates q_t generated from the OL with the first-order oracle on instances in which the agent's domain is either a continuous knapsack, polytope, or a binary knapsack type. These results also confirm the theoretical relationship among the regrets for different loss functions we have established in Section 5.4.

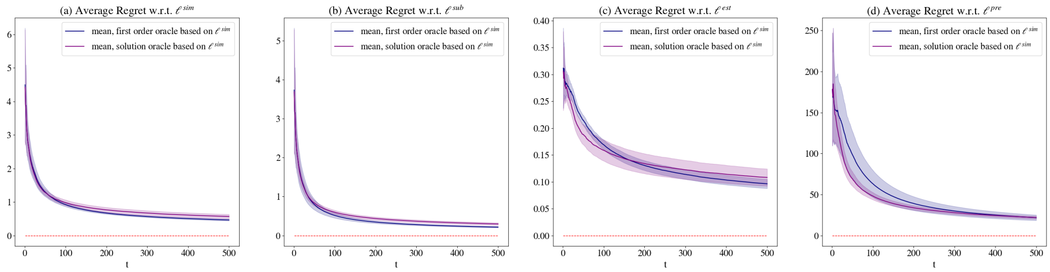


Fig. 5.5 Means of average regrets with respect to different loss functions over $T = 500$ iterations for equality constrained knapsack instances; the shaded region is 95% confidence interval for the means.

Learning a CES Utility Function

Here, we examine the case when the agent's utility function is of form (5.11) and summarize our findings on the average regrets in Figure 5.5. We note that the OL with the first-order oracle has a quite noticeable advantage over the implicit OL with a solution oracle in terms of the regret convergence. In this case, on a typical instance, OL with the first-order oracle takes 0.12 seconds to complete and ℓ^{sim}-based implicit OL with the solution oracle takes $.92$ seconds.

5.6.4 Imperfect Information Experiments

We next study the performance of the two ℓ^{sim}-based OL algorithms when the observations are corrupted with random noise. We test this imperfect information setup on two types of instances where (1) the agent is maximizing a concave quadratic utility function on a continuous knapsack domain, and (2) the agent is maximizing a CES utility function over an equality constrained knapsack domain. We observed that the impact of the noises on the solution time of the OL algorithms was negligible in both of these instance types.

We plot the outcomes differently from the perfect information case. We still show the average regret with respect to ℓ^{sim} to illustrate that the ℓ^{sim}-based OL algorithms remain valid under noises. For ℓ^{sub} and ℓ^{est}, we report the difference between the average loss incurred from $\{q_t, x(q_t; u_t)\}$ and the average loss from $\{q_{true}, x(q_{true}; u_t)\}$, where the latter evaluates the loss due to imperfect information. Lastly, we plot the average squared norm distance between $\{x(q_t; u_t)\}$ and $\{x(q_{true}; u_t)\}$ as a measure of prediction loss.

We report the results for when the agent's problem has the form (5.10) with the domain $\mathcal{X}(u_t) = \mathcal{X}^{ck}(u_t)$ in Figures 5.6 and 5.7. First, we observe that under small noises, the average regret with respect to ℓ^{sim} has a similar convergence trend as in the perfect information experiment results plotted in Figure 5.1, and the convergence appears to be slower under large noises. For the three loss based measures, both OL algorithms lead to decreasing trends; in particular, the decreasing average squared norm distance between the predicted actions and the true actions indicates that these OL algorithms' predictions of the agent's actions are becoming more accurate as T increases. Not surprisingly,

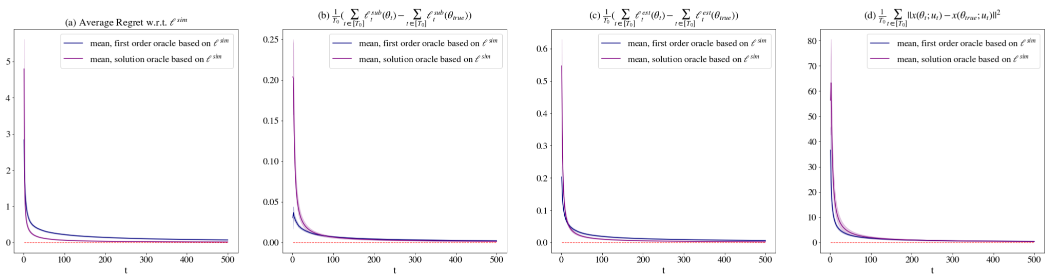


Fig. 5.6 Learning a quadratic utility function under small noises: means of selected performance measures over $T = 500$ iterations for continuous knapsack instances; the shaded region is 95% confidence interval for the means.

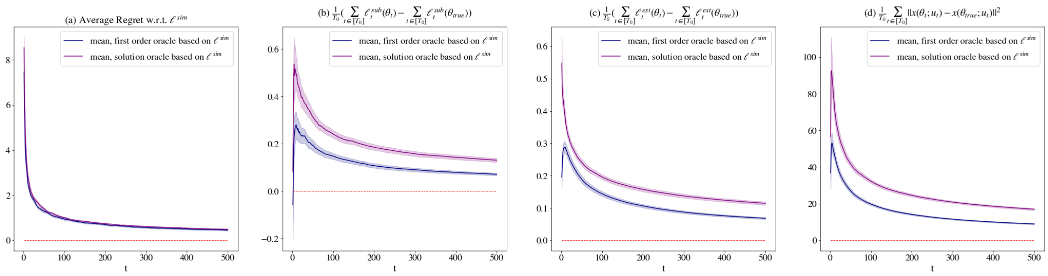


Fig. 5.7 Learning a quadratic utility function under large noises: means of selected performance measures over $T = 500$ iterations for continuous knapsack instances; the shaded region is 95% confidence interval for the means.

the effects of large noises on performances are more noticeable, and it is worth noting that the OL algorithm with first order oracle outperforms the one with solution oracle under large noises.

We next discuss results in the CES setup, i.e., when the agent's problem has the form (5.11) with the equality constrained knapsack domain, i.e., $X(u_t) = X^{eck}(u_t)$, under small noises in Figure 5.8, and under large noises in Figure 5.9. Similar to the instance of learning a quadratic utility function, both ℓ^{sim}-based OL algorithms lead to converging average regrets with respect to ℓ^{sim} and decreasing average loss gaps, which are consistent with our above theoretical analysis. We note that the magnitudes of noises appears to have much smaller effects on the learning performance, as both figures show similar output.

We study the online preference learning task, where a learner wishes to learn a non-strategic agent's private utility function through observing the agent's utility-maximizing actions in a changing environment. We adopt an online inverse optimization setup, where the learner observes a stream of agent's actions in an online fashion and the learning performance is measured by regret associated with a loss function. Due to the inverse optimization component, attaining or proving convexity is difficult for all of the usual loss functions in the literature. We address this challenge by designing ℓ^{sim}, a loss function that is convex under relatively mild assumptions. We establish that the regret with respect to ℓ^{sim} also bounds the regret with respect to the three classical loss functions $\ell^{sub}, \ell^{est}, \ell^{pre}$, commonly used in the inverse optimization literature. This allows us to design a flexible OL framework that

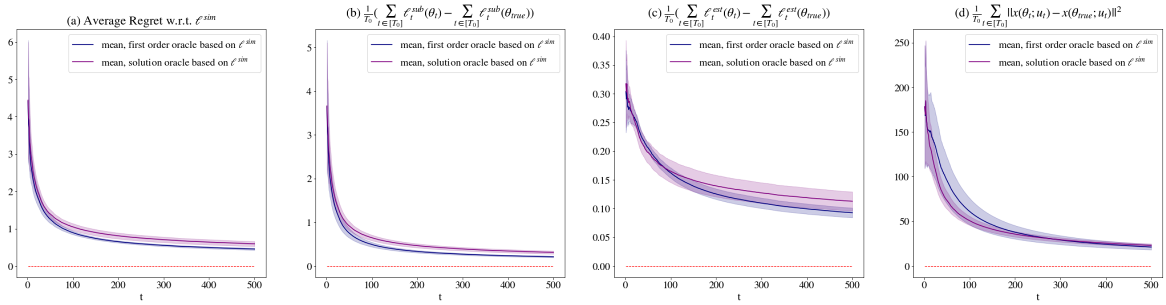


Fig. 5.8 Learning a CES utility function under small noises: means of selected performance measures over $T = 500$ iterations for equality constrained knapsack instances with $n = 50$; the shaded region is 95% confidence interval for the means.

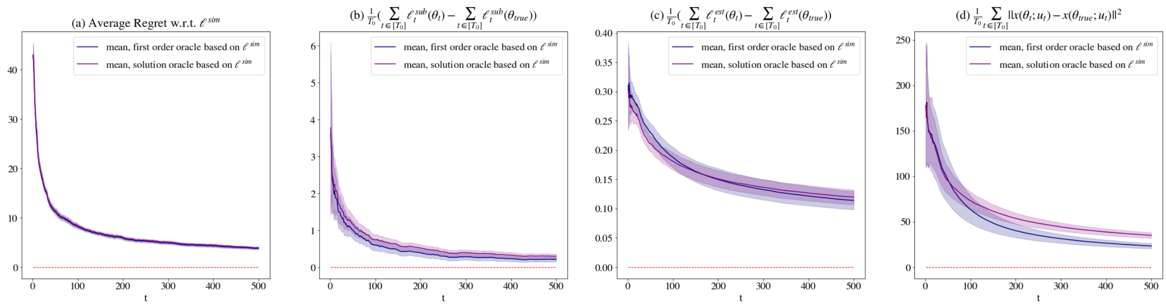


Fig. 5.9 Learning a CES utility function under large noises: means of selected performance measures over $T = 500$ iterations for equality constrained knapsack instances with $n = 50$; the shaded region is 95% confidence interval for the means.

enables a uni ed treatment of loss functions without restrictive technical assumptions and supports a variety of online convex optimization algorithms. We demonstrate with theoretical and empirical evidence that our framework based on the new loss function (coupled in particular with the online Mirror Descent) has signi cant advantages in terms of regret performance and solution time over other OL algorithms from the literature.

Chapter 6

Modeling and Inferring Dynamic Ethical Judgments Around High-stakes Allocations

6.1 Introduction

AI and ML tools are permeating various facets of life in modern society. These powerful technologies are applied in numerous policy domains to inform or make consequential decisions impacting people's lives. In particular, they increasingly inform or automate high-stakes *allocation* decisions in domains such as lending, employment, education, and healthcare. The past decade has witnessed an overwhelming body of evidence establishing the need for AI and ML to reflect social values and moral ideals, such as justice and fairness. However, translating these principles into computationally tractable and verifiable forms has proven to be challenging. As an example, consider outcome fairness through ML-based decisions. Existing mathematical formulations of fairness for ML (e.g., statistical parity) have primarily treated it as a *static*, one-size-fits-all concept–captured in terms of ad hoc predictive parity conditions across socially salient groups. The algorithmic interventions designed to guarantee these notions are similarly *myopic* and *one-shot*. These mathematical treatments fail to account for the *context-dependent* nature of moral ideals and the incremental, local remedies often needed to ensure justice.

A growing body of work has called on the AI-ethics community to bring stakeholders' judgments into the process of formulating moral values and principles for AI. Following this human-centric view toward ethics, we first observe that people's[1] moral judgments are seldom invariable across situations and contexts. In particular, they are often informed by numerous consideration, including but not limited to:

[1] Unless otherwise stated, we use the term `people' to refer to *stakeholders*, not the *public*.

- Who stands to receive harm/benefit as a result of the allocation and what is the extent of the harm/benefit?
- What is the policy's historical backdrop (e.g., does the decision impose additional burdens on historically under-served individuals and groups?)
- What are the future implications of various policy alternatives (e.g., will incurring an extra burden today improve the society's overall welfare and prosperity tomorrow?)

Even absent of disagreement regarding the true answer to the above questions, how they inform allocations could be a genuine point of normative disagreement. So an ethically-minded social planner in charge of the allocation will need to understand and aggregate stakeholders' moral judgments. Additionally, allocational policies are often sequential in nature. Through the decision-making policies the planner enacts today, he/she can impact and shift the above determinants of moral judgments, giving rise to a different state tomorrow. Understanding such dynamics allows the planner to design effective interventions that stir society toward a desirable social state over time.

We consider a setting in which a social planner or policymaker has to make a sequence of decisions regarding the allocation of scarce resources in high-stakes social domains. Our goal is to understand stakeholders' moral judgments regarding such allocational policies In particular, we aim to evaluate the sensitivity of these judgments to the context/history of allocations and their perceived future impact on various socially salient groups. As a concrete example, consider a central agency in charge of allocating scarce medical resources (e.g. vaccines, hospital beds) to patients during a viral epidemic. The allocation decisions shape the overall environment by affecting the welfare of patients, and the urgency and the demand for the particular resources. A patient is associated with many attributes which describe his/her demographic backgrounds, medical characteristics, lifestyle, etc. Depending on the social context, particularly how severe the health issue is and how scarce the needed healthcare resource is, people may have different opinions about which of these attributes should be considered in the policy making process. As we have observed during the current COVID-19 pandemic, age and underlying medical conditions are critical factors to consider to prioritize more vulnerable people. We can imagine that as vaccines roll out and treatment improves, people's emphasis on these factors may reduce; this is an example of external factors changing the social context, which further influences people's moral judgments.

6.1.1 Our Approach and Results

We propose a mathematical model to capture and infer stakeholders' potentially-dynamic moral preferences. Our model utilizes a Markov Decision Process (MDP) to represent the sequential decision making context, more specifically, the sequential allocation of scarce resources. We suppose the reward function of the MDP reflects a stakeholder's moral preferences on the resource distribution decisions, then an allocation policy viewed as more ethically/morally acceptable will have a higher reward. Based on this moral preference model, we can infer a stakeholder's ethical/moral judgments

Moral principles from bioethics. There is a robust history of debate in bioethics about normative principles for the distribution of scarce medical resources across a variety of contexts Cohen et al.[2009], Bayer et al.[2011], Wertheimer and Emanuel[2006]. These contexts can range from non-emergency contexts, such as organ donation and hospital triage, to emergency contexts, such as natural disasters and pandemics (where the health of an entire population is impacted). One solution to this problem is to simply distribute resources randomly by either a lottery or other randomization policy Peterson[2008]. There is an Egalitarian justification for these principles, and they are commonly implemented in non-emergency contexts under "first come, first served." However, in emergency contexts, the policies of hospitals and governments will almost always favor certain groups over others. As Savulescu et al [2020] declare: "there are no egalitarians in a pandemic." We are interested in the normative principles which are used to justify these types of emergency policies.

We group these normative principles for fair distribution into three broad categories:

- Prioritarian:

  This approach favors the most vulnerable members of a population, such as those who are sickest, youngest, or oldest, regardless of how and why one is vulnerable. The concern is only about those who are most likely to be impacted severely by a lack of resources (Low probability of survival, given no treatment)

- Consequentialist:

  This approach attempts to maximize the overall benefits and minimize overall losses, which might mean favoring those who have "most to lose" (high expected value, given treatment), with these impacts existing at a local (family) or global (society) level.

- Desert:

  This approach favors those who are owed compensation or reward because of their eligible features, usually some form of qualifying past behavior like lifestyle choices and effort.

The first two categories will often converge on allocating resources to the most vulnerable, so long as the consequences of patient deaths are roughly comparable, and the probability of survival without treatment is usually inversely proportional to the probability of survival with treatment. However, these assumptions can and do come apart. For instance, in an influential article in the New England Journal of Medicine from early in the COVID-19 pandemic, Emmanuel et al. [2020] provide Consequentialist recommendations which go against a purely Prioritarian focus on vulnerability:

> Operationalizing the value of maximizing benefits means that people who are sick but could recover if treated are given priority over those who are unlikely to recover even if treated and those who are likely to recover without treatment. Because young, severely ill patients will often comprise many of those who are sick but could recover with treatment, this operationalization also has the effect of giving priority to those who are worst off in the sense of being at risk of dying young and not having a full life . . .

are relevant does change over time. In the famous case-study of Memorial Medical Center in New Orleans, when not all patients could be evacuated from the storm-ravaged hospital, the staff changed their normative principles over the course of several days from a Prioritarian to a Consequentialist approach. It is crucial to understand the ways in which not only static but also dynamic features of a situation play a role in the development of policies for resource allocation.

6.2 Problem formulation

We consider a Markov Decision Process (MDP) model representing the sequential allocation of scarce resources. For simplicity, we work with a single type of resource, such as, hospital beds during an epidemic. We suppose the allocation policy proceeds in phases: in each phase, the policy prioritizes people with certain features, such as specific age ranges, medical histories, and occupations. A stakeholder's moral preference is reflected via her/his opinions on who should be prioritized next. As the allocation unfolds over time, stakeholders' moral preferences may shift in line with the evolving societal context.

6.2.1 MDP Model

A standard MDP is defined with a tuple $M = \langle S, A, P, R \rangle$, where S is the set of states, A is the set of actions, $P : S \times A \times S \to [0,1]$ is the transition probability function, with $P(s'|s,a)$ denoting the probability of transitioning to state s' from state s by taking action a, $R : S \to \mathbb{R}$ is a state-based reward function.

A state $s_t \in S$ in our running example describes the current state of affairs—that is, the current profile of immunity and vulnerability to the virus across various groups. Groups are defined as follows:. We consider n ethically relevant features, and uses these features to assign people to groups. We define s_t with the group features $s_t = (s_{t,1}, \ldots, s_{t,n})$, where $s_{t,i}$ is a vector representing the profile of group i at step t. Note that we do not require mutually exclusive features, namely, a person may have multiple features, thus belonging to multiple groups simultaneously. In our running example, $s_{t,i} = (x_i^t, v_i^t, d_i^t)$, which respectively are the proportion of group i that, at step t, have received the resource and thus immune to the virus, are susceptible to infecting the virus, and have passed away from contracting the virus. As we are not aiming to use the most efficient state representations, it may be possible to reduce the size of S with alternative state definitions.

An action $a_t \in A$ represents the current time step's resource allocation decision, that is, how resources will be distributed across the n groups at time t. Similar to the state notation, we can define a_t with the group based action features: $a_t = (a_1^t, \ldots, a_n^t)$, where a_i^t is the proportion of the current step's available resources allocated to group i. For the special case when the policy assigns all available resources to group i, we will have $a_i^t = 1$ and $a_j^t = 0$ for all $j \neq i$.

Next, to define the transition probabilities, we suppose the MDP model is deterministic, namely $P(s'|s,a) \in \{0,1\}$ for all s', s and a. The deterministic assumption fits the resource allocation setting

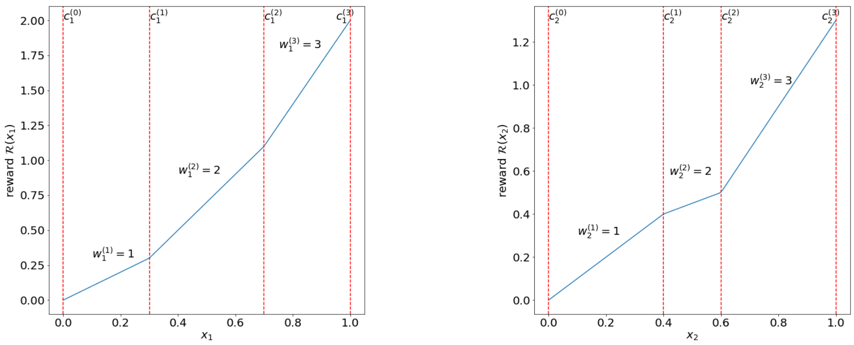


Fig. 6.2 Piecewise linear rewards for group 1 (left) and group 2 (right)

importance ranking among groups remain unchanged throughout state shifts. Therefore, the simple linear reward is insufficient for modeling changing priorities across groups over time.

A more flexible alternative is to use a spline reward function (i.e. a piece-wise polynomial function) to model changes in priorities as the result of changes to the underlying state of the world. Suppose a spline reward function R consists of m pieces $R_1, \dots, R_m$. Each piece represents a different type of moral preference in its corresponding domain. The domain boundary points therefore indicate shifts in preferences. The complete feature space considered in defining rewards is split into these disjoint domains. In our running example, we define a piece-wise linear reward function R with respect to the resource allocation features x_i. For each group i, the interval $[l_i, u_i]$ for possible x_i values is covered by m ordered, disjoint subintervals $[l_i = c_i^{(0)}, c_i^{(1)}], [c_i^{(1)}, c_i^{(2)}], \dots, [c_i^{(m-1)}, u_i = c_i^{(m)}]$. In addition, we have m slope vectors $w^{(1)}, \dots, w^{(m)}$ characterizing all pieces. For $k = 1, \dots, m$, when $(x_1, \dots, x_n) \in [c_1^{(k-1)}, c_1^{(k)}] \times \dots \times [c_n^{(k-1)}, c_n^{(k)}]$, the reward definition follows the k-th piece and prioritizes the groups with weights $w^{(k)}$. We next show an example of a 3-piece reward function with two groups: we illustrate the reward of each group separately for clarity.

Based on this general spline reward, we focus on a simple two-piece case where the first piece is linear and the second piece is constant. We select this particular reward definition to reduce the needed number of parameters.

$$R(s, w, c) := R(x_1, \dots, x_n, (w_1, \dots, w_n), (c_1, \dots, c_n)) := \sum_{i=1}^{n} w_i \min\{x_i, c_i\} \tag{6.1}$$

In this reward function, we skip the piece index in $w^{(k)}, c^{(k)}$ to simplify notations. We can interpret the reward as follows. Throughout the allocation process, group i's feature x_i increases as the group receives more resources. In the beginning, allocations to group i are rewarded linearly with the weight of w_i. After x_i increases to exceed c_i, the cumulative reward will stay fixed at $w_i c_i$, namely, further allocations to group i gain 0 additional rewards. We can interpret c_i as the resource level that is considered sufficiently high for group i so that additional resources given to the group will not be rewarded. Using the two-piece reward function, we can model changes in the importance ranking among groups.

Based on the MDP model, we use a trajectory $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$ to denote a sequence of allocation decisions and the resulting state changes. The trajectory reward is a discounted sum of the immediate reward gained at each state of the trajectory, namely, $R(\tau|w,c) = \sum_{t=1}^{T} \gamma^t r(s_{t+1}, s_t; w, c) = \sum_{t=1}^{T} \gamma^t (R(s_{t+1}; w, c) - R(s_t; w, c))$ with $\gamma > 0$ as the discount factor. $R(\tau; w, c)$ represents a stakeholder's perceived gain from shifting the societal context from s_1 to s_{T+1} through a sequence of allocation policies $a_1, \ldots, a_T$ overtime.

6.2.2 Fairness Preference Model

We take the perspective of a policy planner who wishes to infer a stakeholder's preference by learning her/his reward function. We follow the framework of preference-based reward learning studied in literature, e.g. B y k et al. (2019); Sadigh et al. (2017). The reward learning involves an interactive process where the planner asks a stakeholder to answer a sequence of queries, and uses the query answers to iteratively update the estimates of w and c. Each query is a comparison between two trajectories both starting from the same initial state and of equal length/number of phases. Again, example from the tutorial could help. We next show a sample query in the setting of our running example.

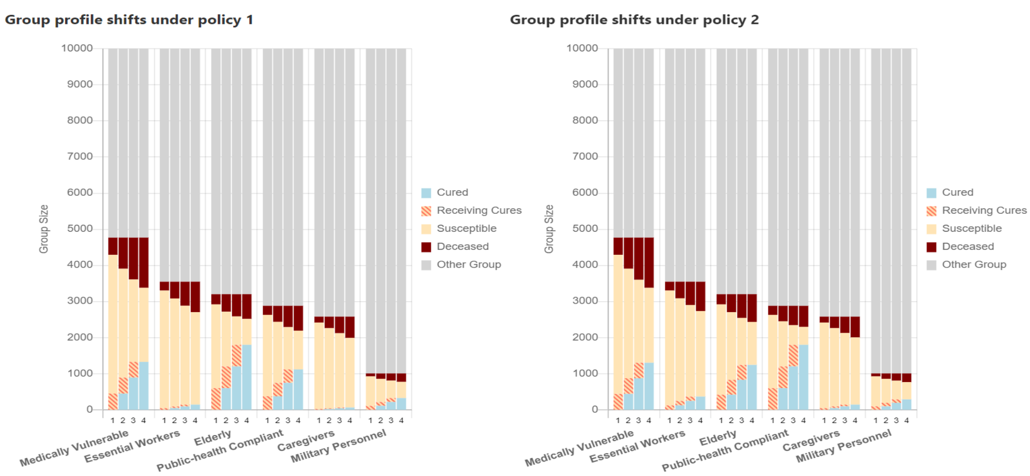


Fig. 6.3 A query of trajectories of three future phases starting from the same initial state

We use the Bradley-Terry model, a standard human choice model Luce (2012), to represent a stakeholder's moral preference model. Suppose a query asks to compare trajectories τ_1 and τ_2. If a stakeholder prefers τ_1 to τ_2, we denote it with $\tau_1 \succ \tau_2$. A stakeholder with reward function $R(s; w, c)$ will choose τ_1 as more preferable, namely, more ethically acceptable with probability:

$$P(\tau_1 \succ \tau_2 | w, c) = \frac{\exp R(\tau_1; w, c)}{\exp R(\tau_1; w, c) + \exp R(\tau_2; w, c)}. \tag{6.2}$$

approximation.

$$
\begin{aligned}
J_t &= \operatorname{argmax}_{Q \in \mathcal{Q}} H(W;C) - \mathbb{E}_{u \sim U_{w,c}(Q)} H(W;C|u) \\
&= \operatorname{argmax}_{Q \in \mathcal{Q}} -\mathbb{E}_{(w,c) \sim P(W;C)} \log P(w,c) + \mathbb{E}_{(w,c) \sim P(W;C);\, u \sim U_{w,c}(Q)} \log P(w,c|u) \\
&= \operatorname{argmax}_{Q \in \mathcal{Q}} \mathbb{E}_{(w,c) \sim P(W;C);\, u \sim U_{w,c}(Q)} \left[\log P(w,c|u;Q) - \log P(w,c)\right] \\
&= \operatorname{argmax}_{Q \in \mathcal{Q}} \mathbb{E}_{(w,c) \sim P(W;C);\, u \sim U_{w,c}(Q)} \left[\log P(u;Q|w,c) - \log P(u)\right] \\
&= \operatorname{argmax}_{Q \in \mathcal{Q}} \mathbb{E}_{(w,c) \sim P(W;C);\, u \sim U_{w,c}(Q)} \left[\log P(u;Q|w,c) - \log\left(\frac{1}{M} \sum_{(w_i,c_i) \in W} P(u;Q|w_i,c_i)\right)\right] \\
&= \operatorname{argmax}_{Q \in \mathcal{Q}} \frac{1}{M} \sum_{(w_j,c_j) \in W} \mathbb{E}_{u \sim U_{w,c}(Q)} \left[P(u;Q|w_j,c_j) \log\left(M \frac{P(u;Q|w_j,c_j)}{\sum_{(w_i,c_i) \in W} P(u;Q|w_i,c_i)}\right)\right].
\end{aligned}
\tag{6.4}
$$

6.4 Experimental Design

We carry out human-subject experiments on Amazon Mechanical Turk to understand people's moral assessments of allocation policies in the hypothetical scenario. We apply the active learning framework introduced in Section 6.3 to elicit our participants' moral preferences regarding the sequential allocation of scarce medical resources in a hypothetical epidemic.

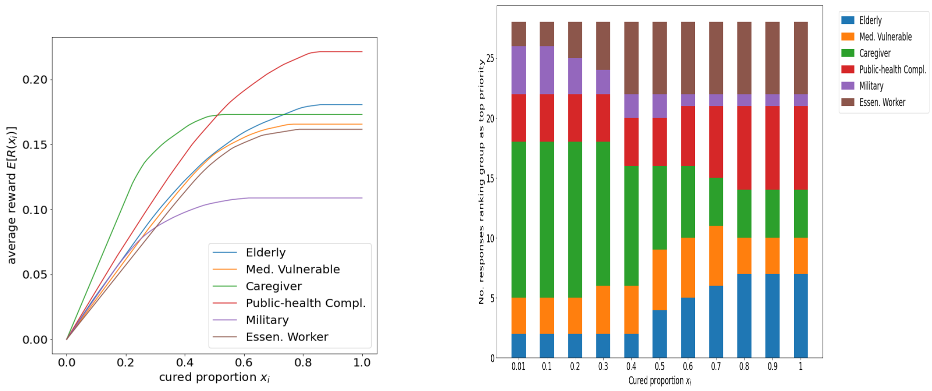
6.4.1 Scenario and Setup

The participant first receives basic information about the study, including study background and goals, the hypothetical decision-making context, and the type of questions they need to respond to if they decide to participate. Then the participant answers a questionnaire consisting of 20 queries. Each question/query consists of comparison between two trajectories each representing a different allocation policy.

The following information is presented to participants who wish to participate in our study.

> **Background and Task Description**
> The goal of this survey is to understand your moral judgments regarding the sequential allocation of scarce medical resources.

> **Hypothetical Scenario**
> Imagine a viral epidemic that has infected millions of people around the world leading to a disease with a very high mortality rate. There is currently only a single highly effective cure for the disease—those who receive the cure will fully recover (if currently infected) and become immune to the virus in the future. Unfortunately, the number of cure doses that can be produced and administered every month is limited, so public health officials need to decide which groups should be prioritized at any given time. In the questionnaire that follows, we will present you with additional information about several possible states of the epidemic and ask you to choose your preferred allocational policy between two cure allocation policies.



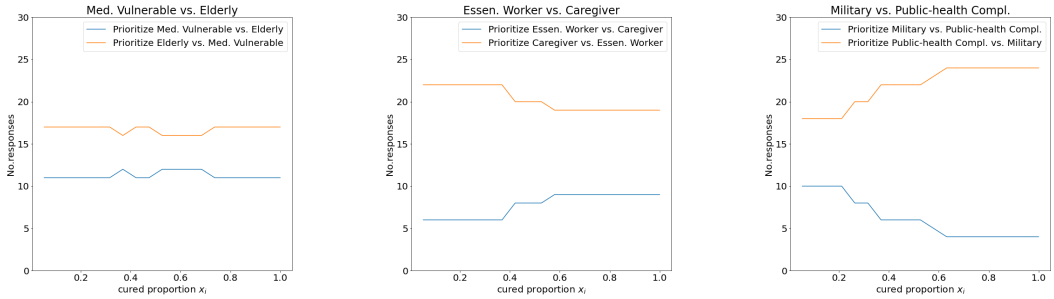
(a) Means of group rewards (b) Response distribution of top priority group

Fig. 6.5 Rewards based on estimates from survey responses

viewed as suf ciently cured. The medically vulnerable and essential workers have similar reward trends as the elderly. Lastly, the military personnel group has slightly higher rewards than the medically vulnerable and essential workers in the beginning of cure allocation, but drops to the least prioritized for majority of the allocation process.

We also analyze the aggregate preferences on a more re ned level. In Fig. 6.5b, we focus on the top priority group throughout the cure allocation process. Each stacked bar shows the distribution of responses prioritizing each group when all groups have the same given cured proportion. We observe the same preference dynamics as in Fig. xx. Caregivers are the most common top priority at lower cured levels. As more people are cured, it becomes increasingly ethical to prioritize essential workers, public-health compliant and elderly groups.

To explore further, we compare groups pairwise to understand the ethical position of each group against another group at different cured levels. We highlight observations from three pairs: elderly vs. medically vulnerable, caregivers vs. essential workers, public health compliant vs. military personnel. As we discuss in Section ??, prioritizing either group in these groups respectively re ect prioritarianism, restorative justice and distributive justice. Comparing each pair could provide further insights on how these ethical values play out in people's ethical judgments. Between the elderly and the medically vulnerable, the respondents are roughly equally split in preferring either group and the prioritization trend is stable at all cured levels. This indicates that when stakeholders care about favoring the worse off in allocation, they may be indifferent to the different reasons leading people to the disadvantaged positions. Between caregivers and essential workers, more respondents prioritize the former throughout the cure allocation process, but the gaps are small. After the cured proportion reaches 40% in both groups, essential workers receive slightly stronger prioritization. Lastly, the public-health compliant are noticeably prioritized over military personnel at all cured levels, and the gaps grow wider as more people receive cures. These comparisons for the latter two groups indicate



(a) Elderly vs. Medically vulnerable

(b) Caregivers vs. Essential workers

(c) Public-health compliant vs. Military personnel

Fig. 6.6 Number of responses prioritizing each group in a pair

that the distinctions among what contribute to people's instrumental values and desertness could impact a stakeholder's moral judgments.

Discussion.In a number of surveys, respondents mention at least once that they prefer a policy because it bene ts multiple groups at once. Recall that we illustrate the impacts of each three-phase allocation policy with a stacked bar plot, and each phase the proportion of people receiving cures is highlighted in all the groups. Due to the overlaps among groups, as we explain previously, the cure allocation to one group effectively affects all groups through the overlaps. In our experiment instance, the six groups have different overlapping pro les. For example, the public-health compliant group has a larger overlap with the medically vulnerable group than the essential worker group. Suppose a respondent views curing the medically vulnerable as the most ethical action, he/she is likely to nd curing the public-health compliant in a phase more ethically acceptable than curing the essential worker, as the former would lead to a more prominent cured effect for the medically vulnerable.

Our current experiment design does not prompt people to state their thresholds explicitly. When asked to explain their choices, respondents could easily indicate their weight preferences for the groups by indicating which groups they believe should be prioritized. On the contrary, it is more dif cult for respondents to specify the exact cure thresholds that would make a group suf ciently protected so that further prioritization is unnecessary. Nevertheless, we receive answers mentioning that a policy is or is not preferred due to a certain group already has a relatively high cured level, which demonstrate that respondents use thresholds implicitly when formulating ethical judgments.

6.6 Conclusion

Chen, V. and Hooker, J. N. (2020a). Balancing fairness and efficiency in an optimization model. *arXiv preprint*, 2006.05963.

Chen, V. and Hooker, J. N. (2020b). A just approach balancing Rawlsian leximax fairness and utilitarianism. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pages 221–227.

Chiarandini, M., Fagerberg, R., and Gualandi, S. (2019). Handling preferences in student-project allocation. *Annals of Operations Research*, 275(1):39–78.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163.

Chouldechova, A., Benavides-Prado, D., Fialko, O., and Vaithianathan, R. (2018). A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*, pages 134–148. PMLR.

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

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*.

Cohen, W. W., Schapire, R. E., and Singer, Y. (2009). Principles for allocation of scarce medical interventions. *Lancet*, 373:423–431.

Corbett-Davies, S. and Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

Cowell, F. A. (2000). Measurement of inequality. In Atkinson, A. B. and Bourguignon, F., editors, *Handbook of Income Distribution*, volume 1, pages 89–166. Elsevier.

Dalton, H. (1920). The measurement of the inequality of incomes. *Economic Journal*, 30:348–461.

Deepak, P. and Abraham, S. S. (2020). Representativity fairness in clustering. *WebSci*, pages 202–211.

Dieterich, W., Mendoza, C., and Brennan, T. (2016). COMPAS risk scales: Demonstrating accuracy equity and predictive parity. Technical report, Northpointe Inc. Research Department. 8 July.

Dong, C., Chen, Y., and Zeng, B. (2018a). Generalized inverse optimization through online learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, NeurIPS '18, pages 86–95.

Dong, C. and Zeng, B. (2020). Expert learning through generalized inverse multiobjective optimization: Models, insights, and algorithms. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2648–2657. PMLR.

Dong, J., Roth, A., Schutzman, Z., Waggoner, B., and Wu, Z. S. (2018b). Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC '18)*, pages 55–70. ACM.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J., and Pontil, M. (2018a). Empirical risk minimization under fairness constraints. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 2796–2806, Red Hook, NY, USA. Curran Associates Inc.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J. S., and Pontil, M. (2018b). Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pages 2791–2801.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012a). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. (2012b). Fairness through awareness. In *Symposium on Innovations in Theoretical Computer Science (ITCS)*, pages 214–226.

Eisenhandler, O. and Tzur, M. (2019). The humanitarian pickup and distribution problem. *Operations Research*, 67:10–32.

Emanuel, E., Persad, G., Upshur, R., Thome, B., Parker, M., Glickman, A., Zhang, C., Boyle, C., Smith, M., and Phillips, J. (2020). Fair allocation of scarce medical resources in the time of covid-19. *New England Journal of Medicine*, 382:2049–2055.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874.

Fan, R.-E., Chen, P.-H., and Lin, C.-J. (2005). Working set selection using second order information for training support vector machines. *Journal of machine learning research*, 6(Dec):1889–1918.

Frankfurt, H. G. (2015). *On Inequality*. Princeton University Press.

Freeman, S., editor (2003). *The Cambridge Companion to Rawls*. Cambridge University Press.

Gabriel, I. (2020). Artificial intelligence, values, and alignment. *Minds and Machines*, 30:411–437.

Gauthier, D. (1983). *Morals by Agreement*. Oxford University Press.

Gini, C. (1912). *Variabilità e mutabilità*. P. Cuppini, reprinted 1955 in E. Pizetti abd T. Salvemini, eds., *Memorie di metodologica statistica*, Rome: Libreria Eredi Virgilio Veschi.

Glasmachers, T., Igel, C., Bennett, K. P., and Parrado-Hernández, E. (2006). Maximum-gain working set selection for svms. *Journal of Machine Learning Research*, 7(7).

Hardt, M., Price, E., and Srebro, N. (2016a). Equality of opportunity in supervised learning. In *Proceedings of 30th International Conference on Neural Information Processing*, pages 3323–3331.

Hardt, M., Price, E., and Srebro, N. (2016b). Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

Harsanyi, J. C. (1977). *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*. Cambridge University Press.

Hazan, E. (2019). Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*.

Heidari, H., Ferrari, C., Gummadi, K., and Krause, A. (2018). Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems*, pages 1265–1276.

Heidari, H., Loi, M., Gummadi, K. P., and Krause, A. (2019). A moral framework for understanding fair ml through economic models of equality of opportunity. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 181–190.

Appendix A

Appendix to Chapter 3

A.1 Pseudocode

A.1.1 Fair Sequential Minimal Optimization for Fair General SVM

Algorithm 1 Fair SMO

Input: Training dataset $D = \{x_i, y_i, z_i\}_{i \in [n]}$, fairness constraints parameters $\{p_l, c_l, e_l\}_{l \in [k]}$, kernel function K, stopping criteria tolerance t.

Output: m, g satisfying KKT conditions (3.26) up to tolerance t

```
Initialize m^(0), g^(0) to all 0
Initialize gradients G, with G_{m_i} = ∇_{m_i} g(m, g) = −1 for all i ∈ [n] and G_{g_l} = ∇_{g_l} g(m, g) = −(e_l − c_l) for all l ∈ [k].
for t = 0, 1, 2, ... do
    if m(m^(t), g^(t)) ≤ M(m^(t), g^(t)) + t and |g_l^(t) G_{g_l}| ≤ t for all l ∈ [k] then
        Return m^(t), g^(t) and terminate.
    else
        Choose working set (i, j) = SelectB(m^(t), g^(t), I, G).
        if j ≠ −1 then
            Update working set variables (m_i^(t+1), m_j^(t+1)) = Update_m(m^(t), {i, j}, G).
            Update G using (3.33).
        end if
        Update g variables and obtain corresponding gradients g^(t+1), G = Update_g(g^(t), G).
    end if
end for
```

Appendix B

Appendix to Chapter 5

B.1 Convexity Status of ℓ^{pre} and ℓ^{est}

In this appendix, we examine the convexity status ℓ^{pre} and ℓ^{est} under our Assumption 2. Recall that we already establish in Lemma 3 that under Assumption 2, $\ell^{sub}(\theta)$ is convex in θ. On the other hand, ℓ^{pre} and ℓ^{est} are not guaranteed to be convex in θ even under Assumption 2 and even when agent's problem is a one dimensional optimization problem.

Example 1. Suppose $\Theta = [-1, 1]$ and $\theta_{true} = 1$, and an agent's forward problem is $\min_x\{\theta x : x \in X\}$. We consider a convex domain $X = [-1, 1]$. Let $x(\theta) := \arg\min_x\{\theta x : x \in X\}$, i.e., the set of optimizers of the agent's problem for given θ. Then, we easily deduce that the agent's optimal action(s) at a given θ are: $x(\theta) = -1$ if $\theta > 0$, $x(\theta) \in X$ if $\theta = 0$, and $x(\theta) = 1$ if $\theta < 0$. Specifically, this implies $x(\theta_{true}) = -1$.

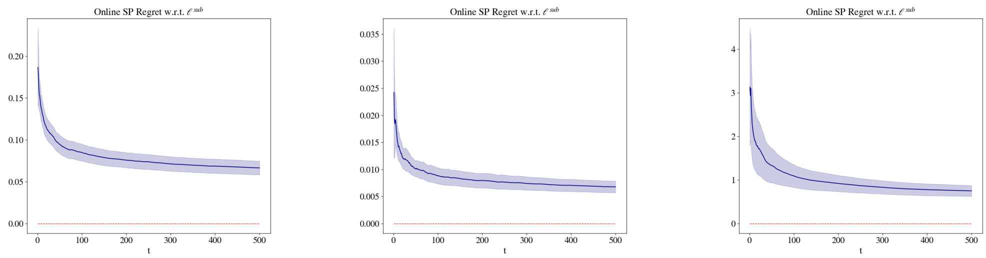
Consider $\theta_1 = 1$, $\theta_2 = -1$ and $\lambda = \frac{1}{4}$, then $\tilde{\theta} = \lambda\theta_1 + (1-\lambda)\theta_2 = -\frac{1}{2}$. By the format of $x(\theta)$ in this problem and the definition of ℓ^{est}, we observe that

$$\ell^{est}(\theta_1) = \theta_{true}(x(1) - x(\theta_{true})) = 0;$$
$$\ell^{est}(\theta_2) = \theta_{true}(x(-1) - x(\theta_{true})) = 2;$$
$$\ell^{est}(\tilde{\theta}) = \theta_{true}(x(-1/2) - x(\theta_{true})) = 2.$$

Therefore, we deduce $\ell^{est}(\tilde{\theta}) > \lambda\ell^{est}(\theta_1) + (1-\lambda)\ell^{est}(\theta_2)$ that shows that ℓ^{est} is not a convex function of θ. Similarly, in the case of ℓ^{pre}, we arrive at

$$\ell^{pre}(\theta_1) = (x(1) - x(\theta_{true}))^2 = 0;$$
$$\ell^{pre}(\theta_2) = (x(-1) - x(\theta_{true}))^2 = 4;$$
$$\ell^{pre}(\tilde{\theta}) = (x(-1/2) - x(\theta_{true}))^2 = 4.$$

Similarly, we arrive at $\ell^{pre}(\tilde{\theta}) > \lambda\ell^{pre}(\theta_1) + (1-\lambda)\ell^{pre}(\theta_2)$ and hence conclude ℓ^{pre} is not convex.



(a) quadratic utility function, continuous knapsack

(b) quadratic utility function, continuous polytope

(c) CES utility function, equality constrained knapsack

Fig. B.1 Means of average SP regrets over $T = 500$ iterations; the shaded region is 95% confidence interval for the means.

average regret with respect to ℓ^{sub} follows a similar convergence trend as the average SP regret. Even though the online SP algorithm appears to be sufficient to guarantee the convergence of the average regret with respect to ℓ^{sub} in this case, we observe that there is still a lack of convergence guarantees for the other regrets, in particular, R_T^{sim} and R_T^{pre} are increasing throughout all iterations.

B.2.2 OCO with Solution Oracle

To apply the implicit OL algorithm with respect to ℓ^{sub}, we need a solution oracle for solving $\min_{\theta \in \Theta} V_\theta(\theta_t) + \eta_t \ell_t^{sub}(\theta)$. Even though $\ell_t^{sub}(\theta)$ is convex in θ, the required solution oracle may be computationally intractable, because the format of $\ell_t^{sub}(\theta)$ makes the underlying optimization model a bilevel program with both θ and x as the decision variables. Recall from Appendix B.3, the solution oracle for ℓ^{sim}-based implicit OL algorithm is naturally a convex program in θ with the convenient structure of ℓ^{sim}. The ℓ^{pre}-based solution oracles are bilevel programs, which may not be tractable in general, as we have discussed for the case of learning a CES utility function.

Similar computational difficulties occur for using ℓ^{sub}-based solution oracles. We again suppose that the squared Euclidean norm is used as the distance generating function in the solution oracle, then the ℓ^{sub}-based implicit OL algorithm updates θ_{t+1} by solving the following bilevel program:

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \left(\langle \theta, c(y_t) - c(x(\theta; u_t)) \rangle - f_1(x(\theta; u_t); u_t) \right)$$

where

$$x(\theta; u_t) \in \arg\min_x \{ f(x; \theta; u_t) : g(x; u_t) \le 0, \; x \in X \}.$$

To obtain a single-level reformulation, we can again apply the KKT conditions to the inner problem to represent $x(\theta; u_t)$ in the constraints. We demonstrate the following reformulations for our experiment instances to further illustrate where the computational challenges rise. First, when the agent's problem has the form (5.10) with a continuous polytope domain $X(u_t) = X^{cp}(A_t; c_t)$, the ℓ^{sub} based solution

oracle simplifies to the following single level model.

$$
\begin{aligned}
\theta_{t+1} = \mathop{\arg\min}_{\theta;x;w;v;y;z} \quad & \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t\left(\langle\theta; x - y\rangle - \frac{1}{2}x^T P x\right)\\
\text{s.t.} \quad & A_t x \geq c_t;\ x \in \mathbb{R}^n_+\\
& w_i \leq M y_i;\ i \in [n]\\
& x_i \leq M(1 - y_i);\ i \in [n]\\
& v_j \leq M z_j;\ j \in [m]\\
& (A_t)_j^{\top} x - (c_t)_j \leq M(1 - z_j);\ j \in [m]\\
& Px - \theta + A_t^{\top} v - w = 0\\
& v \in \mathbb{R}^m_+; w \in \mathbb{R}^n_+;\ y \in \{0,1\}^n;\ z \in \{0,1\}^m\\
& \theta \in \Theta.
\end{aligned}
$$

For the other case, when the agent's problem has the form (5.11) with an equally constrained knapsack domain, $X(u_t) = X^{eck}(p_t; b_t)$, the ℓ^{sub} based solution oracle is equivalent to:

$$
\begin{aligned}
\theta_{t+1} = \mathop{\arg\min}_{\theta;x;w;v;y} \quad & \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t\left(\langle\theta; x - y\rangle - \frac{1}{2}x^T P x\right)\\
\text{s.t.} \quad & p_t x = b_t;\ x \in \mathbb{R}^n_+\\
& w_i \leq M y_i;\ i \in [n]\\
& x_i \leq M(1 - y_i);\ i \in [n]\\
& 2\theta_i x_i + v(p_t)_i - w_i = 0;\ i \in [n]\\
& v \in \mathbb{R}; w \in \mathbb{R}^n_+;\ y \in \{0,1\}^n\\
& \theta \in \Theta.
\end{aligned}
$$

We note that the reformulated objective functions contain the bilinear term $\theta_i x_i$, which means the objectives are not guaranteed to be convex in general. As a result, the ℓ^{sub}-based implicit OL algorithm would require an expensive general purpose nonconvex solution oracle.

B.3 Formulations for the Solution Oracles Used in the Implicit OL Algorithms

In this section, we give the solution oracles used in implicit OL algorithms based on ℓ^{sim} and ℓ^{pre} for two forms of agent's utility functions corresponding to the ones used in our numerical experiments. We include ℓ^{pre}-based implicit OL in our discussion for the sake of comparison between ℓ^{sim}-based OL framework and the previous work Dong et al. (2018a). In our computational study, we implemented the following solution oracles when they can be readily solved by standard optimization software.

B.3.1 Solution Oracle for ℓ^{sim}-based Implicit OL Algorithm

Suppose that the squared Euclidean norm is used as the distance generating function in the implicit OL algorithm with the solution oracle. Recall from Definition 1 that ℓ^{sim} has the following form under Assumption 2:

$$\ell^{sim}(\theta; x(\theta_t; u_t); y_t; u_t) := \langle \theta; c(y_t) - c(x(\theta_t; u_t)) \rangle + \langle \theta_{true}; c(x(\theta_t; u_t)) - c(y_t) \rangle :$$

Since the constant term in $\ell^{sim}(\theta)$ has no impact when $\ell^{sim}_t(\theta)$ is used in the objective function of an optimization problem, it can be ignored in the solution oracle formulation. Then, we deduce that the solution oracle for the ℓ^{sim}-based implicit OL algorithm updates θ_{t+1} as

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \langle \theta; c(y_t) - c(x(\theta_t; u_t)) \rangle :$$

In particular, when the agent's problem has the form (5.10) we have $f(x;\theta;u) = \frac{1}{2}x^\top P x - \langle \theta; x \rangle$, i.e., $c(x) = -x$. Thus, in this case, the solution oracle for the ℓ^{sim}-based implicit OL algorithm updates θ_{t+1} as

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \langle \theta; -y_t + x(\theta_t; u_t) \rangle :$$

In the case of CES utility function, i.e., when the agent's problem has the form (5.11), we have $f(x;\theta;u) = -\sum_{i \in [n]} (\theta)_i x_i^2$, and in this case the solution oracle for the ℓ^{sim}-based implicit OL algorithm updates θ_{t+1} as

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \sum_{i \in [n]} \theta_i \left((y_t)_i^2 - x(\theta_t; u_t)_i^2 \right) :$$

B.3.2 Solution Oracle for ℓ^{pre}-based Implicit OL Algorithm

Suppose that the squared Euclidean norm is used as the distance generating function in the implicit OL algorithm with the solution oracle. Then, the solution oracle for ℓ^{pre}-based implicit OL algorithm updates θ_{t+1} by solving the following bilevel program:

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \|y_t - x(\theta; u_t)\|^2;$$

where

$$x(\theta; u_t) \in \arg\min_x \{ f(x;\theta;u_t) : g(x;u_t) \le 0;\ x \in X \}:$$

Recall that when the agent's problem has the form (5.10) with a continuous polytope domain, i.e., $X(u_t) = X^{cp}(A_t; c_t)$, we have

$$x(\theta; u_t) := \arg\max_x \left\{ -\frac{1}{2}x^\top P x + \langle \theta; x \rangle : A_t x \le c_t;\ x \in \mathbb{R}^n_+ \right\};$$

where $P \in \mathbb{S}^n_{++}$ is a fixed positive definite matrix known by both the learner and the agent. Using the KKT optimality conditions for the inner problem, and then introducing binary variables to linearize the resulting nonlinear relations, it is possible to reformulate this bilevel problem into a single level optimization problem with binary variables. In particular, in this case, following these outlined steps, Dong et al. (2018a) proposed the following reformulation of this bilevel problem into a single level MISOCP:

$$
\begin{aligned}
q_{t+1} = \operatorname*{arg\,min}_{q,x,w,v,y,z} \ & \frac{1}{2}\|q - q_t\|^2 + h_t \|y_t - x\|^2 \\
\text{s.t.}\ & A_t x \le c_t;\ x \in \mathbb{R}^n_+ \\
& w_i \le M y_i;\ i \in [n] \\
& x_i \le M(1 - y_i);\ i \in [n] \\
& v_j \le M z_j;\ j \in [m] \\
& (A_t)_j^\top x - (c_t)_j \ge -M(1 - z_j);\ j \in [m] \\
& Px - q + A_t^\top v - w = 0 \\
& v \in \mathbb{R}^m_+;\ w \in \mathbb{R}^n_+;\ y \in \{0,1\}^n;\ z \in \{0,1\}^m \\
& q \in Q.
\end{aligned}
$$

Here, M is the so-called big-M constant. The variables $v \in \mathbb{R}^m_+, w \in \mathbb{R}^n_+$ are the variables corresponding to the Lagrangian multipliers, the binary variables $y_i \in \{0,1\}$ for all $i \in [n]$ are used to linearize the KKT condition $w_i x_i = 0$, and $z_j \in \{0,1\}$ for all $j \in [m]$ are introduced to linearize the KKT relation $v_j((A_t)_j^\top x - (c_t)_j) = 0$. Therefore, the big-$M$ constants must be selected so that they upper bound the components in the bilinear expressions, e.g., x_i and w_i for the complementarity constraint $w_i x_i = 0$ as well as $(A_t)_j^\top x - (c_t)_j$ and v_j for the constraint $v_j((A_t)_j^\top x - (c_t)_j) = 0$. Because in our instances the agent's domain for x is bounded, we can easily obtain bounds on x and $(A_t)_j^\top x - (c_t)_j$ terms. It is also possible to derive an upper bound for the Lagrange multipliers under a Slater condition assumption on the primal problem. Nevertheless, it is well known that using big-M formulations significantly degrade the optimization solver performance, and instead it is encouraged in Gurobi solver that such big-M constraints are encoded as indicator constraints, which is a form of logical constraints supported by Gurobi. In our experiments, we follow this approach and use the indicator constraint feature of the Gurobi solver. Note that this alternative implementation is possible because the big-M constraints essentially represent a complementarity type logical condition.

Note that the continuous knapsack domain $X^{ck}(p_t; b_t)$ is a special case of the continuous polytope domain $X^{cp}(A_t; c_t)$, and thus the same reformulation also holds in that case.

Finally note that when the agent's problem has the form (5.11) with an equally constrained knapsack domain, i.e., $X(u_t) = X^{eck}(p_t; b_t)$, we have

$$x(\theta; u_t) := \arg\min_{x} \left\{ \sum_{i \in [n]} \theta_i x_i^2 : p_t^\top x = b_t,\ x \in \mathbb{R}^n_+ \right\}.$$

In this case, the bilevel program corresponding to the solution oracle in the ℓ^{pre}-based implicit OL algorithm has the following single level reformulation.

$$\begin{aligned}
\theta_{t+1} = \arg\min_{\theta, x, w, v, y}\ & \frac{1}{2}\|\theta - \theta_t\|^2 + \eta_t \|y_t - x\|^2 \\
\text{s.t. } & p_t x = b_t,\ x \in \mathbb{R}^n_+ \\
& w_i \le M y_i,\ i \in [n] \\
& x_i \le M(1 - y_i),\ i \in [n] \\
& 2\theta_i x_i + v(p_t)_i - w_i = 0,\ i \in [n] \\
& v \in \mathbb{R},\ w \in \mathbb{R}^n_+,\ y \in \{0,1\}^n \\
& \theta \in \Theta.
\end{aligned}$$

Unfortunately, this nonconvex mixed integer program contains the bilinear terms $\theta_i x_i$, where both x and θ are continuous variables, in a general constraint, not of a complementarity type constraint. Note that the primal domain is equality constrained continuous knapsack, and thus we can find an upper bound on x variables. Moreover, for $\theta \in \Theta$ and when Θ is bounded like the Euclidean ball or the simplex case that we focus on in this paper, we can find a bound on θ as well. However, because this bilinear term of $\theta_i x_i$ is appearing in a general constraint and not in a complementary constraint, there is no technique to reformulate this nonconvexity as linear constraints by introducing new binary variables. Hence, in this case the ℓ^{pre}-based implicit OL algorithm requires a computationally expensive general purpose nonconvex solution oracle.

B.4 ℓ^{sim}-based Online Inverse Optimization under Imperfect Information

In this appendix, we look into the learning performance in the imperfect information setup under Assumption 2. Recall that for any of the four loss definitions, the loss value $\ell_t(\theta_t)$ can be split into a component measuring the difference between $x(\theta_t; u_t)$ and $x(\theta_{true}; u_t)$, and another one reflecting the noise shifting $x(\theta_{true}; u_t)$ to y_t. For instance, we can write $\ell^{pre}(\theta_t) = \|y_t - x_t + x_t - x_t\|^2 = \|y_t - x_t\|^2 + \|x_t - x_t\|^2 + 2\langle y_t - x_t, x_t - x_t\rangle$. Moreover, this means $\ell_t(\theta_{true}) = \ell(\theta_{true}; x(\theta_{true}; u_t); y_t; u_t)$ can be viewed as an imperfect information loss.

In Section 5.6.4, we consider a different collection of performance measures, consisting of the average regret with respect to ℓ^{sim}, the difference between the average loss incurred from $\{\theta_t, x(\theta_t; u_t)\}$

and the average imperfect information loss from $\{\theta_{true}; x(\theta_{true}; u_t)\}$ with respect to ℓ^{sub} and ℓ^{est}, and the average squared norm distance between $\{x(\theta_t; u_t)\}$ and $\{x(\theta_{true}; u_t)\}$. We next show that the average regret with respect to ℓ^{sim} can be used to bound the other three loss-based measures.

For notation ease, we denote $x_t := x(\theta_t; u_t)$ and $x_t^* := x(\theta_{true}; u_t)$ for all t.

Proposition 6. Suppose Assumption 2 holds and the observations contain noise, namely, in each time step t, $y_t = x(\theta_{true}; u_t) + \epsilon_t$ for some nonzero ϵ_t. For any sequence $\{\theta_t\}_{t\in[T]}$, we have

$$\text{(a)} \quad \sum_{t\in[T]} \ell_t^{sub}(\theta_t) - \sum_{t\in[T]} \ell_t^{sub}(\theta_{true}) \le R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]});$$

$$\text{(b)} \quad \sum_{t\in[T]} \ell_t^{est}(\theta_t) - \sum_{t\in[T]} \ell_t^{est}(\theta_{true}) = \sum_{t\in[T]} f(x_t; \theta_{true}; u_t) - f(x_t^*; \theta_{true}; u_t)$$
$$\le R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]}) + \sum_{t\in T} \langle \theta_{true} - \theta_t; c(y_t) - c(x_t^*) \rangle;$$

Proof. (a) By definition of ℓ^{sub}, we have

$$\begin{aligned}
\sum_{t\in[T]} \ell_t^{sub}(\theta_t) - \sum_{t\in[T]} \ell_t^{sub}(\theta_{true}) &= \sum_{t\in[T]} f(y_t; \theta_t; u_t) - f(x_t; \theta_t; u_t) - f(y_t; \theta_{true}; u_t) + f(x_t^*; \theta_{true}; u_t) \\
&= \sum_{t\in[T]} \langle \theta_t; c(y_t) - c(x_t) \rangle - f_1(x_t) - \langle \theta_{true}; c(y_t) - c(x_t^*) \rangle + f_1(x_t^*) \\
&= \sum_{t\in[T]} \langle \theta_t - \theta_{true}; c(y_t) - c(x_t) \rangle + \sum_{t\in[T]} f(x_t^*; \theta_{true}; u_t) - f(x_t; \theta_{true}; u_t) \\
&\le \sum_{t\in[T]} \ell_t^{sim}(\theta_t) - \min_{\theta\in\Theta} \sum_{t\in[T]} \ell_t^{sim}(\theta) + \min_{\theta\in\Theta} \sum_{t\in[T]} \ell_t^{sim}(\theta) \\
&\le R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]}) + \sum_{t\in[T]} \ell_t^{sim}(\theta_{true}) = R_T(\{\ell_t^{sim}\}_{t\in[T]}; \{\theta_t\}_{t\in[T]});
\end{aligned}$$

where the initial three equations follow from definition and simple algebra, the first inequality follows from the fact that x_t^* minimizes f given $\theta_{true}; u_t$, and the last equation is due to $\ell_t^{sim}(\theta_{true}) = 0$ for all t.

(b) The equation is a direct consequence of the loss definition.

$$\begin{aligned}
\sum_{t\in[T]} \ell_t^{est}(\theta_t) - \sum_{t\in[T]} \ell_t^{est}(\theta_{true}) &= \sum_{t\in[T]} f(x_t; \theta_{true}; u_t) - f(y_t; \theta_{true}; u_t) - f(x_t^*; \theta_{true}; u_t) + f(y_t; \theta_{true}; u_t) \\
&= \sum_{t\in[T]} f(x_t; \theta_{true}; u_t) - f(x_t^*; \theta_{true}; u_t):
\end{aligned}$$

To derive the inequality, we observe that Lemma 5 still holds under imperfect information, hence we can apply Lemma 5 (b) and (c) to rewrite the loss difference.

$$\begin{aligned}
&\sum_{t\in[T]} \ell_t^{est}(\theta_t) - \sum_{t\in[T]} \ell_t^{est}(\theta_{true}) = \sum_{t\in[T]} \ell_t^{sim}(\theta_t) - \sum_{t\in[T]} \ell_t^{sub}(\theta_t) + \sum_{t\in[T]} \ell_t^{sub}(\theta_{true}) \\
&= \sum_{t\in[T]} \ell_t^{sim}(\theta_t) + \sum_{t\in[T]} \left(\langle \theta_{true} - \theta_t, c(y_t) - c(x_t^*)\rangle + f_1(x_t) - f_1(x_t^*) + \langle \theta_t, c(x_t) - c(x_t^*)\rangle\right) \\
&= \sum_{t\in[T]} \ell_t^{sim}(\theta_t) + \sum_{t\in[T]} \langle \theta_{true} - \theta_t, c(y_t) - c(x_t^*)\rangle + \sum_{t\in[T]} f(x_t;\theta_t;u_t) - f(x_t^*;\theta_t;u_t) \\
&\le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) + \sum_{t\in[T]} \langle \theta_{true} - \theta_t, c(y_t) - c(x_t^*)\rangle;
\end{aligned}$$

where the equations follow from definition and algebra, then the inequality uses the fact that x_t minimizes f given $\theta_t; u_t$.

Under imperfect information, (Mohajerin Esfahani et al., 2018, Proposition 2.5) remains valid for a strongly convex f, namely, $\ell_t^{sub}(\theta) \ge \frac{\gamma}{2}\ell_t^{pre}(\theta)$ for all t and for all $\theta \in \Theta$, with γ being the strong convexity parameter of f. Therefore, we can further derive bounds with respect to $\{\ell_t^{pre}\}_{t\in[T]}$.

Corollary 2. Suppose Assumption 2 holds and the observations $\{y_t\}_{t\in[T]}$ contain noises. Suppose further that f is a strongly convex function of x for every θ with a constant $\gamma > 0$, i.e., $f(x;\theta;u) - f(y;\theta;u) \ge \langle s_y, x - y\rangle + \frac{\gamma}{2}\|x - y\|^2$, where s_y is a subgradient of $f(y;\theta;u)$ with respect to y. Then, for any sequence $\{\theta_t\}_{t\in[T]} \subseteq \Theta$ we have:

(a) $$\frac{\gamma}{2}\left(\sum_{t\in[T]} \ell_t^{pre}(\theta_t) - \sum_{t\in[T]} \ell_t^{pre}(\theta_{true})\right) \le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) - \gamma\sum_{t\in[T]} \|x_t - y_t\|^2 - \sum_{t\in[T]} \langle s_{y_t}, x_t - y_t\rangle;$$

(b) $$\frac{\gamma}{2}\sum_{t\in[T]} \|x_t - x_t^*\|^2 \le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) + \sum_{t\in T} \langle \theta_{true} - \theta_t, c(y_t) - c(x_t^*)\rangle.$$

Proof. The strong convexity of f has a few implications that we will use to derive the desirable statements.

$$f(y_t;\theta_t;u_t) - f(x_t;\theta_t;u_t) \ge \langle s_{x_t}, y_t - x_t\rangle + \frac{\gamma}{2}\|x_t - y_t\|^2 \ge \frac{\gamma}{2}\|x_t - y_t\|^2 \tag{B.2}$$

$$f(x_t^*;\theta_{true};u_t) - f(y_t;\theta_{true};u_t) \ge \langle s_{y_t}, x_t^* - y_t\rangle + \frac{\gamma}{2}\|x_t^* - y_t\|^2 \tag{B.3}$$

$$f(x_t;\theta_{true};u_t) - f(x_t^*;\theta_{true};u_t) \ge \langle s_{x_t^*}, x_t - x_t^*\rangle + \frac{\gamma}{2}\|x_t - x_t^*\|^2 \ge \frac{\gamma}{2}\|x_t - x_t^*\|^2. \tag{B.4}$$

(a) We apply (B.2) and (B.3) to bound the loss gap.

$$
\begin{aligned}
&\frac{\gamma}{2}\left(\sum_{t\in[T]}\ell_t^{pre}(\theta_t) - \sum_{t\in[T]}\ell_t^{pre}(\theta_{true})\right) = \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 - \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 \\
&\le \sum_{t\in[T]} f(y_t;\theta_t;u_t) - f(x_t;\theta_t;u_t) - \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 = \sum_{t\in[T]}\ell_t^{sub}(\theta_t) - \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 \\
&\le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) + \sum_{t\in[T]}\ell_t^{sub}(\theta_{true}) - \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 \\
&= R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) + \sum_{t\in[T]}\left(f(y_t;\theta_{true};u_t) - f(x_t;\theta_{true};u_t)\right) - \frac{\gamma}{2}\sum_{t\in[T]}\|x_t - y_t\|^2 \\
&\le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) - \gamma\sum_{t\in[T]}\|x_t - y_t\|^2 - \langle\sigma_{y_t}; x_t - y_t\rangle;
\end{aligned}
$$

where the first inequality uses (B.2), the second inequality follows from Proposition 6(b), and the last inequality uses (B.3).

(b) We first use (B.4) to bound the squared distance, then apply Proposition 6 (b).

$$
\begin{aligned}
\frac{\gamma}{2}\sum_{t\in[T]}\|x_t - x_t\|^2 &\le \sum_{t\in T} f(x_t;\theta_{true};u_t) - f(x_t;\theta_{true};u_t) \\
&\le R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]}) + \sum_{t\in T}\langle\theta_{true} - \theta_t; c(y_t) - c(x_t)\rangle :
\end{aligned}
$$

In the imperfect information case, a sublinear bound on $R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]})$ has similar implications as in the perfect information, that is, it indicates a vanishing gap between the average ℓ^{sim} incurred by estimates $\{\theta_t\}$ and by the offline optimal estimate. For the other three loss functions, we focus on the difference between the total losses from $\{\theta_t\}$ and from θ_{true}, namely the total imperfect information losses. By Proposition 6(a), with a sublinear ℓ^{sim}-based regret bound, the difference between the average ℓ^{sub} incurred by $\{\theta_t\}$ and the average imperfect information loss vanishes overtime. This means the generated estimates eventually perform at least as well as θ^{true} with respect to ℓ^{sub}.

For ℓ^{est} and ℓ^{pre}, we have weaker results. Proposition 6(b) shows that the average loss difference between $\{\theta_t\}$ and θ_{true} decreases overtime to be below a noise-dependent limit $\frac{1}{T}\sum_{t\in T}\langle\theta_{true} - \theta_t; c(y_t) - c(x_t)\rangle$. Lastly, in the special case of a strongly convex forward objective, Corollary 2(a) bounds the prediction loss gap with $R_T(\{\ell_t^{sim}\}_{t\in[T]};\{\theta_t\}_{t\in[T]})$ and additional noise-dependent terms. Corollary 2(b) proves that the same bound on the average loss difference based on ℓ^{est} applies to the average squared norm distance between the predicted actions and the true actions.

Note that the experiment results from learning a quadratic utility function given in Section 5.6.4 illustrate these theoretical bounding relations. Moreover, as we next show, the results from learning a CES utility function demonstrate similar patterns.