Multi-agent social choice for dynamic fairness-aware recommendation

ANONYMOUS AUTHOR(S)

Algorithmic fairness in the context of personalized recommendation presents significantly different challenges to those commonly encountered in classification tasks. Researchers studying classification have generally considered fairness to be a matter of achieving equality of outcomes between a protected and unprotected group, and built algorithmic interventions on this basis. We argue that fairness in real-world application settings in general, and especially in the context of personalized recommendation, is much more complex and multi-faceted, requiring a more general approach. We propose a model to formalize multistakeholder fairness in recommender systems as a two stage social choice problem. In particular, we express recommendation fairness as a novel combination of an allocation and an aggregation problem, which integrate both fairness concerns and personalized recommendation provisions, and derive new recommendation techniques based on this formulation. Simulations demonstrate the ability of the framework to integrate multiple fairness concerns in a dynamic way.

CCS Concepts: • **Information systems → Recommender systems**; • **Computing methodologies → Multi-agent systems**; • **Social and professional topics → User characteristics**.

Additional Key Words and Phrases: recommender systems, fairness, computational social choice

ACM Reference Format:
Anonymous Author(s). 2022. Multi-agent social choice for dynamic fairness-aware recommendation. In *FAccT '22: ACM Conference on Fairness, Accountability and Transparency, June 21–24, 2022, Seoul, South Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/xx.yyyy/zzzzzz.wwwwwww

1 Introduction

Recommender systems are personalized machine learning systems that support users' access to information in applications as disparate as rental housing, video streaming, job seeking, social media feeds and online dating. The challenges of ensuring fair outcomes in such systems have been addressed in a growing body of research literature surveyed by Ekstrand et al [12]. Despite these research efforts, some key limitations have remained unaddressed, limitations that render this work inadequate for the applications for which it is intended.

The first limitation we see in current work is that researchers have generally assumed that the problem of group fairness can be reduced to the problem of ensuring equality of outcomes between a protected and unprotected group, or in the case of individual fairness, that there is a single type of fairness to be addressed for all individuals. Where fairness for multiple groups has been considered (e.g., Kearns et al. [14], Sonboli et al. [25]), it is defined in the same way for all groups.

We believe that this limitation is severe and not representative of realistic recommendation tasks in which fairness is sought. US anti-discrimination law, for example, identifies multiple protected categories relevant to settings such as housing, education and employment including gender, religion, race, age, and others [3]. But even in the absence of such external criteria, it seems likely that any setting in which fairness is a consideration will need to incorporate the viewpoints of multiple groups.

We also expect that fairness will mean different things for different groups. Consider, for example, a system recommending news articles. Fairness might require that, over time, readers see articles that are geographically representative of their region: rural and urban or uptown vs downtown, for example. But fairness in presenting viewpoints might also require that any given day's set of headlines represent a range of perspectives. These are two different views of what fairness means, entailing different measurements and potentially different types of algorithmic interventions.

The second limitation that we see in current work is that fairness-aware interventions in recommender systems as well as many other machine learning contexts, have a static quality. In many applications, a system is optimized for some criterion and when the optimization is complete, it produces decisions or recommendations based on that learned state [21]. We think of fairness as a dynamic state, especially when what is of primary concern are fair outcomes. A recommender system's ability to produce outcomes that meet some fairness objective may be greatly influenced by context: what items are in inventory, what types of users arrive, how fair the most recent set of recommendations has been, and many others. A static policy runs the risk of failing to capitalize on opportunities to pursue fairness when they arise and/or trying to impose fairness when its cost is high, by not being sensitive to the context.

Our contribution in this paper is the design of an architecture for implementing fairness in recommender systems that addresses both of these limitations. We start from the assumption that multiple fairness concerns will be active at any one time, and that these fairness concerns can be relatively unrestricted in form. Secondly, we build the framework to be dynamic in that decisions are always made in the context of historical choices and results.

Our research in fairness examines concepts inspired by the application context of Kiva Microloans, which offers a platform (Kiva.org) for crowd-sourcing the funding of microloans, mostly in the developing world. Kiva's users (lenders) choose among the loan opportunities offered on the platform; microloans from multiple lenders that are aggregated and distributed through third party non-governmental organizations around the world. Kiva Microloans' mission specifically includes considerations of "global financial inclusion"; as such, incorporating fairness in its recommendation of loans to potential users (lenders) is a key goal. We will use Kiva's platform as an example throughout this paper. However, the analytic findings are not specific to this setting.

2 Formalizing Fairness Concerns

A central tenet of our work is that fairness is a contested concept [19]. From an application point of view, this means that ideas about fairness will be grounded in specific contexts and specific stakeholders, and that these ideas will be multiple and possibly in tension with each other. From a technical point of view, this means that any fairness-aware recommender system should be capable of integrating multiple fairness concepts, arising as they may from this contested terrain.

A central concept in this work is the idea of a *fairness concern*. We define a fairness concern as a specific type of fairness being sought, relative to a particular aspect of recommendation outcomes, evaluated in a particular way. For example, a possible fairness concern in the microlending context might be group fairness relative to different geographical regions considered in light of the exposure of loans from these regions in recommendation lists.[1] The concern identifies a particular aspect of the recommendation outcomes (in this case, their geographical distribution), the particular fairness logic and approach (more about this below), and the metric by which fair or unfair outcomes are determined.

[1]We are currently conducting research to characterize fairness concerns appropriate to Kiva's recommendation applications. At this stage, we can only speculate about the fairness concerns that might arise in that work. None of the discussion here is intended to represent design decisions or commitments to particular concerns and/or their formulation.

The first consideration in building a fairness-aware recommender system is the question of what fairness concerns surround the use of the recommender system, itself. Many such concerns may arise and like any system-building enterprise, there are inevitably trade-offs involved in the formulation of fairness concerns. An organization may decide to incorporate only the highest-priority concerns into its systems. An initial step in fairness-aware recommendation is for an organization to consult its institutional mission and its internal and external stakeholders with the goal of eliciting and prioritizing fairness concerns. An example of this kind of consultation can be seen in the WeBuildAI project [15] and its participatory design framework for AI.

In addition to addressing different aspects of system outcomes, different fairness concerns may invoke different logics of fairness. Welfare economists have identified a number of such logics and we follow Moulin [18] who identifies four:

Exogenous Right: A fairness concern is motivated by exogeneous right if it follows from some external constraint on the system. For example, the need to comply with fair lending regulations may mean that male and female borrowers should be presented proportionately to their numbers in the overall loan inventory.

Compensation: A fairness concern that is a form of compensation arises in response to observed harm or extra costs incurred by one group versus others. For example, loans with longer repayment periods are often not favored by Kiva users because their money is tied up for longer periods. To compensate for this tendency, these loans may need to be recommended more often.

Reward: The logic of reward is operational when we consider that resources may be allocated as a reward for performance. For example, if we know that loans to large cooperative groups are highly effective in economic development, we may want to promote such loans as recommendations so that they are more likely to be funded and realize their promise.

Fitness: Fairness as fitness is based on the notion of efficiency. A resource should go to those best able to use it. In a recommendation context, it may mean matching items closely with user preferences. For example, when loans have different degrees of repayment risk, it may make sense to match the loan to the risk tolerance of the lender.

It is clear that fairness logics do not always pull in the same direction. The invocation of different logics are often at the root of political disagreements: for example, controversies over the criteria for college admissions sometimes pit ideas of reward for achievement against ideas of compensation for disadvantage.

Recommender systems often operate as two-sided platforms, where one set of individuals are receiving recommendations and possibly acting on those recommendations (consumers), and another set of individuals is creating or providing items that may be recommended (providers) [7]. Consumers and providers are considered, along with the platform operator, to be the direct stakeholders in any discussion of recommender system objectives. Fairness concerns may derive from any stakeholder, and may need to be balanced against each other. The platform may be interested in enforcing fairness, even when other stakeholders are not. For example, the average recommendation consumer might only be interested in the best results for themselves, regardless of the impact on others. Fairness concerns can arise on behalf of other, indirect, stakeholders who are impacted by recommendations but not a party to them. An important example is *representational fairness* where we are concerned about the way the outputs of a recommender system operate to represent the world and classes of individuals within it: for example, the way the selection of news articles might end up representing groups of people unfairly [20] (see [12] for additional discussion). As a practical matter, representational fairness concerns can be handled in the same way as provider-side fairness for our purposes here.

Finally, we have the consideration of group versus individual fairness. This dichotomy is well understood as a key difference across types of fairness concerns, defining both the target of measurement of fairness and the underlying principle being upheld. Group fairness requires that we seek fairness across the outcomes relative to predefined protected groups. Individual fairness asks whether each individual user has an appropriate outcome and assumes that users with similar profiles should be treated the same. Just as there are tensions between consumer and provider sides in fairness, there are fundamental incompatibilities between group and individual fairness. Treating all of the outcomes for a group in aggregate is inherently different than maintaining fair treatment across individuals considered separately. Friedler et al. offer a thorough discussion of this topic [13].

Label	Fairness type	Logic	Side	Who is Impacted	Evaluation
LowCountry	Group	Comp.	Provider	Borrowers from countries with lower funding rates	Exposure of loans in recommendation lists
LargeAmt	Group	Reward	Provider	Borrowers in consortia seeking larger loans	Exposure of loans in recommendation lists
Repay	Individual	Reward	Provider	All borrowers	Loan exposure proportional to repayment probability
LowSector	Group	Exo. right	Provider	Borrowers in sectors with lower funding rates	Exposure of loans in recommendation lists
AllCountry	Individual	Exo. right	Provider	All borrowers	Catalog coverage by country
AccuracyLoss	Group	Exo. right	Consumer	All lenders	Accuracy loss due to fairness objective is fairly distributed across protected groups of users.
RiskTolerance	Individual	Fitness	Consumer	All lenders	Riskier loans are recommended to users with greater risk tolerance

Table 1. Potential fairness concerns and their logics.

Putting all of these dimensions together gives us a three-dimensional ontology of fairness concerns in recommendation: fairness logic, consumer- vs provider-side, and group vs individual target. Table 1 illustrates a range of different fairness concerns that are speculatively derived from the microlending context. This list illustrates a number of the points relative to fairness concerns raised so far. We can see that all four of Moulin's fairness logics are represented. We also see that the fairness concerns can be group or individual: for example, we are attentive to individual qualities in the **RiskTolerance** concern, but group outcomes in **LargeAmt**. The **AccuracyLoss** concern is a consumer-side concern, relevant to lenders, but other concerns are on the provider side. We also see that it is possible for a single objective, here the geographic diversity of loan recommendation, to be represented by multiple fairness concerns: **LowCountry** and **AllCountry**. In spite of having the same target, these concerns are distinguished because they approach the objective from different logics and evaluate outcomes differently.

2.1 Fairness Agents

Our architecture SCRUF-D (Social Choice for Recommendation Under Fairness – Dynamic) builds on the SCRUF architecture introduced in [8, 24]. It is designed to allow multiple fairness concerns to operate simultaneously in a recommendation context. Fairness concerns, derived from stakeholder consultation, are instantiated in the form of fairness agents, each having three capabilities:

Evaluation: A fairness agent can evaluate whether the current historical state is fair, relative to its particular concern. Without loss of generality, we assume that this capability is represented by a function m_i for each agent i that takes as input a history of the system's actions and returns an number in the range $[0, 1]$ where 1 is maximally fair and 0 is totally unfair, relative to the particular concern.

Compatibility: A fairness agent can evaluate whether a given recommendation context represents a good opportunity for its associated items to be promoted. We assume that each agent i is equipped with a function c_i that can evaluate a user profile ω and associated information and return a value in the range $[0, 1]$ where 1 indicates the most compatible user and context and 0, the least.

Preference: An agent can compute a preference for a given item whose presence on a recommendation list would contribute (or not) to its particular fairness concern. Again, without loss of generality, we assume this preference can be realized by a function that accepts an item as input and returns a preference score in $\mathbb{R}_+$ where a larger value indicates that an item is more preferred.

2.2 Recommendation Process

We assume a recommendation generation process that happens over a number of time steps t as individual users arrive and recommendations are generated on demand. Users arrive at the system one at a time, receive recommendations, act on them (or not), and then depart. When a user arrives, a recommendation process produces a recommendation list ℓ_s that represents the system's best representation of the items of interest to that user, generated through whatever recommendation mechanism is available. We do not make any assumptions about this process, except that it is focused on the user and represents their preferences. A wide variety of recommendation techniques are well studied in the literature, including matrix factorization, neural embeddings, graph-based techniques, and others.

The first step to incorporating fairness into the recommendation process is to determine which fairness concerns / agents will be active in responding to a given recommendation opportunity. This is the *allocation phase* of the process, the output of which is a set of non-negative weights β, summing to one, over the set of fairness agents, indicating to what extent each fairness agent is considered to be allocated to the current opportunity.

Once the set of fairness agents have been allocated, they have the opportunity to participate in the next phase of the process, which is the *choice phase*. In this phase, all of the active (non-zero weighted) agents and their weights participate in producing a final list of recommendations for the user. We view the recommender system itself as being an agent that participates in this phase.

3 The SCRUF-D Architecture

The two phases of the SCRUF-D architecture are detailed in Figures 1 and 2. The original SCRUF framework [24] concentrated on the representation of user preferences, as computed by the recommender system, and fairness concerns, as derived from stakeholder consultation as discussed in Section 2.1, and their integration. SCRUF-D incorporates the history of system decisions and the fairness achieved over time to control the allocation of fairness concerns. We will first provide a high level overview of the system and describe each figure in detail with formal notation: Table 2 provides a reference to this notation.

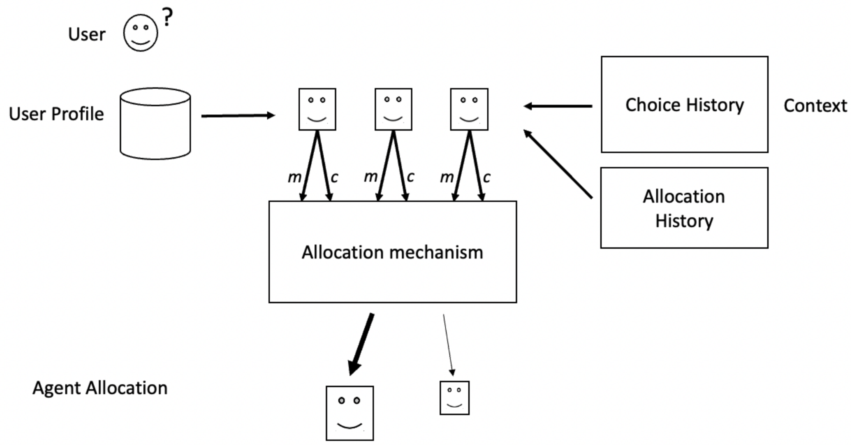


Fig. 1. SCRUF-D Framework / Allocation Phase: Recommendation opportunities are allocated to fairness concerns based on the context.

3.1 Overview

We can think of a recommender system as a two-sided market in which the recommendation opportunities that arise from the arrival of a user $u \in \mathcal{U}$ to the system, and each are allocated to a set of items $v \in \mathcal{V}$ from the system's catalog. This market has some similarities to various forms of online matching markets including food banks [1], kidney allocation [2, 16], and ride sharing [10], in that users have preferences over the items; however, in our case this preference is known only indirectly through either the prior interaction history or a recommendation function. Additionally, the items are not consumable or rivalrous. For example, a loan can be recommended to any number of users – it is not "used up" in the recommendation interaction.[2] Also, users are not bound to the recommendations provided; in most systems including Kiva, there are multiple ways to find items of which the recommender system is only one.

This problem has some similarities with those found in computational advertising, where specific messages are matched with users in a personalized way [26, 27]. Because advertising is a paid service, these problems are typically addressed through mechanisms of monetary exchange, such as auctions. There is no counterpart to budgets or bids in our context, which means that solutions in this space do not readily translate to supporting fair recommendation [11, 28, 29]. Once we have a collection of fairness agents we must solve two interrelated problems: (1) what agent(s) are allocated to a particular recommendation *opportunity* and (2) how do we *balance* between the allocated agents and the user's individual preferences?

Figure 1 shows the first phase of this process, allocation [5], in which we decide which fairness concerns / agents should be allocated to a particular fairness opportunity. This is an online and dynamic allocation problem where we must consider many factors including the history of agent allocations so far, the generated lists from past interactions with users, and how fair the set of agents believes this history to be. As described in Section 2.1, agents take these histories and information about the current user profile and calculate two values: m, a measure of fairness relative to their agent-specific concern, and c, a measure of compatibility between the current context and the agent's fairness concern. The allocation mechanism takes these metrics into account producing a probability distribution over the

[2] Loans on Kiva's platform may be exhausted eventually through being funded, but many other objects of recommendation such as streaming media assets are effectively infinitely available.

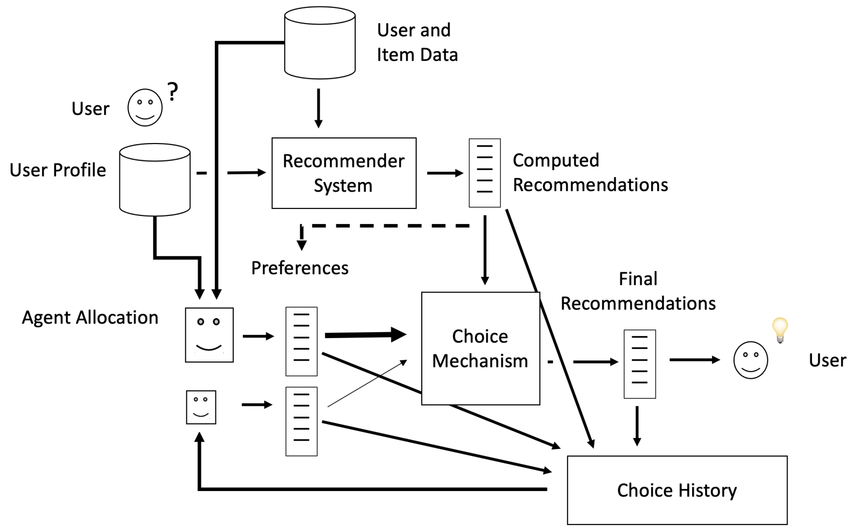


Fig. 2. SCRUF-D Framework / Choice Phase: The preferences derived from the recommender system and the fairness concerns are integrated by the choice mechanism.

fairness agents that we call the *agent allocation*, which can be interpreted as weights in the choice stage or be used to select a single agent via a lottery, e.g., a randomized allocation scheme [6].

In the second phase, shown in Figure 2, the recommender system generates a list of options, considered to represent the user's preferences. The fairness concerns generate their own preferences as well. These preferences may be global in character, i.e., preferences over all items, in which case they may be independent of what the recommender system produces. Or, as indicated by the dashed line, these preferences may be scoped only over the items that the recommender system has generated. In either case, the preference function of the fairness agent, like the one for the user, generates a list of items and scores. The choice mechanism combines these preferences of both the user and fairness agents, along with the allocation weights of the fairness agents, to arrive at a final recommendation list to be delivered to the user. The list itself, and possibly interactions the user has with it, become a new addition to the choice history and the process continues for the next user.

3.2 Formal Description

In our formalization of a recommendation system setting we have a set of users $\mathcal{U} = \{u_1, \dots u_n\}$ and a set of items $\mathcal{V} = \{v_1, \dots, v_m\}$. For each item $v_i \in \mathcal{V}$ we have a k-dimensional feature vector $\phi = \langle \phi_1, \dots \phi_k \rangle$ over a set of categorical features ϕ, each with finite domain. Some of these features may be sensitive, e.g., they are associated with one or more fairness agent concerns, we denote this set as ϕ^s. Without loss of generality, we assume that all elements in $\mathcal{V}$ share the same set of features ϕ. Finally, we assume that each user is associated with a profile of attributes $\omega = \langle \omega_1, \dots \omega_j \rangle$, of which some also may be sensitive $\omega^s \subseteq \omega$, e.g., they are associated with one or more fairness agents.

As in a standard recommendation system we assume that we have (one or more) recommendation mechanism that take a user profile ω and a (set of) items v and produces a predicted rating $\hat{r} \in \mathbb{R}_{+}$. We will often refer to a recommendation list, $\ell = \langle \{v_1, \hat{r}_1\}, \dots \{v_i, \hat{r}_i\} \rangle$, which is generated for user ω by sorting according to $\hat{r}$, i.e., $sort(\mathcal{R}_i(\omega, \mathcal{V})) \rightarrow \ell$. Note that this produces a permutation (ranking) over the set of items for that user, i.e. a recommendation. As a practical

Rec. System	$\mathcal{U}(u)$	Users (user).
	$\mathcal{V}(v)$	Items (item).
	$\phi = \langle \phi_1, \dots \phi_k \rangle$	Item Features.
	$\omega = \langle \omega_1, \dots \omega_j \rangle$	User Profile.
	$\phi^s \subseteq \phi$	Sensitive Item Features as a subset of all item features ϕ.
	$\omega^s \subseteq \omega$	Sensitive Aspects of User Profile as a subset of all user profile features ω.
	$\mathcal{R}_i(\omega, v) \rightarrow \{v, \hat{r}\}$	Recommendation mechanism that takes a user profile ω and a (set of) items v and produces a predicted rating $\hat{r} \in \mathbb{R}_+$.
	$\ell = \langle \{v_1, \hat{r}_1\}, \dots \{v_i, \hat{r}_i\} \rangle$	Recommendation List as an ordered list of item, predicted rating pairs.
	$sort(\mathcal{R}_i(\omega, \mathcal{V})) \rightarrow \ell$	Recommendation List for user ω sorted by $\hat{r}$.
Fairness Agents	$\mathcal{F} = \{f_1, \dots, f_i\}$	Set of Fairness Agents.
	$f_i = \{m_i, c_i, \mathcal{R}_i\}$	Fairness agent i defined by a fairness metric m_i, a compatibility metric c_i, and a ranking function $\mathcal{R}_i$.
	$m_i(\vec{L}, \vec{H}) \rightarrow [0, 1]$	Fairness metric for agent i that takes a choice history $\vec{L}$ and allocation history $\vec{H}$ and produces a value in $[0, 1]$ according to the agent's evaluation of how fair recommendations so far have been.
	$c_i(\omega) \rightarrow [0, 1]$	Compatibility metric for agent i that takes a particular user profile ω and produces a value in $[0, 1]$ for how compatible fairness agent i believes they are for user ω.
	$\mathcal{R}_i(\omega, v) \rightarrow \{v, \hat{r}\}$	Fairness Agent Recommendation function.
	$\ell_{\mathcal{F}} = \{\mathcal{R}_1(\omega, \mathcal{V}), \dots, \mathcal{R}_i(\omega, \mathcal{V})\}$	Set of Fairness Agent Recommendation Lists indexed by fairness agent label i.
Allocation	$\mathcal{A}(\mathcal{F}, m_{\mathcal{F}}(\vec{L}, \vec{H}), c_{\mathcal{F}}(\omega)) \rightarrow \beta \in \mathbb{R}_+^{\lvert\mathcal{F}\rvert}$	Allocation mechanism $\mathcal{A}$ that takes a set of fairness agents $\mathcal{F}$, the agents' fairness metric evaluations $m_{\mathcal{F}}(\vec{L}, \vec{H})$, and the agents' compatibility metric evaluations $c_{\mathcal{F}}(\omega)$ and maps to an agent allocation β.
	$\vec{H} = \langle \beta^1, \dots, \beta^t \rangle$	Allocation History $\vec{H}$ that is an ordered list of agent allocations $\mathcal{A}$ at time t.
Choice	$C(\ell, \beta, \ell_{\mathcal{F}}) \rightarrow \ell_C$	Choice Function Output List as a function from a recommendation list ℓ, agent allocation β, and fairness agent recommendation list(s) $\ell_{\mathcal{F}}$ to a combined list ℓ_C.
	$\vec{L} = \langle \ell^t, \ell^t_{\mathcal{F}}, \ell^t_C \rangle$	Choice History that is an ordered list of user recommendation list ℓ, agent recommendation list(s) $\ell_{\mathcal{F}}$, and choice function output lists ℓ_C, indexed by time step t.

Table 2. Notations for our formal description of the SCRUF-D architecture.

matter, the recommendation results will almost always contain a subset of the total set of items, typically the head (prefix) of the permutation up to some cutoff number of items or score value. For ease of exposition we assume we are able to score all items in the database.

In the SCRUF-D architecture, fairness concerns map directly onto agents $\mathcal{F} = \{f_1, \dots, f_i\}$. In order for the agents to be able to evaluate their particular concerns, they take account of the current state of the system and voice their evaluation of how fairly the overall system is currently operating, their compatibility for the current recommendation opportunity, and their preference for how to make the outcomes more fair. Hence, each fairness agent i is described as a set, $f_i = \{m_i, c_i, \mathcal{R}_i\}$ consisting of a fairness metric, $m_i(\vec{L}, \vec{H}) \rightarrow [0, 1]$, that takes a choice history $\vec{L}$ and allocation history $\vec{H}$ and produces a value in $[0, 1]$ according to the agent's evaluation of how fair recommendations so far have been; a compatibility metric, $c_i(\omega) \to [0, 1]$, that takes a particular user profile ω and produces a value in $[0, 1]$ for how compatible fairness agent i believes they are for user ω; and a ranking function, $\mathcal{R}_i(\omega, v) \to \{v, \hat{r}\}$, that gives the fairness agent preferences.

In the allocation phase (Figure 1), we must allocate a set of fairness agents to a recommendation opportunity. Formally, this is an allocation function, $\mathcal{A}(\mathcal{F}, m_{\mathcal{F}}(\vec{L}, \vec{H}), c_{\mathcal{F}}(\omega)) \to \beta \in \mathbb{R}_{+}^{|\mathcal{F}|}$ that takes a set of fairness agents $\mathcal{F}$, the agents' fairness metric evaluations $m_{\mathcal{F}}(\vec{L}, \vec{H})$, and the agents' compatibility metric evaluations $c_{\mathcal{F}}(\omega)$ and maps to an agent allocation β, where β is a probability distribution over the agents $\mathcal{F}$. The allocation function itself is allocating fairness agents to recommendation opportunities by considering both the fairness metric for each agent as well as each fairness agent's estimation of their compatibility.

The allocation function can take many forms, e.g., it could be a simple function of which ever agent voices the most unfairness in the recent history [24], or it could be a more complex function from social choice theory such as the probabilistic serial mechanism [4] or other fair division or allocation mechanisms. Note here that the allocation mechanisms is directly comparing the agent valuations of both the current system fairness and compatibility. Hence, we are implicitly assuming that the agent fairness evaluations are comparable. While this is a somewhat strong assumption, it is less strong than assuming that fairness and other metrics, e.g., utility or revenue, are comparable as is common in the literature [30]. So, although we are assuming different voicing of fairness are comparable, we are only assuming that fairness is comparable with fairness, and not other aspects of the system. We plan to explore options for the allocation function in our empirical experiments. We track the outputs of this function as the allocation history, $\vec{H} = \langle \beta^1, \ldots, \beta^t \rangle$, an ordered list of agent allocations β at time t.

In the second phase of the system (Figure 2), we must take the set of allocated agents and combine their preferences (and weights) with those of the current user ω. To do this we define a choice function, $C(\ell, \beta, \ell_{\mathcal{F}}) \to \ell_C$, as a function from a recommendation list ℓ, agent allocation β, and fairness agent recommendation list(s) $\ell_{\mathcal{F}}$ to a combined list ℓ_C. Each of the fairness agents is able to express their preferences over the set of items for a particular user, $\mathcal{R}_i(\omega, v) \to \{v, \hat{r}\}$, and we take this set of lists,$\ell_{\mathcal{F}} = \{\mathcal{R}_1(\omega, \mathcal{V}), \ldots, \mathcal{R}_i(\omega, \mathcal{V})\}$, as input to the choice function that generates a final recommendation that is shown to the user, ℓ_C.

We again leave this choice function unspecified as this formulation provides a large design space: we could use a simple voting rule, a simple additive utility function or something much more complicated like rankings over the set of all rankings [5]. Note that the choice function can use the agent allocation β as either a lottery to, e.g., select one agent to voice their fairness concerns, or as a weighting scheme. We will investigate a range of choice functions in further research. In order for the fairness agents to be able to evaluate the status of the system we also track the choice history, $\vec{L} = \langle \ell^t, \ell^t_{\mathcal{F}}, \ell^t_C \rangle$, as an ordered list of user recommendation list ℓ, agent recommendation list(s) $\ell_{\mathcal{F}}$, and choice function output lists ℓ_C, indexed by time step t.

4 Design Considerations

Within this framework there are a number of important design considerations to take into account for any particular instantiation of the SCRUF-D architecture. We have left many of the particular design choices open for future investigation. We allow for any type of recommendation algorithm; fairness agents may incorporate any type of compatibility function or fairness evaluation function. Similarly, we do not constrain the allocation or choice mechanisms. With SCRUF-D, we are able to explore many definitions of fairness and recommendation together in a principled uniform way. In this section, we discuss a few of the design parameters that may be explored in future work.

4.1 Agent Design

We can expect that an agent associated with a fairness concern will typically have preferences that order items relative to a particular feature or features associated with that concern. Items more closely related to the sphere of concern will be ranked more highly and those unrelated, lower. However, this property means that agents associated with different concerns might have quite different rankings – the gender parity concern will rank women's loans highly regardless of their geography, for example. Thus, we cannot assume consistency or single-peakedness across the different agents.

As noted above, agents may have preferences over disjoint sets of items or they may be constrained only to have preferences over the items produced by the recommender system for the given user. This second option corresponds to a commonly-used *re-ranking* approach, where the personalization aspect of the system controls what items can be considered for recommendation and fairness considerations re-order the list [12]. If an agent can introduce any item into its preferences, then we may have the challenge in the choice phase of integrating items that are ranked by some agents but not others. Some practical work-arounds might include a constraint on the recommender system to always return a minimum number of items of interest to the allocated agents or a default score to assign to items not otherwise ranked.

Despite our terminology, it is clear that our architecture as described is sufficiently general that an agent could be designed that pushes the system to act in harmful and unfair ways rather than beneficial and fairness-enhancing ones. Thus, the importance of the initial step of stakeholder consultation and the careful crafting of fairness concerns. Because fairness concerns are developed within a single organization and with beneficence in mind, we assume that we do not need to protect against adversarial behavior, such as collusion among agents or strategic manipulation of preferences. The fact that the agents are all "on the same team" allows us to avoid constraints and complexities that otherwise arise in multi-agent decision contexts.

4.2 Agent Efficacy

The ability of an agent to address its associated fairness concern in non-deterministic. It is possible that the agent may be allocated to a particular user interaction, but its associated fairness metric may still fail to improve. One likely reason for this is the primacy of the personalization objective. Generally, we expect that the user's interests will have the greatest weight in the final recommendations delivered. Otherwise, the system might have unacceptably low accuracy, and fail in its primary information access objective.

One design decision therefore is whether (and how) to track agent efficacy as part of the system history. If the agent's efficacy is generally low, then opportunities to which it is suited become particularly valuable; they are the rare situations in which this fairness goal can be addressed. Another aspect of efficacy is that relationships among item characteristics may mean that a given agent, while targeted to a specific fairness concern, might have the effect of enhancing multiple dimensions of fairness at once. Consider a situation in which geographic concerns and sectoral concerns intersect. Promoting an under-served region might also promote an under-served economic sector. Thus, the empirically-observed multidimensional impact of a fairness concern will need to be tracked to represent its efficacy.

Efficacy may also be a function of internal parameters of the agent itself. A separate learning mechanism could then be deployed to optimize these parameters on the basis of allocation, choice and user interaction outcomes.

4.3 Mechanism Inputs

Different SCRUF implementations may differ in what aspects of the context are known to the allocation and/or choice mechanisms. Our hope is that we can leverage social choice functions in order to limit the complexity of the information that must be passed to the allocation and/or choice mechanisms. However, if a sophisticated and dynamic representation of agent efficacy is required, it may be necessary to implement a bandit-type mechanism to explore the space of allocation probabilities and/or agent parameters as discussed above. Recent research on multidimensional bandit learning suggests possible approaches here [17].

4.4 Agent Priority

As we have shown, agent priority in the allocation phase may be a function of user interests, considering different users as different opportunities to pursue fairness goals. It may also be a function of the history of prior allocations, or the state of the fairness concerns relative to some fairness metric we are trying to optimize. As the efficacy consideration would indicate, merely tracking allocation frequency is probably insufficient and it is necessary to tie agent priority to the state of fairness. Allocation priority is also tied to efficacy as noted above. It may be necessary to compute expected fairness impact across all dimensions in order to optimize the allocation.

We plan to leverage aspects of social choice theory to help ameliorate some of these issues. There is a significant body of research on allocation and fair division mechanisms that provide a range of desirable normative properties including envy-freeness [9], e.g., the guarantee that one agent will not desire another agent's allocation, Pareto optimally, e.g., that agents receive an allocation that is highly desirable according to their compatibility evaluations [4]. An important and exciting direction for research is understanding what allocation properties can be guaranteed for the SCRUF-D architecture overall depending on the allocation mechanism selected [5].

We note that in most practical settings the personalization goal of the system will be most important and therefore the preference of this agent will have topmost priority. It is always allocated and is not part of the allocation mechanism. Thus, we cannot assume that the preference lists of the agents that are input to the choice system are anonymous, a common assumption in the social choice literature on voting [5].

4.5 Bossiness

Depending on how the concept of agent / user compatibility is implemented, it may provide benefits to *bossy* users, those with very narrow majoritarian interests that do not allow for the support of the system's fairness concerns. Those users get results that are maximally personalized and do not share in any of the potential accuracy losses associated with satisfying the system's fairness objectives. Other, more tolerant users, bear these costs. A system may wish to ensure that all users contribute, at some minimal level, to the fairness goals. In social choice theory, a mechanism is said to be non-bossy if an agent cannot change the allocation without changing the allocation that they receive by modifying their preferences [22].

4.6 Fairness Types

We concentrate in this paper and our work with Kiva generally on provider-side group fairness, that is characteristics of loans where protected groups can be distinguished. However, it is also possible to use the framework for other fairness requirements. On the provider side, an individual fairness concern is one that tracks individual item exposure as opposed to the group as a whole. It would have a more complex means of assessing preference over items and of assessing fairness state, but still fits within the framework.

Consumer-side fairness can also be implemented through use of the compatibility function associated with each agent. For example, the example of assigning risk appropriately based on user risk tolerance becomes a matter of having a risk reduction agent that reports higher compatibility for users with lower risk tolerance.

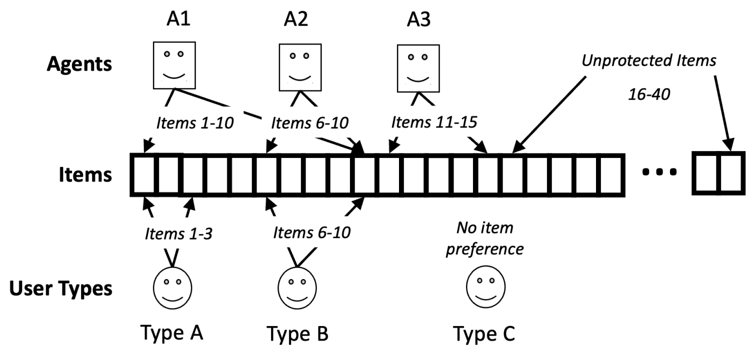


Fig. 3. Experimental setup. Multiple fairness agents interact with multiple user types.

5 Experimental Studies

5.1 Experimental setup

As an initial examination of the properties of the SCRUF-D architecture, we constructed a simple simulation. A sketch of the evaluation configuration is shown in Figure 3. There are three fairness agents A1, A2, and A3. Each agent has a set of items of interest. The first agent A1 treats items 1-10 as protected for its purposes, the second A2, items 6-10 and the third A3, items 11-15. The remaining items 16-40 are unprotected items, which are not advanced by any agent. The overlap between A1 and A2 means that when A2 is selected it will be promoting items that also satisfy A1. A3 is the "pickiest" agent, the one with the most difficult to satisfy fairness constraint, because it only has 5 items and they are not of interest to any other agent.

User profiles are generated randomly from user types. Each user type consists of a table of target ratings for each item along with bias and noise parameters, which are used to generate recommendations. There are three user types. Type A is narrowly interested in items 1-3. Type B is interested in items 6-10, coinciding with Agent 2. The third user type has references that vary randomly across all items.

Recommendations are generated by selecting a user type at random and generating recommendations for each item based on the target rating, bias and noise parameters. The recommended items are ranked and the top 10 items are returned. The accuracy of these recommendations is measured by the rankin accuracy (NDCG@10) of this list relative to the target ratings for the user profile type.

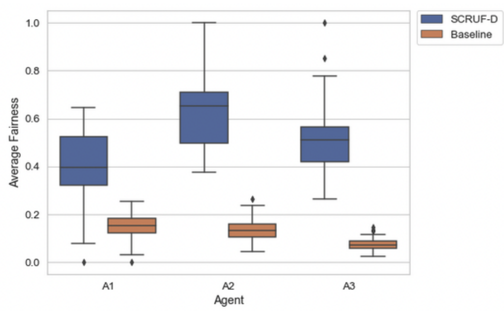
All of the agents evaluate fairness in the same way. The recommendation lists from the previous algorithm iterations are combined. The fraction of this combined list that consists of the item-specific associated items is computed. If the fraction is equal or greater than 0.5, then the metric returns 1 (maximum fairness). If the list contains a smaller fraction, the fairness is twice the fraction so that it scales smoothly from 1 to 0.

For the purposes of these experiments, we ignore the compatibility function and allow all agents to be equally compatible with all users. The allocation mechanism uses a lottery as in [24], choosing a single agent with probabilities determined by the fairness scores. If all agents have a score of 1, which happens at the beginning of the simulation, no agents are allocated.

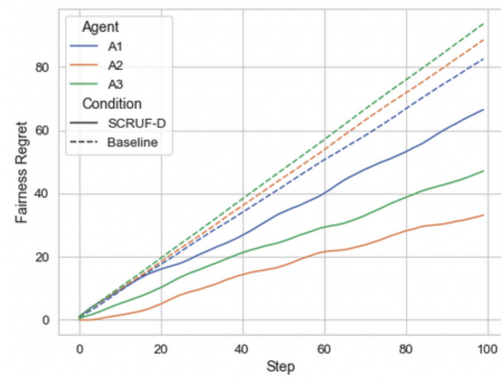
After the fairness agent is allocated, the scores from the recommender systems are adjusted such that each item among the v^p protected items has its score augmented by the constant $\delta = 0.5$. Then the recommendations are sorted and the top ten items chosen as the recommendation list. This list is evaluated for accuracy with NDCG@10.

5.2 Results

Figure 4a shows the average agent fairness in recommendation lists across the whole experiment, compared to a baseline in which no reranking is performed. As we can see, the fairness increases for all the protected item groups, but especially for Agent 2, since, as expected, it benefits from the allocation of Agent 2 as well as itself. Agent 3 also fares well since it starts from a lower point. Note that the baseline implementation achieved a cumulative NDCG@10 of 3.94; the addition of the fairness agents reduces this by about 8% to 3.61 in order to achieve improved fairness.



(a) Fairness results



(b) Cumulative fairness loss (regret)

Fig. 4. Results from simulation experiment.

Figure 4b shows cumulative fairness loss (regret), calculated as $(1 - m_i)$ in each time interval over a sample simulation run. The three colored lines indicate the different fairness agents. The corresponding dashed lines show the sample calculation but for the recommender alone without reranking. The figure shows the improvement in fairness for all agents as expected, and particularly for Agent 2.

We also conducted an experiment in which users of Type B did not appear in the test run until time step 50, and then users of Type A do not appear after this time. This means that, although Agent 2 does get allocated in the initial phase, those interventions are unlikely to be effective. The baseline recommender would have to score items 6-10 highly enough so that augmenting their score by δ is sufficient to get them in to the top 10. As we can see in Figure 5, the regret for Agent 2 increases steadily during this time. (Agent 3 also increases but this is due to the fact that its fairness condition is harder to satisfy.)

After time 50, the interventions by Agent 2 become more effective and Agent 1 not longer has Type A users to draw from. The fairness gap closes and then turns around as Agent 1's effectiveness changes. The change in the outcome for Agent 3 may seem unexpected since, but note that users of Type A have very narrow interests and opportunities for Agent 3 are rare in the first phase of the experiment.

We note also that, because the compatibility function was not included in this simulation, the allocation mechanism does not "notice" when an Agent-2-compatible user (Type B) comes along, and the choice of Agent 2 is not influenced by this knowledge.

These experiments show that the SCRUF-D architecture is capable of representing and applying multiple fairness concerns in a modular and agent-based way and balancing among them dynamically. Thorough empirical evaluation of

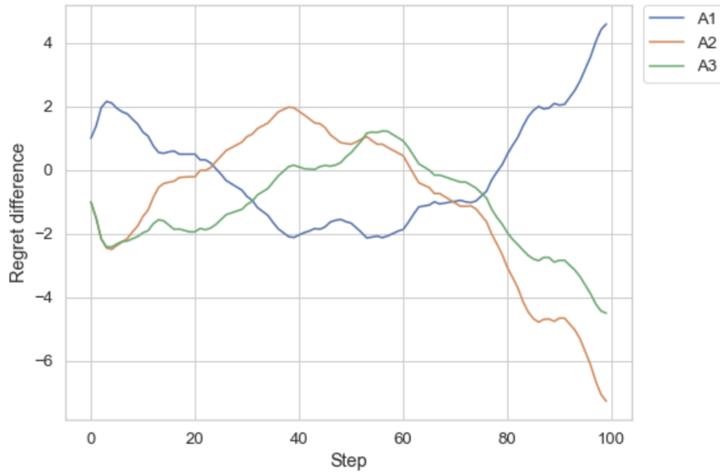


Fig. 5. Difference in fairness regret vs baseline. Negative values are better, representing greater fairness.

the architecture with real data and fairness concerns is a subject for future work, as is the incorporation and study of a full range of allocation and choice mechanisms.

6 Conclusion and Future Work

We have introduced the SCRUF-D architecture for integrating multiple fairness concerns into recommendation generation leveraging social choice. The design is general and allows for many different types of fairness concerns—involving multiple fairness logics and encompassing both provider and consumer aspects of the recommendation platform. The architecture is also general in that it makes few assumptions about the nature of the allocation and choice mechanisms by which fairness is maintained, allowing for a large design space incorporating many types of functions. We have included experiments that demonstrate the basic properties of the architecture in a synthetic setting.

Future work will proceed in multiple research arcs. One arc of future work is to apply the architecture in more realistic settings, particularly with Kiva. We are working with Kiva stakeholders and beginning the process of identifying fairness concerns. In the meantime, we also plan to conduct additional experiments with a variety of off-line data sets, exploring a range of different fairness concern formalizations and social choice options.

We have made the mechanisms and the agents fairly simple by design. Further experimentation will show how effective this structure is for maintaining fairness over time and allowing a wide variety of fairness concerns to be expressed. However, there are some areas of exploration that we can anticipate.

A key feature of the recommendation context is that the decisions of the recommender system only influence the exposure of protected items. There is no guarantee that a given user will show any interest in an item just because it is presented. In some settings and for some fairness concerns, exposure might be enough. But in cases where utility derives from usage rather than exposure, there would be some value in having the system learn about the relationship between exposure and utility. This setting has the attributes of a multi-objective bandit learning problem [17], where the fairness concerns represent different classes of rewards and the allocation of agents represents different choices.

Even when we consider exposure as our main outcome of interest, it is still the case that the allocation of different agents may result in differential improvements in fairness. Perhaps the items associated with one agent are more common in recommendation lists and can be easily promoted through re-ranking while other agents' items are not.

The weight associated with the allocation of agents may need to be adjusted to reflect the expected utility of allocation, and this expected utility would need to be learned as in the case above.

The current architecture does not make any assumptions about the distribution of user characteristics. That is, suppose fairness concern f_i is "difficult" to achieve in that users with an interest in related items appear rarely. In that case, we should probably allocate f_i whenever a compatible user arrives, regardless of the state of the fairness metrics. This example suggests that the allocation mechanism could be adapted to look forward (to the distribution of future opportunities) as well as backwards (over fairness results achieved). This would require a model of opportunities similar to [23], and others studied in computational advertising settings.

The current architecture envisions fairness primarily in the context of group fairness expressed over recommendation outcomes. We believe that the architecture will support other types of fairness with additional enhancements. For example, a representational fairness concern would be incompatible with the assumption that fairness can be aggregated over multiple recommendation lists. Consider the examples in Noble's *Algorithms of Oppression*: it would not be acceptable for a recommender system to deliver racist or sexist results at times, even if those results were balanced out in some overall average. Representational fairness imposes a stricter constraint than those considered here, effectively requiring that the associated concern be allocated for every recommendation opportunity.

The model expressed here assumes that fairness agents have preferences only over items. But it is also possible to represent agents as having preferences over recommendation lists. This would allow agents to express preferences for combinations of items: for example, a preference that there be at least two Agriculture loans in the top 5 items of the list. This kind of preference cannot be expressed simply in terms of scores associated with items. Agents would naturally have to become more complex in their ability to reason about and generate such preferences, and the choice mechanism would become more like a combinatorial optimization problem. It is possible that we can characterize useful subclasses of the permutation space and avoid the full complexity of arbitrary preferences over subsets.

Another interesting direction for research is more theoretical in nature. Much of the research in social choice focuses on providing guaranteed normative properties of various mechanisms. However, the models used in traditional social choice theory do not take into consideration the dynamics of recommender systems as most mechanisms are designed to work in one-off scenarios without dynamic aspects. As such, it will also be important to understand the properties of existing social choice mechanisms for allocation and choice when deployed in these dynamic contexts and to develop new methods with good properties.

References

[1] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. 2015. Online Fair Division: Analysing a Food Bank problem. In *Proc. 24th International Joint Conference on AI (IJCAI)*. IJCAI, 2540–2546.

[2] P. Awasthi and T. Sandholm. 2009. Online Stochastic Optimization in the Large: Application to Kidney Exchange.. In *Proc. 21st International Joint Conference on AI (IJCAI)*. IJCAI, 405–411.

[3] Solon Barocas and Andrew D Selbst. 2016. Big Data's Disparate Impact. *California law review* 104, 3 (2016), 671. https://doi.org/10.15779/Z38BG31

[4] Anna Bogomolnaia and Hervé Moulin. 2001. A new solution to the random assignment problem. *Journal of Economic theory* 100, 2 (2001), 295–328.

[5] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[6] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *American economic review* 103, 2 (2013), 585–623.

[7] Robin Burke. 2017. Multisided Fairness for Recommendation. In *Workshop on Fairness, Accountability and Transparency in Machine Learning (FATML)*. Halifax, Nova Scotia, 5 pages. https://arxiv.org/abs/1707.00093

[8] Robin Burke, Amy Voida, Nicholas Mattei, and Nasim Sonboli. 2020. Algorithmic Fairness, Institutional Logics, and Social Choice. In *Harvard CRCS Workshop on AI for Social Good at 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*. 5 pages.

[9] Yuga J Cohler, John K Lai, David C Parkes, and Ariel D Procaccia. 2011. Optimal envy-free cake cutting. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[10] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. 2018. Allocation Problems in Ride Sharing Platforms: Online Matching with Offline Reusable Resources. In *Proc. Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 1007–1014.

[11] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American economic review* 97, 1 (2007), 242–259.

[12] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. arXiv:2105.05779 [cs.IR]

[13] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2021. The (Im)possibility of fairness: different value systems require different mechanisms for fair decision making. *Commun. ACM* 64, 4 (April 2021), 136–143. https://doi.org/10.1145/3433949

[14] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. arXiv:1711.05144 [cs.LG]

[15] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, and Alexandros Psomas. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.

[16] N. Mattei, A. Saffidine, and T. Walsh. 2018. An Axiomatic and Empirical Analysis of Mechanisms for Online Organ Matching. In *Proceedings of the 7th International Workshop on Computational Social Choice (COMSOC)*. 24 pages.

[17] Rishabh Mehrotra, Niannan Xue, and Mounia Lalmas. 2020. Bandit based Optimization of Multiple Objectives on a Music Streaming Platform. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3224–3233.

[18] Hervé Moulin. 2004. *Fair division and collective welfare*. MIT press.

[19] Deirdre K Mulligan, Joshua A Kroll, Nitin Kohli, and Richmond Y Wong. 2019. This thing called fairness: disciplinary confusion realizing a value in technology. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–36.

[20] Safiya Umoja Noble. 2018. *Algorithms of Oppression: How search engines reinforce racism*. NYU Press.

[21] Cathy O'Neil. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books.

[22] Szilvia Pápai. 2000. Strategyproof assignment by hierarchical exchange. *Econometrica* 68, 6 (2000), 1403–1433.

[23] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. 2012. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 804–812.

[24] Nasim Sonboli, Robin Burke, Nicholas Mattei, Farzad Eskandanian, and Tian Gao. 2020. "And the Winner Is...": Dynamic Lotteries for Multi-group Fairness-Aware Recommendation. arXiv:2009.02590 [cs.IR]

[25] Nasim Sonboli, Farzad Eskandanian, Robin Burke, Weiwen Liu, and Bamshad Mobasher. 2020. Opportunistic Multi-Aspect Fairness through Personalized Re-Ranking. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (Genoa, Italy) *(UMAP '20)*. Association for Computing Machinery, New York, NY, USA, 239–247. https://doi.org/10.1145/3340631.3394846

[26] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. arXiv:1610.03013 [cs.GT]

[27] Shuai Yuan, Ahmad Zainal Abidin, Marc Sloan, and Jun Wang. 2012. Internet Advertising: An Interplay among Advertisers, Online Publishers, Ad Exchanges and Web Users. arXiv:1206.1754 [cs.IR]

[28] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.

[29] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1077–1086.

[30] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1153–1162.