Dynamic Fairness-aware Recommendation Through Multi-agent Social Choice

AMANDA AIRD, Information Science, University of Colorado at Boulder, Boulder, United States
PARESHA FARASTU, Computer Science, University of Colorado at Boulder, Boulder, United States
JOSHUA SUN, Computer Science, University of Colorado at Boulder, Boulder, United States
ELENA STEFANCOVÁ, Computer Science, Comenius University in Bratislava, Bratislava, Slovakia
CASSIDY ALL, Computer Science, University of Colorado at Boulder, Boulder, United States
AMY VOIDA, Information Science, University of Colorado at Boulder, Boulder, United States
NICHOLAS MATTEI, Computer Science, Tulane University, New Orleans, United States
ROBIN BURKE, Information Science, University of Colorado at Boulder, Boulder, United States

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 (or some other metric) 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. To address the fundamental problem of fairness in the presence of multiple stakeholders, with different definitions of fairness, we propose the Social Choice for Recommendation Under Fairness–Dynamic architecture, which formalizes multi-stakeholder fairness in recommender systems as a two-stage social choice problem. In particular, we express recommendation fairness as a 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. We demonstrate the ability of our framework to dynamically incorporate multiple fairness concerns using both real-world and synthetic datasets.

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

Authors Burke, Voida and Aird were supported by the National Science Foundation under grant awards IIS-1911025 and IIS-2107577. Nicholas Mattei was supported by NSF Grant IIS-2107505. Author Stefancova was supported by Slovak Research and Development Agency under Contract no. APVV-20-0353 and the Fulbright program.

Authors' Contact Information: Amanda Aird, Information Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: amanda.aird@colorado.edu; Paresha Farastu, Computer Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: paresha.farastu@colorado.edu; Joshua Sun, Computer Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: joshua.sun@colorado.edu; Elena Stefancová, Computer Science, Comenius University in Bratislava, Bratislava, Bratislava, Slovakia; e-mail: elenastefancova@gmail.com; Cassidy All, Computer Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: cassidy.all@colorado.edu; Amy Voida, Information Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: amy.voida@colorado.edu; Nicholas Mattei, Computer Science, Tulane University, New Orleans, Louisiana, United States; e-mail: nsmattei@tulane.edu; Robin Burke, Information Science, University of Colorado at Boulder, Boulder, Colorado, United States; e-mail: robin.burke@colorado.edu.

ACM 2770-6699/2024/11-ART21
https://doi.org/10.1145/3690653

ACM Reference Format:
Amanda Aird, Paresha Farastu, Joshua Sun, Elena Stefancová, Cassidy All, Amy Voida, Nicholas Mattei, and Robin Burke. 2024. Dynamic Fairness-aware Recommendation through Multi-agent Social Choice. *ACM Trans. Recomm. Syst.* 3, 2, Article 21 (November 2024), 35 pages. https://doi.org/10.1145/3690653

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, including the unique issues that come from recommendation ecosystems, have been addressed in a growing body of research literature surveyed in several works, including Ekstrand et al. [21] and Patro et al. [45]. Despite these research efforts, some key limitations have remained unaddressed, including the dynamic and multistakeholder nature of recommendations systems, and these limitations leave many solutions inadequate for the full range of applications for which recommender systems are deployed.

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 and that there is a single type of fairness to be addressed for all individuals. Fairness is a complex, multifaceted concept that can and does have different definitions, at different times, to different stakeholders, and these issues must be considered in the context in which the systems are deployed [28, 48].

We believe that limiting a system to a single definition of fairness is a severe restriction and not representative of realistic recommendation tasks in which fairness is sought. U.S. anti-discrimination law, for example, identifies multiple protected categories and different definitions of fairness for many of these categories in settings such as housing, education, and employment, including gender, religion, race, age, and others [5]. But even in the absence of such external requirements, 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. 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 diversity of fairness definitions in a single system is rarely addressed: Where fairness for multiple groups has been considered (e.g., Sonboli et al. [52] and Kearns et al. [31]), it is defined in the same way for all groups.

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 [41]. We believe it is more realistic to 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, among 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 [14].

In this article, we detail the design of a system architecture for implementing fairness in a traditional e-commerce recommender system, i.e., where users arrive and are presented with a ranked set of items from a very large inventory that addresses both of these limitations.[1] Our architecture, **Social Choice for Recommendation Under Fairness–Dynamic (SCRUF-D)**, starts from the assumption that multiple fairness concerns should be active at any one time and that these fairness concerns can be relatively unrestricted in form. Second, we build the framework to be *dynamic* in that decisions are always made in the context of historical choices and results. To provide maximum flexibility at the point of recommendation generation, we implement these ideas using a re-ranking design in which fairness considerations are applied to the recommender system output for each user.

The primary contributions in this article are the following:

(1) We formulate the variety of fairness concerns as different agents in the SCRUF-D agent-based architecture. Each of these agents is both able to evaluate a set of recommendations (ranked lists) in terms of fairness and modify a given list (re-rank) to increase fairness according to the agents' concern. The agent-oriented framework allows for multiple fairness concerns and for their definitions to be heterogeneous and independent of each other.

(2) We formulate fairness-aware re-ranking in terms of social choice mechanisms through which fairness agents and the recommender system interact. Specifically, there is an allocation phase where one or more fairness agents are allocated to an arriving user and then the interventions are combined into a delivered ranking using a choice mechanism.

(3) We conduct experiments showing the effectiveness of this technique in allowing multiple fairness concerns to be represented and applied.

(4) We explore a variety of allocation and choice mechanisms from the social choice literature and demonstrate their characteristics and their ability to balance both optimality and fairness according to heterogeneous definitions.

1.1 Motivating Application Setting: Kiva Microloans

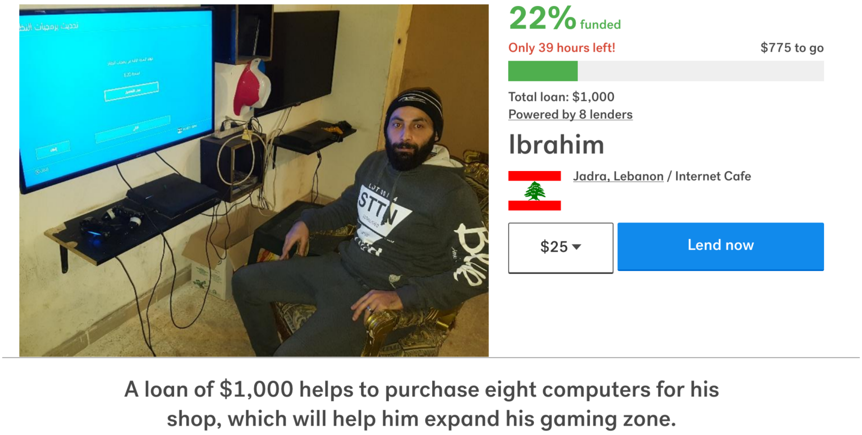
Our research in fairness examines concepts inspired by the application context of Kiva Microloans, a non-profit organization that offers a platform (Kiva.org) for crowd-sourcing the funding of microloans, mostly in the developing world. Kiva's users (lenders) choose among the loan opportunities offered on the platform; microloans from multiple lenders that are aggregated and distributed through third party non-governmental organizations around the world. Figure 1 shows an example of a listing for a loan that a user might choose to support.

Kiva Microloans's 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. To a large part, these fairness issues are around ensuring that all of the borrower's whose loans are shown on Kiva have a fair chance to receive funding and counteracting biases that lenders might have about who is "worthy" of support. There are many complexities in international development and a number of different ways that fairness might be formulated. This complexity was a key motivator for us to seek an architecture that could integrate multiple heterogeneous concerns into the recommendation process.[2]

Kiva's platform, and the motivating setting for our article, is a web- and mobile app-based recommendation system. Users arrive at the Kiva.org website and are served an ordered list of loans

[1] Note that portions of this work appeared in workshop form in Burke et al. [11].

[2] The authors have been working with Kiva on a multi-faceted project addressing a number of related issues in fairness-aware recommendation; see also References [12, 49].



22% funded

Only 39 hours left! $775 to go

Total loan: $1,000

Powered by 8 lenders

Ibrahim

Jadra, Lebanon / Internet Cafe

A loan of $1,000 helps to purchase eight computers for his shop, which will help him expand his gaming zone.

Fig. 1. An example loan posting from the Kiva site.

that they may choose to fund. Hence, our system outputs a (small) ranked list of items from a larger inventory. We will use Kiva's platform as an example throughout this article. However, our findings and associated implementations are not specific to this setting.

2 Related Work

In this section, we discuss the broad topic of fairness in recommender systems including the general problem, some specific approaches to increasing fairness, and the relationship of SCRUF-D to other re-ranking systems.

2.1 General Approaches to Recommender System Fairness

Fairness in recommender systems is a broad topic that has received significant attention in recent years due to number, variety, and impact of these systems. As pointed out by Wang et al. [57], recommender systems often allocate scarce social resources including jobs and money, and hence ensuring the fairness of these systems is a critical ethical concern. Two recent surveys of the landscape each review over 150 papers in the area. Wang et al. [57] provide a survey of papers in top conferences that focus on recommendation system fairness. They include a taxonomy of fairness research that includes our views here: multistakeholder and multi-definitional. Indeed, our work here addresses two key directions for future research outlined in the survey, specifically *win-win models for fairness and accuracy* and *fairness in a real system*. Deldjoo et al. [18] provide a survey of the fairness in recommendation systems field including many of the definitions and contexts in which the problem are studied. In particular, the authors note that many works are focused on abstract problem representations and that more work with stakeholders is necessary. Our work directly addresses this call.

2.2 Multisided Fairness in Recommender Systems

There have been a number of efforts that explicitly consider the multisided nature of fairness in recommendation and matching platforms. Patro et al. [44] investigate fairness in two-sided matching platforms where there are both producers and consumers. They note, as we do, that optimizing fairness for only one side of the market can lead to very *unfair* outcomes for the other side of the market. Patro et al. [44] also appeal to the literature on the fair allocation of indivisible goods from the social choice literature [54]. They devise an algorithm that guarantees max-min share fairness of exposure to the producer side of the market and envy-free up to one item to the consumer side of the market [3]. Their work is closest to the allocation phase of SCRUF-D. However, in contrast to our work, they only use exposure on the producer side and relevance on the consumer side as fairness metrics, whereas SCRUF-D is able to capture additional definitions. Also, we note that envy-freeness is only applicable when valuations are shared: a condition not guaranteed in a personalized system. It is possible for a user with unique tastes to receive low utility recommendations and still not prefer another user's recommendation lists. Also, our fairness formulation extends beyond the users receiving recommendations to providers of recommended items and envy-freeness provides no way to compare users who are recieving different types of benefits from a system. In addition, our fairness definitions are dynamic, a case not considered by Patro et al. [44]. Indeed, in a recent survey paper, Patro et al. [45] provide an overview of some of the current work and challenges in fair recommendation, highlighting the focus of our work, multi-sided, multistakeholder, dynamic, and contextual issues, as key challenges.

Like Patro et al. [44], the work of Sühr et al. [53] investigates fairness in two-sided platforms, specifically those like Uber or Lyft where income opportunities are allocated to drivers. However, unlike our work and the work of Patro et al. [44], Sühr et al. [53] take proportionality as their definition of fairness, specifically proportionality with respect to time in a dynamic setting, and ensure that there is a fair distribution of income to the provider side of the platform. Buet-Golfouse and Utyagulov [9] also tackle the problem of multi-sided fair recommendation, arguing that recommender systems are built on sparse data and that the algorithms must take into account fairness. To this end they propose a regularization term that can be incorporated into the recommendation algorithm itself and account for various biases, e.g., exposure and/or selection bias, depending on the particular definition of the regularization term. The work of Buet-Golfouse and Utyagulov [9] is different from ours in that we make use of post-processing/re-ranking, as it allows for more flexibility with respect to the fairness definitions and allows us to treat the recommendation phase as a black-box, thus our system can be layered with any traditional user/item recommendation algorithm. A post-processing approach also has the advantage that the set of fairness concerns and their relative importance can be adjusted on the fly without model retraining.

2.3 Multiple Fairness Definitions and Computational Social Choice

Working with large item sets and multiple definitions of fairness can be computationally challenging. For example, Zehlike et al. [62] propose a system that allows for multiple protected groups on the provider side and their system works online, i.e., not in an offline, batch mode. But it is designed more for a general ranking task and is very computationally inefficient for recommendation. Every new recommendation list to be reranked requires the generation of a binary tree of size $|G|^k$, where G is the number of protected groups and k is the size of the output list.

Freeman et al. [24] investigate what they call **dynamic social choice functions** in settings where a fixed set of agents select a single item to share over a series of timesteps. The work focuses on overall utility to the agents instead of considering the multiple sides of the recommendation interaction. Their problem is fundamentally a voting problem, since all agents share the result, whereas we are focused on personalized recommendation. Their goal is to optimize the Nash Social Welfare of the set of agents (that remains fixed at each timestep) and present four algorithms to find approximately optimal solutions. This work has a similar flavor to classical online learning/weighting experts problems [16] in the sense that the agent preferences remain fixed and the goal is to learn to satisfy them over a series of timesteps. Similarly, Kaya et al. [30] focus on the problem of group recommendation, which is similar to classical social choice setting of multi-winner voting [67]; i.e., a group of k items are recommended as a shared resource to a set of users, and fairness is defined w.r.t. to this group of users *preferences*.[3] Kaya et al. [30] propose methods that are similar to those found in the multi-winner voting literature that attempt to be fair w.r.t. the individual preferences of the group, and these algorithms are meant to aggregate user preferences, not deliver specific recommendations on a per user basis, a key difference with our work.

2.4 Dynamic, Temporal, and Online Learning Methodologies for Fairness

Ge et al. [26] investigate the problem of long-term dynamic fairness in recommendation systems. This work, like ours, highlights the need to ensure that fairness is modeled as a *temporal concept* and not as a static, one off, decision as is often done in batch fairness models [57]. To this end, they propose a framework to ensure fairness of exposure to the producers of items by casting the problem as a constrained Markov Decision Process where the actions are recommendations and the reward function takes into account both utility and exposure. Ge et al. [26] propose a novel actor-critic deep reinforcement learning framework to accomplish this task at the scale of large recommender systems with very large user and item sets. Again, this work fixes definitions of fairness *a priori*, although their learning methodology may serve as inspiration for an online extension of our allocation stage problems in future work. The subject of the long-term impacts of recommender systems is also considered by Akpinar et al. [1] who investigate the effects of various fairness interventions on the whole system, over time, finding similar effects to those of Ge et al. [26]. This reinforces the decisions in SCRUF-D to take into account the long-term dynamic effects of any fairness intervention.

Reinforcement learning and other online learning methodologies have been proposed for various settings of fair recommendation. Zhang and Wang [64] provide a short position paper proposig using reinforcement learning and an underlying Markov Decision Process to learn priorities and feedback online during user interaction. While this is similar to a typical learning to rank system/framework [63] there are some key differences in their proposal, including learning fairness metrics. While Zhang and Wang do not propose a system, we agree that it is an intriguing direction and methodology to incorporate **reinforcement learning (RL)** into fair recommendation frameworks. In the learning to rank space, wherein a recommendation system is attempting to learn the (set of) users' preference function online, as in reinforcement learning, Morik et al. [37] propose a new algorithm that is able to take into account notions of amortized group fairness while still learning user preferences. This is a fundamentally different setting than what we consider in that we are not performing learning to rank and we do not want to fix a set of fairness criteria *a priori* into our recommendation algorithm; rather, we treat the recommendation algorithm itself as an input.

Morik et al. [37] investigate the problem of learning to rank over large item sets while ensuring fairness of merit-based guarantees to groups of item producers. Specifically, they adapt existing methods to ensure that the exposure is *unbiased*, e.g., that it is not subject to rich-get-richer dynamics, and *fairness* defined as exposure being proportional to merit. Both of these goals are built into the regularization of the learner. In essence, the goal is to learn user preferences while ensuring the above two desiderata. In contrast, our work factors out the recommendation methodology and we encapsulate the desired fairness definitions as separate agents rather than embedded in the learning algorithm.

Similarly to the RL settings described above, other recommendation systems contexts include session-based and streaming (sequential) recommendation. Wu et al. [59] investigate the setting of **session-based recommender systems** that are short, memory-less recommendation

[3] Note that in social choice the term *preference* is used to denote a users ranking over items. Traditional user/item recommendation algorithms can be seen as a type of *preference function* as they produce a ranking or score for each item in a set.

experiences with, e.g., non-logged-in users on a website. Like our work, they propose using the recommendation lists delivered over time to give the overall system some memory as to how fair it has been in the past, proposing a new time-decaying notion of exposure fairness. They also employ a post-processing concept for the overall recommendation: As they focus on session-based recommender systems, there is no notion of long-term user engagement. This is an important direction for potential future work, e.g., non-logged-in users are prevalent in many domains. Within the sequential recommender setting, Li et al. [33] propose a system for a (set of) users that incorporates feedback in learning both the preferences of the agent as well as a regularization term in the online learning algorithms to control for a type of interaction fairness per user. They use a deep learning framework work and a knowledge graph over the items to embed the user feedback and interaction and then use this to ensure that fairness of interaction is happening across protected item groups. Their model assumes fairness on a per-user (though dynamic) basis but more importantly that a large knowledge graph of item information is available, where we make no such assumptions and can define fairness per-user, per-item, and across user sets.

2.5 SCRUF-D, Dynamic Re-ranking, and Computational Advertising

In early stages of this project, we presented an alpha version of the SCRUF-D architecture at a non-archival workshop Sonboli et al. [51]. The alpha version of SCRUF-D considered only a small set of fixed lotteries for allocation and only a single fairness agent and computed fairness metrics at fixed intervals rather than in a fully dynamic way. Through extensive work with Kiva to identify additional concerns and fairness definitions [49], we continued to iterate on the system. The version reported in this work incorporates the ability to dynamically compute allocations for each user arrival based on multiple agents and definitions of fairness, a more extensive set of fairness definitions, choice rules, re-rankers, recommendation models, and metrics.

Similarly to SCRUF-D and working with stakeholders, Ferraro et al. [23] propose a novel set of re-rankers for a music recommendation platform after a set of interview studies with users and producers of music on the system. After these studies, they find that gender fairness is a key issue for many artists, with real-world data they propose a re-ranking methodology that attempts to address this imbalance. The overall methods in this article mirrors ours in that a close examination of a real-world system, and interviews with stakeholders, give rise to a particular notion and measurement of fairness [49]. However, rather than a case study, we propose a more robust framework that treats the multistakeholder and contextual definitions of fairness as first–order concerns.

Finally, our recommendation allocation problem has some similarities with those found in computational advertising, where specific messages are matched with users in a personalized way [56, 60]. 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 [20, 61, 65].

3 Example

In this section, we work through a detailed example demonstrating the function of the architecture through several iterations of user arrivals before introducing the full system formally in Section 5.3. The examples of fairness concerns articulated in our case study of Kiva.org arise from extensive interviews conducted with stakeholders at Kiva.org as a part of the larger scope of this project [49]. Indeed, in our research we found that there were many competing definitions of fairness, e.g., proportional parity or minimal levels of exposure, that act on different sides of the market, i.e., the producers, consumers, or both. In what follows, we give a detailed overview of building up the system from these concerns.

The architecture of SCRUF-D views view the entire recommendation ecosystem as a *multi-agent system* [47]. Informally, each fairness concern can be represented by an agent, i.e., a collection of methods that is able to look at a (set of) recommendations and judge, for themselves, if these (potential) recommendations are *fair* according to the definition of that agent. Likewise, users of the system can be viewed as a disjoint set of agents that only care if the provided recommendations conform to their interests, as judged by the user/item recommendation algorithm. Note that, inspired by the multiagent systems literature, we view the output of a traditional recommendation algorithm, which reports a ranking or scoring for a set of items for a given user as estimating the *preferences* or *preference function* of that user. Given this view, we envision a system where, as a user agent arrives, one or more fairness agents are allocated to that user according to a user agent/fairness agent compatibility function. The fairness agent is able to articulate a list that would be *more fair*, i.e., provide a reranking methodology, and then in the aggregation stage we combine the ranked lists from the fairness agent with the ranked list generated by the user agent preferences to come to a final recommendation.

Given this agent-based view of the recommendation ecosystem, we have a multi-agent system where we are solving both (1) classical social choice allocation problem, i.e., allocating fairness agents to arriving users, and (2) an aggregation (voting) problem, where we combine (possibly competing) lists of recommended items [7]. In the following sections, we articulate, in case study form, how one may approach the problem of formalizing multiple fairness concerns as *agents* within our proposed system.

3.1 Agents (Fairness Concerns)

Consider the following set of fairness agents and their associated fairness definitions, evaluations, and intervention/preference (re-ranker/weighting of items). We assume in this example that in all cases the agents' compatibility functions follow the pattern described in Sonboli et al. [52] where the entropy of the user profile relative to the sensitive feature is calculated and users with high entropy are determined to be good targets for fairness-enhancing interventions. Note that in these examples, the definition of a fair outcome is different for each agent, demonstrating the range of fairness definitions that can be incorporated in our system as follows:

f_H: **Health** This agent is concerned with promoting loans to the health sector. Its evaluation function compares the proportion of loans in the database in the health sector against the proportion of health recommendations in the recommendation list history. If the proportion of health-related recommendations is below that in the inventory, then it considers the prior history to be (proportionately) unfair. Its preference function is binary: If the loan is in the health sector, then the score is 1; otherwise, zero.

f_A: **Africa** This agent is concerned with promoting loans to Africa. Its evaluation function, however, is listwise. It counts lists in the recommendation if they have a least one loan recommendation to a country in Africa, and consider a fair outcome one in which every list has at least one such loan. Its preference function will be similarly binary as the f_H agent.

f_G: **Gender Parity** This agent is concerned with promoting gender parity within the recommendation history. If, across the previously generated recommendation lists, then the number of men and women presented is proportional to their prevalence in the database, its evaluation will return 1. However, it is preference function is more complex than those above. If the women are underrepresented in the history, then it will prefer loans to female borrowers, and conversely for men.[4]

[4] Note: At the time of our data gathering efforts, Kiva's borrower database recognized only binary gender categories.

Table 1. Set of Potential Loans

	ϕ_1^s : Region	ϕ_2^s : Gender	ϕ_3^s : Sector	ϕ_4 : Amount
v_1	Africa	Male	Agriculture	\$5,000-\$10,000
v_2	Africa	Female	Health	\$500-\$1,000
v_3	Middle-East	Female	Clothing	\$0-\$500
v_4	Central America	Female	Clothing	\$5,000-\$10,000
v_5	Central America	Female	Health	\$0-\$500
v_6	Middle-East	Female	Clothing	\$0-\$500

f_L**: Large** This agent is concerned with promoting loans with larger total amounts: over \$5,000. Internal Kiva research has shown that such loans are often very productive, because they go to cooperatives and have a larger local impact. However, the same research has shown that Kiva users are less likely to support them, because each contribution has a smaller relative impact.[5] This agent is similar to the f_A agent above in that it seeks to make sure each list has one larger loan.

3.2 Loans (Items to Be Recommended)

Consider the contents of Table 1. For the sake of example, we will assume these loans, characterized by the Region, Gender, Section, and Amount, constitute the set of loans available for recommendation.

3.3 Mechanisms for Allocation and Aggregation

For the sake of exposition, we posit two very simple mechanisms for allocation and choice. We will assume that our allocation mechanism is a single outcome lottery, e.g., a randomized allocation mechanism [8], wherein one agent will be chosen to participate in the choice mechanism, based on a random draw with probabilities based on the historic unfairness and user compatibility as measured by each agent.

We assume that the recommendation lists are of size 3 and the choice mechanism uses a weighted voting/score-based mechanism [7] using a weighted sum of 0.75 on the personalized results for the recommender system and 0.25 on the output of the allocated fairness agent.

3.4 A Simulation of User Arrivals

At time t_1, **User u_1** arrives at the system and the recommendation process is triggered. The user has previously supported small loans only in Central America and Middle East but has lent to a wide variety of sectors and genders.

For the sake of example, we will assume that the agents measure their prior history relative to their objectives as equally unfair at 0.5, except the Gender Parity agent, which starts out at parity and therefore returns a value of 1. However, the compatibility functions for f_A and f_L returns lower scores because of the user's historical pattern of lending. This yields a lottery in which f_G has probability zero, f_A has a low probability, and f_H a higher one. The allocation mechanism chooses randomly, and we will assume that f_H, the health-focused agent, is picked.

The recommender returns the following list of items and predicted ratings $[\{v_6, 0.6\}, \{v_4, 0.5\}, \{v_5, 0.3\}, \{v_3, 0.3\}, \{v_1, 0.0\}, \{v_2, 0.0\}]$. The f_H agent gives a score of 1 to the health-related loans v_2 and v_5 and 0 to all others.

[5] Pradeep Ragothaman, Personal communication.

The choice mechanism combines these scores as described above and returns the final recommendation list $[\{v_5, 0.475\}, \{v_6, 0.45\}, \{v_4, 0.375\}]$. Note that the Health agent has successfully promoted its preferred item to the first position in the list.

For the sake of example, we assume that the agents' evaluation functions are very sensitive. Therefore, when **User $\boldsymbol{u_2}$** arrives, the results of the previous recommendations have caused the evaluations to shift such that the Health f_H and Large f_L agents are now satisfied (note that v_4 is included in u_1's list and it was a large loan), the Gender parity agent f_G is now at 0.9 (note that there is only one male loan in the database) but the Africa agent f_A, which got nothing in u_1's list, considers recent results to be unfair with a score of 0.25. We assume that u_2 is similar to u_1 in profile and therefore compatibility, but f_A has a much worse fairness score than f_G, and therefore a high allocation probability. We will assume f_A is chosen.

Because this user has similar preferences to u_1, they get the same recommendations: $[\{v_6, 0.6\}, \{v_4, 0.5\}, \{v_5, 0.3\}, \{v_3, 0.3\}]$, $\{v_1, 0.0\}, \{v_2, 0.0\}]$. The f_A agents scores the two loans from Africa (v_1 and v_2) at 1 and the others at 0.

So, after randomly breaking the tie between v_1 and v_2, the final recommendation list is $[\{v_6, 0.45\}, \{v_4, 0.375\}, \{v_1, 0.25\}]$.

When **User $\boldsymbol{u_3}$** arrives, all four agents find themselves scoring fairness at 1 over the evaluation window and so no agents are allocated. The results from the recommendation algorithm pass through the choice mechanism unchanged and are delivered to the user.

In this example, we see the interplay between users' compatibility with agents and the computed fairness outcomes to allocate opportunities to pursue fairness among different agents over time.

4 Formalizing Fairness Concerns

A central tenet of our work is that fairness is a contested concept [39]. 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. As shown in the example above, 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.[6] 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.

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 tradeoffs 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. We report on our initial phases of stakeholder consultation with Kiva.org in Smith et al. [49]. Although not in the recommendation domain,

[6] We are currently conducting research to characterize fairness concerns appropriate to Kiva's recommendation applications. See Smith et al. [49] for some initial findings from this work. None of the discussion here is intended to represent design decisions or commitments to particular concerns or their formulation.

another relevant project is the WeBuildAI project [32] 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 [38] who identifies four as follows:

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, as noted above, 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, then 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 (e.g., test scores) against ideas of compensation for disadvantage (e.g., poverty or group discrimination).

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) [10]. 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 concerns arise 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 [40]; see Ekstrand et al. [21] 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

Table 2. Potential Fairness Concerns and Their Logics

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	Ex. right	Provider	Borrowers in sectors with lower funding rates	Exposure of loans in recommendation lists
AllCountry	Individual	Ex. right	Provider	All borrowers	Catalog coverage by country
AccuracyLoss	Group	Ex. 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

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 from maintaining fair treatment across individuals considered separately. Friedler et al. [25] offer a thorough discussion of this topic.

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 2 illustrates a range of different fairness concerns that could be derived from the microlending context and all of which have at least some support from the interview study by Smith et al. [49]. 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.

5 The SCRUF-D Architecture

In this section, we will first provide a high level overview of the system and describe each aspect in detail with formal notation: Table 3 provides a reference to this notation.

Table 3. Notations for Our Formal Description of the SCRUF-D Architecture

Rec. System	$\mathcal{U}(u)$	Users (user).
	$\mathcal{V}(v)$	Items (item).
	$\phi = \langle \phi_1, \ldots \phi_k \rangle$	Item Features.
	$\omega = \langle \omega_1, \ldots \omega_j \rangle$	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.
	$\mathcal{R}_x(\omega, v) \rightarrow \{v, \hat{r}\}$	Recommendation mechanism that takes a user profile ω and a (set of) items v and produces a predicted rating $\hat{r} \in \mathbb{R}_+$.
	$\ell = \langle \{v_1, \hat{r}_1\}, \ldots \{v_m, \hat{r}_m\} \rangle$	Recommendation List as an ordered list of item, predicted rating pairs.
	$sort(\mathcal{R}_x(\omega, \mathcal{V})) \rightarrow \ell$	Recommendation List for user ω sorted by $\hat{r}$.
Fairness Agents	$\mathcal{F} = \{f_1, \ldots, f_d\}$	Set of Fairness Agents.
	$f_i = \{m_i, c_i, \mathcal{R}_i\}$	Fairness agent $i \in \lvert\mathcal{F}\rvert$ 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. Note: The compatibility metric combines preferences on the agent side and those on the user side (inferred from the profile). If these preferences are symmetrical, then we have a one-sided matching problem, but two-sided cases are also possible.
	$\mathcal{R}_i(\omega, v) \rightarrow \{v, \hat{r}\}$	Fairness Agent (Re)-Scoring function.
	$\ell_{\mathcal{F}} = \{\mathcal{R}_1(\omega, \mathcal{V}), \ldots, \mathcal{R}_i(\omega, \mathcal{V})\}$	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.
	$\vec{H} = \langle \beta^1, \ldots, \beta^t \rangle$	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 is a function from a recommendation list ℓ, agent allocation β, and fairness agent recommendation list(s) $\ell_{\mathcal{F}}$ to a combined output 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 lists ℓ, agent recommendation lists $\ell_{\mathcal{F}}$, and choice function output lists ℓ_C, indexed by timestep t.

5.1 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 [2], kidney allocation [4, 35], and ride sharing [19] 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

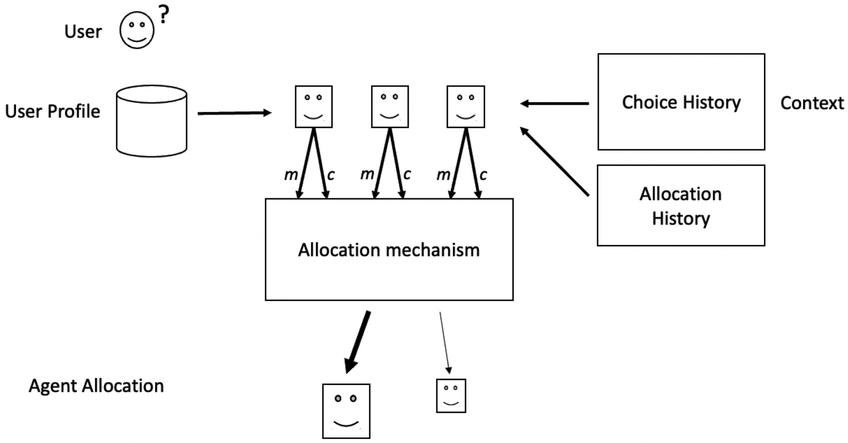


Fig. 2. SCRUF-D Framework/Allocation Phase: Recommendation opportunities are allocated to fairness concerns based on the context.

to any number of users—it is not “used up” in the recommendation interaction.[7] Also, users are not bound to the recommendations provided; in most online platforms including Kiva, there are multiple ways to find items, of which the recommender system is only one.

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*?
(2) How do we *balance* between the allocated agents and the user’s individual preferences?

We assume a recommendation generation process that happens over a number of timesteps 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 step of determining which fairness concerns/agents will be active in responding to a given recommendation opportunity 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 [7].

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 is always allocated and therefore always participates in this phase.

Figure 2 shows the allocation phase of this process. This is an online and dynamic allocation problem where we may consider many factors including the history of agent allocations so far,

[7] 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.

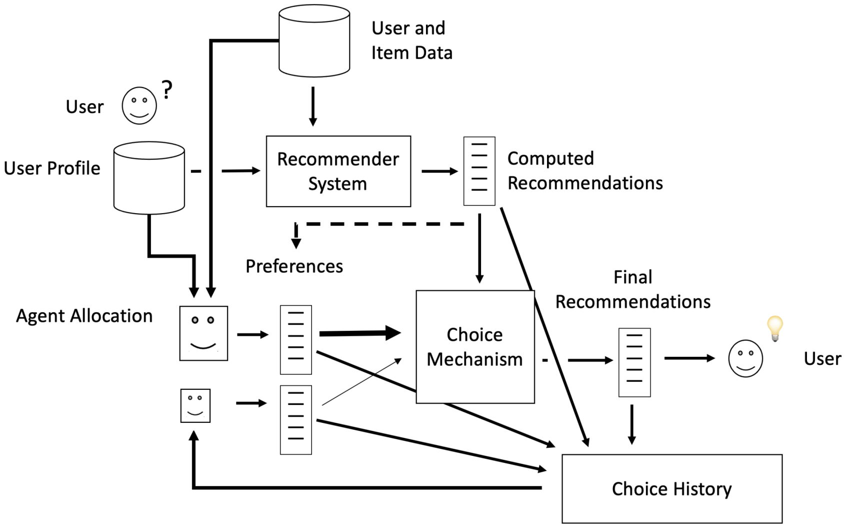


Fig. 3. SCRUF-D Framework/Choice Phase: The preferences derived from the recommender system and the fairness concerns are integrated by the choice mechanism.

the generated lists from past interactions with users, and how fair the set of agents believes this history to be. As described in Section 5.2 below, 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 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 [8].

In the second phase, shown in Figure 3, 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; we call this a recommendation function below. Or, as indicated by the dashed line, these preferences may be scoped only over the items that the recommender system has generated; named a scoring function. 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 the interactions the user has with it, becomes a new addition to the choice history and the process continues for the next user. Because the output of the base recommender system $\mathcal{R}_x$ is an input to this phase, we can think of the entire process as one of fairness-aware re-ranking.

5.2 Fairness Agents

As we have discussed, fairness concerns, derived from stakeholder consultation, are instantiated in the form of fairness agents, each having three capabilities, formalized in Table 3:

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.[8]

5.3 Formal Description

In our formalization of a recommendation system setting we have a set of n users $\mathcal{U} = \{u_1, \ldots u_n\}$ and a set of m items $\mathcal{V} = \{v_1, \ldots, v_m\}$. For each item $v_i \in \mathcal{V}$ we have a k-dimensional feature vector $\phi = \langle \phi_1, \ldots \phi_k \rangle$ 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 $\omega = \langle \omega_1, \ldots \omega_j \rangle$, 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 mechanisms 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, $\ell = \langle \{v_1, \hat{r}_1\}, \ldots \{v_m, \hat{r}_m\} \rangle$, which is generated for user ω by sorting according to $\hat{r}$, i.e., $sort(\mathcal{R}_x(\omega, \mathcal{V})) \rightarrow \ell$. Note that this produces a permutation (ranking) over the set of items for that user, i.e., a recommendation. As a practical 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 are "embodied" as a set of d agents $\mathcal{F} = \{f_1, \ldots, f_d\}$. 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 $i \in |\mathcal{F}|$ is described as a tuple, $f_i = \{m_i, c_i, \mathcal{R}_i\}$ 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, $c_i(\omega) \rightarrow [0, 1]$, 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, $\mathcal{R}_i(\omega, v) \rightarrow \{v, \hat{r}\}$, that gives the fairness agent preferences/ranking over the set of items.

In the allocation phase (Figure 2), we must allocate a set of fairness agents to a recommendation opportunity. Formally, this is an allocation function, $\mathcal{A}(\mathcal{F}, m_{\mathcal{F}}(\vec{L}, \vec{H}), c_{\mathcal{F}}(\omega)) \rightarrow \beta \in \mathbb{R}_+^{|\mathcal{F}|}$ that takes a set of fairness agents $\mathcal{F}$, the agents' fairness metric evaluations $m_{\mathcal{F}}(\vec{L}, \vec{H}) =$

[8] A more complex preference scenario is one in which agents have preferences over entire lists rather than individual items. We plan to consider such preference functions in future work.

$\{m_1(\vec{L},\vec{H}),\ldots,m_d(\vec{L},\vec{H})\}$, 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 every agent voices the most unfairness in the recent history [51], or it could be a more complex function from social choice theory such as the probabilistic serial mechanism [6] 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 [66]. So, although we are assuming different normalized fairness values are comparable, we are only assuming that fairness is comparable with fairness, and not other aspects of the system. We explore options for the allocation function in our empirical experiments below. 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 3), we 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, $C(\ell,\beta,\ell_{\mathcal{F}}) \rightarrow \ell_C$, 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, $\mathcal{R}_i(\omega,v) \rightarrow \{v,\hat{r}\}$, 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 [7]. 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 investigate a range of choice functions in our experiments. 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 timestep t.

6 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 and uniform way. In this section, we discuss a few of the design parameters that may be explored in future work.

6.1 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 that these preferences exhibit nice social choice theoretic properties like consistency or single-peakedness across the different agents [67].

As noted above, fairness 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 [21]. 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 fairness-oriented motivation, it should be clear that our architecture 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. The system has no inherent representation of fairness and would not be able to detect such usage. 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.

6.2 Agent Efficacy

The ability of an agent to address its associated fairness concern is 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 understand 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.

6.3 Mechanism Inputs

Different SCRUF-D 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 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, then 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 [36].

6.4 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 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 [17], 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 [6]. 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 [7].

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 [7].

6.5 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 [43]. Some preliminary discussions of this problem specifically for fairness-aware recommendation appear in Farastu et al. [22].

6.6 Fairness Types

We concentrate in this article 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.

7 Experimental Methodology

As an initial examination of the properties of the SCRUF-D architecture, we conducted a series of experiments with real and simulated data, run on a Python implementation of the SCRUF-D architecture. See associated GitHub repository for the source code.[9] Configuration files, data and Jupyter notebooks for producing the experiments and visualizations are found in a separate repository.[10]

7.1 Datasets

7.1.1 Microlending Data. We used the Microlending 2017 dataset [50], which contains anonymized lending transactions from Kiva.org. The dataset has 2,673 pseudo-items, 4,005 lenders, and 110,371 ratings/lending actions. See References [52] and [50] for a complete description of the dataset.

We considered two loan feature categories, loan size and country, as protected features. Prior work [52] identified loan size as a dimension along which fairness in lending may need to be sought. About 4% of loans had this feature and were considered protected items. We set the fairness target to be 20%. For the second protected feature, we followed Sonboli et al. [52] in identifying the 16 countries whose loans have the lowest rates of funding and labeled these as the protected group for the purposes of geographic fairness. We set the fairness target to be 30%. Compatibility scores were defined using the entropy of a user's ratings versus the protected status of funded loans using the method in Reference [52].

7.1.2 MovieLens Data. We also used the MovieLens 1M dataset [27], which contains user ratings for movies. The dataset has 3,900 movies, 6,040 users, and approximately 1 million ratings. We selected movies with female writers and directors as one protected category and movies with non-English scripts as the other. We set the fairness targets for these to be 12% and 28%, respectively, which mirrors their prelevance in the item catalog.

For both Kiva and MovieLens data, we use a single temporal split of the data with the most recent 20% of each user profile used for evaluation throughout.

7.1.3 Synthetic Data. The purpose of synthetic data in our simulations is to supply realistic recommender system output as input to the SCRUF-D reranker, allowing experimental control of the number of sensitive features, the prevalence of sensitive features among items, and the differing receptiveness of users toward those features.

We create synthetic data via latent factor simulation. That is, we create matrices of latent factors similar to those that would be created through factorization of a user/item rating matrix and then generate sample ratings from these matrices. Let $\hat{U}$ and $\hat{V}$ be the user and item latent factor matrices with k latent factors. We designate the first k_s of the latent factors as corresponding to protected features of items, and the remaining $k - k_s$ factors correspond to other aspects of the items.

As a first step, we generate a vector of real-valued propensities for each user $\Phi_i = \langle \phi_1, \ldots, \phi_{k_s} \rangle$ corresponding to the sensitive features plus additional values for each of the non-sensitive features, drawn from an experimenter-specified normal distribution. Thus, it is possible to adjust the preferences of the user population regarding different sensitive features. The propensities associated with a sensitive feature can also been seen represent the user's compatibility with the respective fairness agent, a value that in a non-synthetic case is derived from the pre-existing user profile as in Reference [52].

[9] https://github.com/that-recsys-lab/scruf_d
[10] https://github.com/that-recsys-lab/scruf_TORS_2024

From Φ_i, we perform an additional generation step to draw a latent factor vector U_i from a normal distribution centered on the propensities. This two-step process avoids having the latent factors tied exactly to the user propensities, which would otherwise make the compatibility of users with agents highly deterministic.

The profiles for items are generated in a similar way except that items have a binary association with their associated sensitive features and so the experimenter input consists of parameters for a multi-variate Bernoulli distribution. Each item's propensity is generated as a binary vector Φ_j using these probabilities. As with users, there is a second step of latent factor generation, in which the elements of an item's latent feature vector V_j are drawn from a normal distribution centered on the item's (binary) propensity for that feature. This two-step procedure allows us to identify an item as possessing a particular feature (particularly protected ones) without the latent factor encoding this exactly.

After $\hat{U}$ and $\hat{V}$ have been generated, we then select m items at random for each user i and compute the product of the $\hat{U}_i$ and $\hat{V}_j$ as the synthetic rating for each user i, item j pair. To simulate the bias for which a fairness solution is sought, we impose a rating penalty γ on ratings generated for items with sensitive attributes. We sort these values and select the top m' as the recommender system output. The sorting/filtering process ensures that the output is biased toward more highly rated items, which is what one would expect in recommender system output. The code for generating synthetic recommender system output using this method is available as open source from GitHub.[11]

For the experiments in this article, we generated 1,500 users (500 users with a high propensity toward the first factor, 500 users with a high propensity for the second protected factor and 500 users with an average propensity for both) and 1,000 items. For each user, we generated 200 sample ratings and used the top 50 as the recommendation lists. Item propensities were set to 0.1, 0.3 for the first two sensitive factors and the other values were randomly set. The standard deviation of the factors was 1.0. Corresponding user propensities for the features were $\mu = 0.5, \sigma = 0.05$ for the both protected factors in the case of the average propensity batch of users, $\mu = 0.1, \sigma = 0.1$ for the low propensity factor and $\mu = 0.9, \sigma = 0.1$ for the high propensity factor in the other two options. The generation parameters were based on proportions seen in real-world datasets including the Microlending dataset described above.

To explore the dynamic response of the system, we created an artificial ordering of the synthetic users with three segments $\langle A, B, C\rangle$, each arriving in sequence. We placed the synthetic users with high compatibility to Agent 2 and low compatibility with Agent 1 in segment A and then reversed this affinity in segment B. Segment C contained users without high compatibility with either agent. These data are referred to as the *Synthetic* dataset in the experiments.

7.2 Fairness Metrics

The agent-specific fairness metric allows each agent to calculate their current state of fairness given the user interactions that have occurred within the evaluation window. As we have noted above, SCRUF-D allows for a wide variety of metrics and makes no assumptions that agents have metrics with similar logic. For the experiments in this study, we have chosen to have uniform fairness metric across agents to more easily assess the impact of varying other platform parameters including the allocation and choice mechanisms.

For each agent defined on fairness concern relative to each dataset, we assign a target exposure value as noted above. That is, an agent with a 20% target exposure will return a value of 1.0 (perfect fairness) if, across all of the recommendation lists in the evaluation window, there is at least an

[11] https://github.com/that-recsys-lab/lafs

average of 20% of items with its associated protected feature. More items do not result in a higher score, but fewer items would yield a value linearly scaled toward zero, as the value when no protected items have been recommended. We have set the targets artificially high to investigate how the system respond to this pressure to include more protected items.

7.3 Mechanisms

As noted above, there is a wide variety of different allocation mechanisms that can be studied. For our purposes in this article, we are exploring mechanisms with widely differing logics to understand the implication of these choices for recommendation outcomes.

- **Least Fair:** The fairness agent with the lowest fairness score m_i is chosen. This simple and commonly used rule ensures that low fairness agents get attention, but it does not take into account the compatibility between a user and an item and so may cause more accuracy loss than others.
- **Lottery:** A lottery is constructed with probabilities proportional to the product of agent unfairness and compatibility: $p(f_i) \propto (1 - m_i) * c_i$, normalized to sum to 1. A single agent is chosen by drawing from this lottery.
- **Weighted:** All agents are allocated to every recommendation opportunity but their weight is determined by the product of their unfairness and compatibility, similarly to the lottery probabilities above: $\beta_i \propto (1 - m_i) * c_i$, normalized.

7.4 Choice/Aggregation Mechanisms (Voting Rules)

We examine four different choice mechanisms. In computational social choice, choice mechanisms are classically understood as integrating the preferences of multiple agents together to form a single societal preference [7].[12]

Rescore: The simplest mechanism is one in which each agent contributes a weighted score for each item and these scores are summed to determine the rank of items. Each fairness agent has a fixed score increment δ that is added to all protected items, weighted by its allocation in the previous phase. This is combined with the scores computed by the recommendation algorithm.

Borda: Under the Borda mechanism [67], ranks are associated with scores and the original scores used to compute those ranks are ignored. The ranks across agents are summed and the result determines the final ranking.

Copeland: The Copeland mechanism calculates a win–loss record for each item considering all item–item pairs in a graph induced by the preferences. Item i scores one point over item j if the majority of allocated agents prefer i to j. We then sum these pairwise match-ups for each item i and order the list of items using these scores [42].

Ranked Pairs: The Ranked Pairs voting rule [55] computes the pairwise majority graph as described for Copeland but orders the resulting ranking by how much a particular item wins by, selecting these to create a complete ranking, skipping a pair if and only if it would induce a cycle in the aggregate ranking.

Each of these choice mechanisms implements a fundamentally different logic for aggregating preferences: score based, ordinal based, consistency based, and pairwise preference [67]. As we show in our results, choice mechanisms yield quite different accuracy/fairness tradeoffs.

[12] Our setting differs from classical social choice in that voting is not anonymous (the recommender system plays a different role from the other agents) and weights and scores are typically employed. Typically, rankings are preferred, because they do not require agent utility to be known or knowable.

While the agent weights in these mechanisms fixed by the allocation mechanism (and normalized to 1), the recommender weight is a parameter, which determines how much the recommender systems results are emphasized relative to the fairness agents. Under different conditions and mechanisms, different recommender weights may be optimal. We refer to this weight throughout as λ.

7.5 Baseline Algorithms

As noted in Section 2, there are very few recommendation algorithms that allow for dynamic reranking with multiple fairness concerns at once. The Multinomial FA*IR method described in [62] is not practical for recommendation because of its time complexity. For these experiments, we use OFair [52] and MultiFR [58].

OFair is a multi-group fairness-aware recommendation reranking method based on the technique of maximum marginal relevance [15] for diversity enhancement in reranking. OFair seeks to enhance fairness for multiple groups by treating each sensitive aspect as a dimension of diversity and greedily building a recommendation list by adding items that enhance its diversity in this respect. OFair adds an additional consideration of personalization by weighting different features according to user compatibility, building on the work of Reference [34] using profile entropy relative to the sensitive features. OFair also has a λ parameter, controlling how much the recommender systems scores are weighted in the reranking process.

Multi-FR is described as a "multi-objective optimization framework for fairness-aware recommendation" [58]. Multi-FR models fairness constraints using a smoothed stochastic ranking policy and optimizes for fairness constraints using multiple gradient descent. The method finds multiple solutions along a theoretical Pareto frontier and chooses the best solution using a least misery strategy. Note that Multi-FR uses a batch-oriented strategy, attempting to address the fairness concerns over the entire set of recommendations at once. Still, Multi-FR is one of the few existing algorithms for fairness-aware recommendation that supports multiple fairness concerns, including provider-side and consumer-side constraints.

One key limitations of Multi-FR is that it represents provider-side fairness only in terms of mutually exclusive provider groups. SCRUF-D has no such limitation and supports intersecting fairness concerns. Because of this difference, in our experiments, we had to create a cross-product of all possible combinations of protected features to capture the multiple features in our datasets. Also, Multi-FR only supports a single type of fairness objective on the provider side: minimizing the difference between actual and ideal exposure of item categories. For the experiments below, target exposures were set as follows: Microlending: loan size/0.20, country/0.30; both features/0.026; MovieLens: women writer or director/0.09, non-English/0.25, both features/0.02. Multi-FR has its own method of balancing accuracy and fairness and so does not have a parameter controlling the balance between fairness and accuracy.

Batch-oriented processing as found in Multi-FR is a fairly common approach for fairness-aware recommendation [21] and it is true that recommendation results are often cached, so processing many users at once is a practical approach. However, it should be noted that a batch approach to fairness-aware recommendation does not guarantee fairness in the recommendations that are delivered. The system can guarantee that a good fairness/accuracy tradeoff is found across the recommendation lists that are processed in a given batch, but, these may not in fact be the recommendations that are delivered to users over any interval. The users compatible with a particular protected group (for example, those interested in foreign language movies) may not happen to show up very often. So, the careful balance between criteria achieved within the batch process may not be realized when the recommendations are delivered in practice: the recommendations that are fairness-enhancing may sit in the cache and never be output. One reason to prefer an on-demand approach to fairness enhancement is that it is responsive to fairness outcomes in the moment. Still, we have included Multi-FR as a comparator to indicate what batch-oriented algorithm can achieve by considering all the recommendations at once.

7.6 Evaluation

We evaluate ranking accuracy using **normalized discounted cumulative gain (nDCG)** and fairness using our (normalized) fair exposure relative to the target proportion set for each protected group. Note that the fairness metric as computed by each agent is different from the overall fairness computed over the experiment, for the simple reason that agents only look back over a fixed window in computing their fairness at each time point. We will use the notation $\bar{m}_i$ to refer to the global fairness for agent i.

To derive a single score representing both the combined fairness of both agents and the disparity between them, we use the $L_{1/2}$-norm, which for our purposes is defined as

$$L_{1/2} = \frac{1}{4}\left(\sum_{\forall i} \sqrt{\bar{m}_i}\right)^2. \tag{1}$$

The factor of $1/4$ is used to give the resulting value the same scale as the original fairness scores. If we consider a simple average $\bar{m}^*$ across all the agents, then the $L_{1/2}$ norm is maximized (and equal to $\bar{m}j$) if all of the fairness values are the same $\bar{m}_i = \bar{m}^*$. A mix of lower and higher values with the same average will give a lower result.

A dynamic way to look at local fairness is to consider **fairness regret** over the course of the experiment. At each timestep, we calculate $1 - m_i$, that is, the difference from perfect fairness as the agent defines it, and then sum these values over the course of the simulation. This is similar to the notion of regret in reinforcement learning but using fairness instead of utility. Cumulative fairness regret G_i for agent i is defined as

$$G_i(s) = \sum_{t=0}^{s} 1 - m_i\left(\vec{L}_t, \vec{H}_t\right). \tag{2}$$

8 Results

For the two datasets, we present results showing the results of adjusting λ, the weight associated with the recommender agent. Lower λ values put more weight on the reranking mechanisms. We then select a single λ value for further analysis of each combination of mechanisms.

8.1 MovieLens Dataset

Figure 4 shows the results for the MovieLens data organized by choice mechanism and showing the results of adjusting λ, the weight associated with the recommender system. In general, as one might expect, as the weight decreases, accuracy drops and fairness increases. We note that the pairwise Copeland and Ranked Pairs mechanisms prove difficult to tune when there is only a single agent being allocated. There is only a single value for all cases when $\lambda < 1.0$. This is because these mechanisms depend only on rankings and so as long as the recommender outweighs the allocated agent, it wins all of the pairwise comparisons and when it does not, it loses all of them.

As was also seen in Reference [13], there exist some "win-win" regions with the Rescore mechanisms in conjunction with Weighted allocation. For small decreases in λ, we see both fairness and accuracy increase at the same time, indicating that some reranking is actually beneficial to accuracy, even as measured in this off-line setting. In addition, the shallow slope of some of these curves

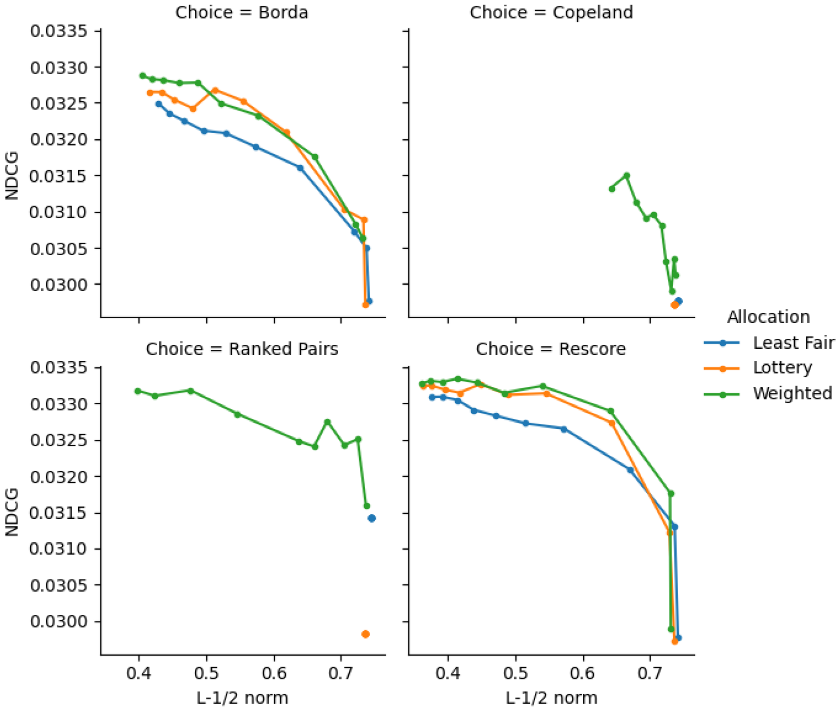


Fig. 4. Accuracy vs. fairness for the MovieLens dataset at different values of λ.

suggests that implementers can increase fairness quite significantly over the baseline without too much loss of ranking accuracy. The other mechanisms have a steeper accuracy loss.

For the remainder of this discussion, we select the λ values where accuracy loss is less than or equal to 5% and consider what is the best fairness that can be achieved within this constraint. We do not have the ability to tune Multi-FR because of its design, which does Pareto optimization internally, so we report the results from this algorithm as designed.

Table 4 includes all of the findings across the different mechanisms. We include both the $L_{1/2}$ norm and the average in the table. Where these are close in value, the agents are getting similar fairness outcomes. We can see that the *women writers and directors* fairness target is quite a bit more difficult to hit than the *non-English* target. There is a large difference already in the unreranked baseline and this carries through to the rerankers. Many of them are able to achieve and exceed the fairness target for the non-English feature, but none do better than 0.4 for the other feature. One exception is Multi-FR, which overshoots the fairness targets and ends up with very low accuracy. Both Least Fair and Weighted in conjunction with the Ranked Pairs mechanism do well in this respect. However, these mechanisms are not the best with respect to maintaining accuracy.

Figure 5 shows the accuracy vs. fairness results for all of the mechanisms and the baseline algorithms. We see some clustering by choice mechanism, except for the two pairwise algorithms, Copeland and Ranked Pairs. For these algorithms, the Lottery mechanism yields very poor ranking accuracy. For Copeland, this is also true of the Least Fair mechanism. The reason is that the Weighted mechanism is so different here is that it is bringing multiple agents to the choice mechanism in these cases. (The others only promote a single agent to the choice phase.) With these multiple agents in the mix, the recommender is dominant and so the agents are less effective at increasing fairness.

Table 4. Accuracy and Fairness Results for the Movielens Data

Allocation	Choice	λ	nDCG	$\bar{m}_1$	$\bar{m}_2$	$L_{1/2}$	Avg
—	—	1.0	0.0331	0.5116	0.1103	0.2742	0.3109
—	*OFair	1.0	0.0261	0.8035	0.2818	0.5092	0.5426
—	Multi-FR	N/A	0.0076	1.082	1.721	1.402	1.383
Least Fair	Borda	0.41	0.0316	1.0950	0.3051	0.6390	0.7000
	*Copeland	0.11	0.0298	1.2601	0.3594	0.7414	0.8097
	Ranked Pairs	0.11	0.0314	1.2043	0.3963	0.7456	0.8003
	Rescore	0.31	0.0321	1.1532	0.3164	0.6694	0.7348
Lottery	Borda	0.41	0.0321	1.0528	0.2985	0.6181	0.6757
	*Copeland	0.11	0.0297	1.2461	0.3596	0.7361	0.8028
	*Ranked Pairs	0.11	0.0298	1.2439	0.3591	0.7350	0.8015
	Rescore	0.31	0.0327	1.0991	0.3086	0.6431	0.7039
Weighted	Borda	0.31	0.0318	1.0483	0.3617	0.6604	0.7050
	Copeland	0.91	0.0315	1.0053	0.3940	0.6645	0.6997
	Ranked Pairs	0.11	0.0316	1.1853	0.3947	0.7370	0.7900
	Rescore	0.21	0.0318	1.2393	0.3542	0.7296	0.7967

Results were chosen to be the greatest $L_{1/2}$ fairness with nDCG loss no greater than 5% over baseline, except for the mechanisms indicated by * which were unable to hit this target at any setting.

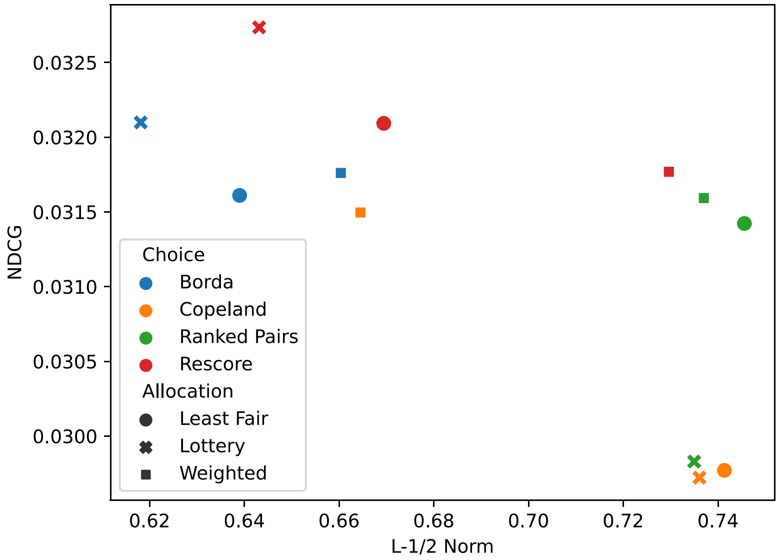


Fig. 5. Comparison of mechanisms on MovieLens data. OFair is omitted as it is far off the chart to the lower left. Multi-FR is far below in terms of accuracy.

Except for the Rescore mechanism, the Lottery allocation data points are all dominated by other points along the Pareto frontier, which would seem to indicate a disadvantage to allocating only a single agent in the allocation phase. Theoretically, Lottery and Weighted allocations are the same in expectation, since the lottery is drawn from the same numerical distribution. However, that is not borne out in the results here. In the case of the Rescore mechanism, these two allocation mechanisms represent very different positions in the tradeoff space.

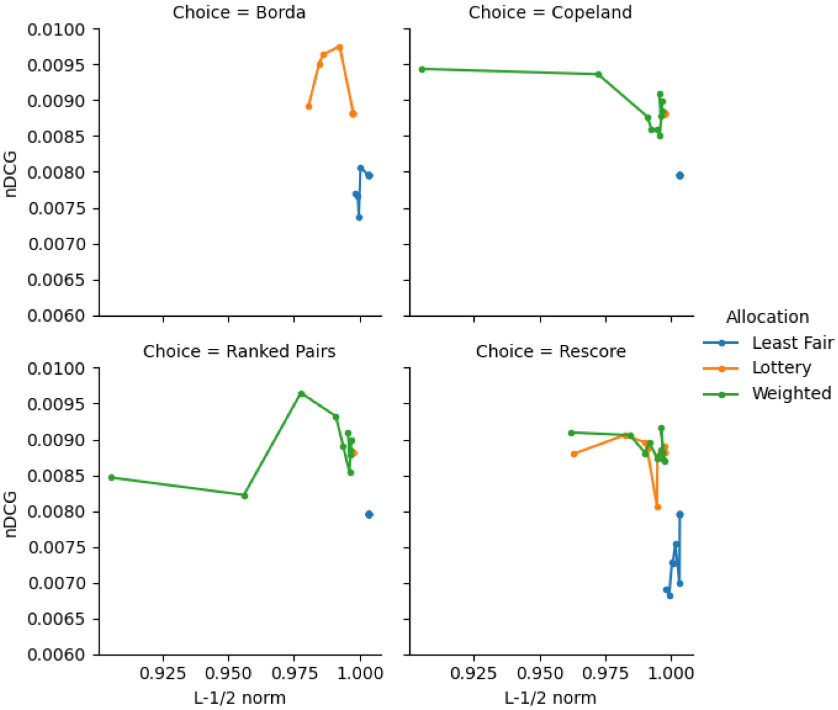


Fig. 6. Accuracy vs. fairness for the Microlending dataset at different values of λ. Value for Weighted + Borda had much lower fairness and were omitted.

Overall, the Rescore and Ranked Pairs mechanisms occupy the dominant positions: Rescore at higher accuracy levels and Ranked Pairs at higher fairness levels. Somewhat surprisingly, the Least Fair allocation stakes out two spots on the Pareto frontier, even though it ignores user compatibility. It is generally lower in accuracy than Lottery or Weighted mechanisms when the same choice mechanism is applied although there are exceptions in the experiments.

Both OFair and Multi-FR are dominated by the SCRUF mechanisms at different points. In the case of Multi-FR, it appears to have inherent limitations on how much accuracy loss it is willing to entertain to increase fairness. We also note that its reranking decisions are made off-line in a batch mechanism and so it is not able to respond dynamically to fairness issues in the moment. OFair also has lower fairness and accuracy. Like the Weighted mechanism, it is trying to address all of the fairness concerns at once in each recommendation list.

8.2 Microlending Dataset

Figure 6 shows the results for the Microlending dataset organized by choice mechanism adjusting λ, the weight associated with the recommender system. The Microlending dataset turns out to be quite different from MovieLens for several reasons. One is that the fairness targets are much easier to achieve. As can be seen in Figure 6, almost all of the mechanisms are able to achieve fairness very close to 1.0. We also see fairly noisy behavior across the different λ values rather than the smoother accuracy/fairness tradeoff seen in the MovieLens data. We believe that this is a side-effect of the easier fairness target: There is not much trading off that the system needs to perform.

Table 5 shows the complete results and we note that the mechanisms are finding identical solutions in many cases, with the same fairness and accuracy values. This suggests that the original

Table 5. Accuracy and Fairness Results for the Microlending Dataset

Allocation	Choice	λ	nDCG	$\bar{m}_1$	$\bar{m}_2$	$L_{1/2}$	Avg
—	—	1.0	0.0074	0.6564	0.0397	0.3481	0.2547
—	OFair	1.0	0.0080	0.6616	0.4539	0.5529	0.5578
—	Multi-FR	N/A	0.0262	0.4472	0.6205	0.5338	0.5303
Least Fair	Borda	0.61	0.0080	1.0077	0.9991	1.0034	1.0034
	Copeland	0.71	0.0080	1.0077	0.9991	1.0034	1.0034
	Ranked Pairs	0.71	0.0080	1.0077	0.9991	1.0034	1.0034
	Rescore	0.41	0.0080	1.0077	0.9991	1.0034	1.0034
Lottery	Borda	0.61	0.0088	1.0011	0.9940	0.9975	0.9975
	Copeland	0.71	0.0088	1.0011	0.9940	0.9975	0.9975
	Ranked Pairs	0.51	0.0088	1.0011	0.9940	0.9975	0.9975
	Rescore	0.41	0.0089	1.0015	0.9940	0.9977	0.9978
Weighted	Borda	0.61	0.0082	0.6548	0.5054	0.5777	0.5801
	Copeland	0.31	0.0090	0.9995	0.9938	0.9966	0.9966
	Ranked Pairs	0.31	0.0090	0.9995	0.9938	0.9966	0.9966
	Rescore	0.41	0.0087	0.9995	0.9955	0.9975	0.9975

Results were chosen to be the best tradeoff for each respective mechanism.

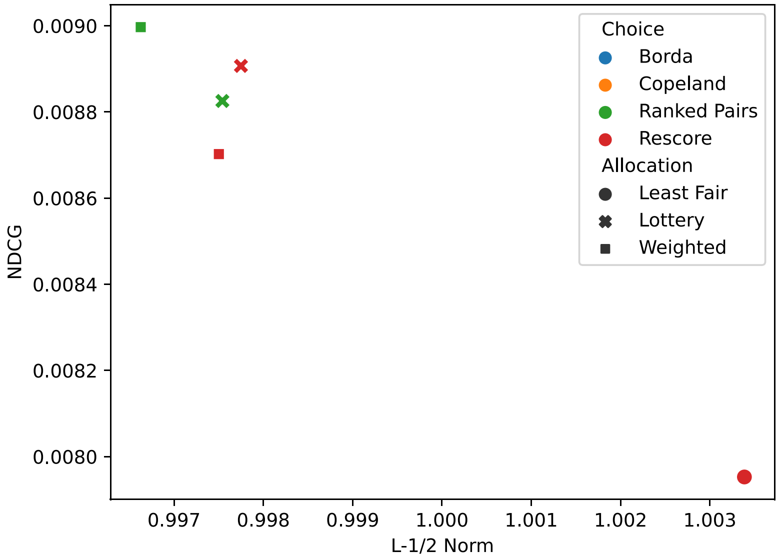


Fig. 7. Comparison of mechanisms on Microlending data. OFair is omitted as it is far off the chart to the lower left. Multi-FR has higher NDCG but is also omitted due to its far lower fairness.

results contain only a limited number of protected items in each list and so there is only so much room for the reranker to alter the results. We believe that part of the reason for this effect is our choice of biased matrix factorization as our base recommendation algorithm. This algorithm is known to suffer from popularity bias and be limited in the range of items that it recommends [29]. In our future work, we will examine alternate base algorithms with better diversity.

The results for the Microlending data are summarized in the scatterplot in Figure 7. There only five points representing the 14 experimental conditions. OFair and Multi-FR are omitted, because they are far from the optimal tradeoff region (although we know from the table that Multi-FR has quite good accuracy). All four of the Least Fair conditions have the same values at the lower right, as do many of the Lottery conditions, except for Rescore.

With these limited data, it is hard to draw too many conclusions. Unlike in the MovieLens case, Least Fair occupies only the most extreme lower accuracy condition in these data. Rescore still seems to be a good strategy and Ranked Pairs with the Weighted allocation is still on the Pareto frontier as it was in the MovieLens case.

8.3 Fairness Dynamics

The compatibility of users with diverse sensitive features turns out to be highly correlated in our real-world datasets, which is perhaps not surprising, but it makes it difficult to explore dynamic scenarios simulating the arrival of disparate types of users at different times. For this reason, we used the *Synthetic* dataset described above in simulated experiments of user arrivals to examine how the balance between agents is achieved over time. With this synthetic dataset, we do not have ground-truth user preferences and so we evaluate recommendation accuracy only relative to the original rankings in the simulated data.

In Figure 8, we look closely at the cumulative fairness regret for the different allocation mechanisms. We keep the choice mechanism fixed (Borda) to isolate the impact of allocation on agent outcomes. Recall that the arrival of users is segmented so that users compatible with Agent 2 arrive as the first 500, followed by another 500 compatible with Agent 1, and then a third 500 user segment without strong compatibility to either agent. Note that the y-axis on these plots has a log scale.

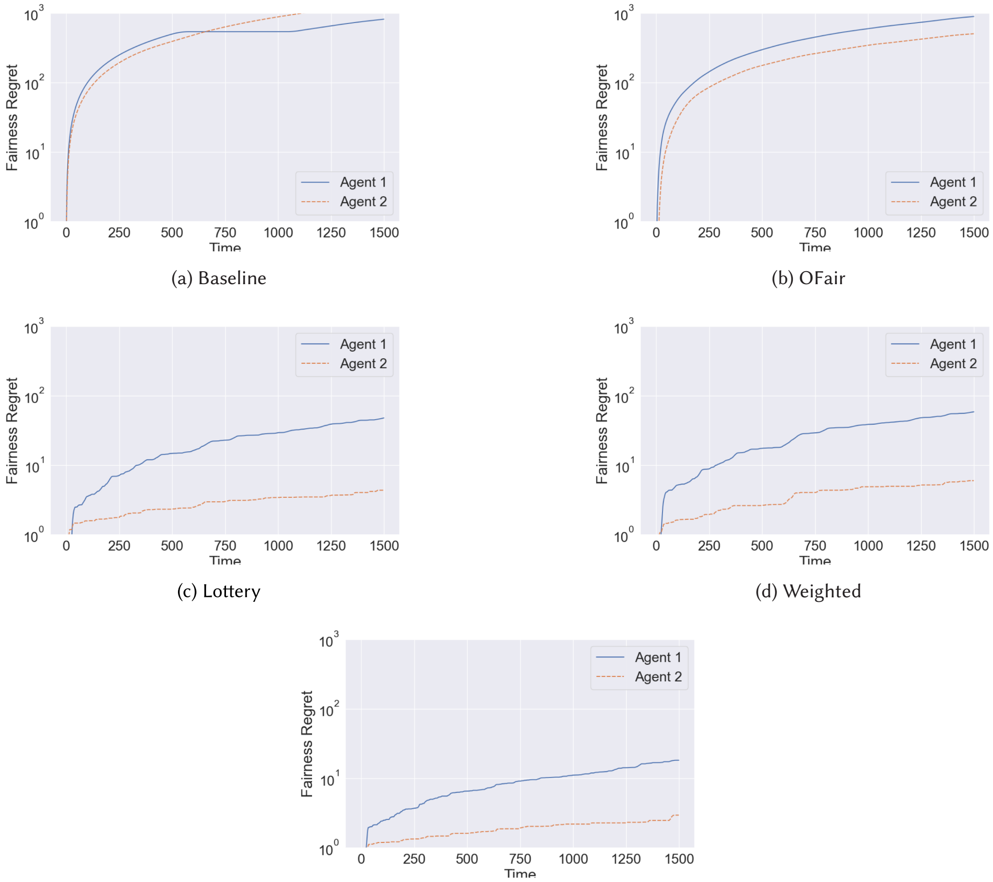
For the Baseline algorithm, without reranking, the impact of the different segments can be seen in the flattening of the Agent 1 curve for users 500–1000. These users are compatible with Agent 1 and already have some of these protected items in their recommendation lists. The regret ends up quite high for both agents.

The OFair algorithm does not fare much better. It is trying to satisfy all of the fairness constraints at once. While its fairness regret is lower, especially for Agent 2, it is still quite high. The other allocation mechanisms fare much better, maintaining 10× or greater improvement in regret over the course of the experiment. The Lottery and Weighted mechanisms are quite similar to each other, with the Weighted mechanism doing slightly better for Agent 2. By not trying (as hard) to satisfy Agent 1 when its compatible users are rare, the system is able to achieve better fairness for both agents.

The Least Fair mechanism seems even better still for both agents. However, there is more to the story. By ignoring compatibility, this mechanism cannot achieve ranking accuracy as high as the others as we have seen in our prior studies. For the synthetic data, we do not have ground-truth user data, so we represent the fairness/accuracy tradeoff by computing nDCG considering the input recommendations to be ground truth. This simulated nDCG measures how much reranking has occurred under each algorithm. These values are shown in Table 6, which shows the loss of accuracy incurred by each algorithm together with the overall fairness values. Differences between the algorithms are small (except for OFair) and we see that the Least Fair mechanism is lowest on accuracy.

9 Conclusion and Future Work

We have introduced the SCRUF-D architecture for integrating multiple fairness concerns into recommendation generation by leveraging social choice. Specifically, the architecture instantiates fairness-aware re-ranking by assigning agents responsible for different fairness concerns, and integrating those agents' perspectives into recommendation through a combination of allocation and



(a) Baseline

(b) OFair

(c) Lottery

(d) Weighted

(e) Least Fair

Fig. 8. Cumulative fairness regret for fairness agents with different allocation mechanisms.

Table 6. Accuracy and Fairness Results for the Synthetic Dataset with Borda as a Choice Mechanism

Allocation	Choice	simulated nDCG	$\bar{m}_1$	$\bar{m}_2$	$L_{1/2}$	Avg
—	—	1.0	1.1170	0.1328	0.5051	0.6249
—	OFair	0.7413	0.6300	0.1127	0.3189	0.3713
Least Fair	Borda	0.9925	1.1970	0.1932	0.5880	0.6951
Lottery	Borda	0.9963	1.1277	0.1860	0.5574	0.6568
Weighted	Borda	0.9963	1.1180	0.1868	0.5547	0.6524

MultiFR is not included, because it incorporates recommendation generation and is not compatible with synthetic data.

preference aggregation. 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.

Our experiments represent a first step at exploring the interactions of different allocation and choice mechanisms at the heart of the SCRUF-D model. We have found that the interaction between mechanisms is quite data and application specific although some general patterns emerge. A simple Rescoring technique is often just as good or better than more complex choice mechanisms. The Least Fair allocation mechanism, which ignores user preferences in allocating agents, is in some cases quite competitive with more sensitive allocations but some times incurs a substantial accuracy loss as we saw in the Microlending dataset.

Our experiments with synthetic data show that the SCRUF-D architecture is capable of balancing among multiple fairness concerns dynamically and in the end, much better fairness results can be achieved by dynamic mechanisms able to respond to the current fairness needs of each agent, as opposed to the static approaches seen in OFair and Multi-FR.

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 formalizing fairness concerns as documented in Reference [49].

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. Our experiments raise the question of the impact of the base recommender and data characteristics on potential outcomes. We will explore additional choices for recommendation algorithms and datasets to explore and confirm the findings here. Likewise, investigating how SCRUF-D performs in domains with dynamic item sets (like Kiva) or where user preferences change rapidly are interesting directions for future work. The architecture itself is able to leverage different base recommender algorithms so any advances for those can be incorporated into SCRUF-D in a straightforward manner. Defining and computing agent compatibility in domains with dynamic inventories presents a challenge, but defining compatibility based on, e.g., item types or protected attributes, as we do in this work, should work well; though more study is needed.

For reasons of space, some key variations on the experiments shown here were not explored. All of the experiments contain only two fairness agents, although this is not a limitation of the architecture. It will be important to see how the results found here extend to larger cohorts of agents. Similarly, we have limited our agent definitions so that all agents have the a similar fairness definition (targeted exposure). We will explore more diverse and heterogenous fairness definitions across agents in future work. We also note that our synthetic data experiments were limited only to examining user arrival sequencing but with a flexible data generation scheme, there are many additional variables to explore including studying how well niche users are served.

We note that in a recommendation context 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 [36], where the fairness concerns represent different classes of rewards and the allocation of agents represents different choices. It would also require extending our overall model to track user responses to recommendations in addition to the recommendation lists themselves.

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, the efficacy problem noted above. 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.

The current architecture does not make any assumptions about the distribution of user characteristics and this can reduce its effectiveness. 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 similarly to Reference [46], 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* [40]: It would not be acceptable for a recommender system to deliver results that reinforced racist or sexist stereotypes at times, even if those results were balanced out at other times 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.

As noted above, 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. One direction would be to formulate the allocation phase of the architecture as an online matching problem, where fairness agents represent one side of the matching and users arrive online on the other side, revealing their compatibility metric. Similarly to work in online ad allocation, each fairness agent might have some budget or capacity that limits the number of users they are matched with, to balance between various fairness concerns. It will 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.

Acknowledgments

Many thanks to Pradeep Ragothaman and our other collaborators at Kiva for sharing their data and many insights into their important work supporting international development through microlending. Thanks also to Joshua Paup and Lalita Suwattee for their assistance in conducting experiments.

References

[1] Nil-Jana Akpinar, Cyrus DiCiccio, Preetam Nandy, and Kinjal Basu. 2022. Long-term dynamics of fairness intervention in connection recommender systems. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, New York, NY, 22–35.

[2] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh. 2015. Online fair division: Analysing a food bank problem. In *Proceedings of the 24th International Joint Conference on AI (IJCAI'15)*. IJCAI, 2540–2546.

[3] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. 2023. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.* 322 (2023), 25 pages.

[4] P. Awasthi and T. Sandholm. 2009. Online stochastic optimization in the large: Application to kidney exchange. In *Proceedings of the 21st International Joint Conference on AI (IJCAI'09)*. IJCAI, 405–411.

[5] Solon Barocas and Andrew D. Selbst. 2016. Big data's disparate impact. *Cal. Law Rev.* 104, 3 (2016), 671. https://doi.org/10.15779/Z38BG31

[6] Anna Bogomolnaia and Hervé Moulin. 2001. A new solution to the random assignment problem. *J. Econ. Theory* 100, 2 (2001), 295–328.

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

[8] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *Am. Econ. Rev.* 103, 2 (2013), 585–623.

[9] Francois Buet-Golfouse and Islam Utyagulov. 2022. Towards fair multi-stakeholder recommender systems. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, New York, NY, 255–265.

[10] Robin Burke. 2017. Multisided Fairness for Recommendation. arxiv:1707.00093 [cs.CY]. Retrieved from https://arxiv.org/abs/1707.00093

[11] Robin Burke, Nicholas Mattei, Vladislav Grozin, Amy Voida, and Nasim Sonboli. 2022. Multi-agent social choice for dynamic fairness-aware recommendation. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, New York, NY, 234–244.

[12] Robin Burke, Pradeep Ragothaman, Nicholas Mattei, Brian Kimmig, Amy Voida, Nasim Sonboli, Anushka Kathait, and Melissa Fabros. 2022. A performance-preserving fairness intervention for adaptive microfinance recommendation. In *Proceedings of the KDD Workshop on Online and Adapting Recommender Systems (OARS'22)*. KDD, 6 pages.

[13] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced neighborhoods for multi-sided fairness in recommendation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*(Proceedings of Machine Learning Research, Vol. 81), Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, New York, NY, 202–214.

[14] 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'20)*. IJCAI, 5 pages.

[15] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 335–336.

[16] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. 1997. How to use expert advice. *J. ACM* 44, 3 (1997), 427–485.

[17] Yuga Cohler, John Lai, David Parkes, and Ariel Procaccia. 2011. Optimal envy-free cake cutting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 25. AAAI, 626–631.

[18] Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. 2024. Fairness in recommender systems: Research landscape and future directions. *User Model. User-Adapt. Interact.* 34, 1 (2024), 59–108.

[19] 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 *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*. AAAI, 1007–1014.

[20] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *Am. Econ. Rev.* 97, 1 (2007), 242–259.

[21] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in information access systems. arxiv:2105.05779 [cs.IR]. Retrieved from https://arxiv.org/abs/2105.05779

[22] Paresha Farastu, Nicholas Mattei, and Robin Burke. 2022. Who pays? Personalization, bossiness and the cost of fairness. arxiv:2209.04043 [cs.IR]. Retrieved from https://arxiv.org/abs/2209.04043

[23] Andres Ferraro, Xavier Serra, and Christine Bauer. 2021. Break the loop: Gender imbalance in music recommenders. In *Proceedings of the Conference on Human Information Interaction and Retrieval*. ACM, New York, NY, 249–254.

[24] Rupert Freeman, Seyed Majid Zahedi, and Vincent Conitzer. 2017. Fair social choice in dynamic settings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17).* International Joint Conferences on Artificial Intelligence, 4580–4587.

[25] 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 (Apr. 2021), 136–143. https://doi.org/10.1145/3433949

[26] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards long-term fairness in recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining.* ACM, New York, NY, 445–453.

[27] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2015), 19.

[28] Ben Hutchinson and Margaret Mitchell. 2019. 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency.* ACM, New York, NY, 49–58.

[29] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. *User Model. User-Adapt. Interact.* 25 (2015), 427–491.

[30] Mesut Kaya, Derek Bridge, and Nava Tintarev. 2020. Ensuring fairness in group recommendations by rank-sensitive balancing of relevance. In *Proceedings of the 14th ACM Conference on Recommender Systems.* ACM, New York, NY, 101–110.

[31] 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]. Retrieved from https://arxiv.org/abs/1711.05144

[32] 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. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (2019), 1–35.

[33] Cheng-Te Li, Cheng Hsu, and Yang Zhang. 2022. FairSR: Fairness-aware sequential recommendation through multi-task learning with preference graph embeddings. *ACM Trans. Intell. Syst. Technol.* 13, 1 (2022), 1–21.

[34] Weiwen Liu and Robin Burke. 2018. Personalizing fairness-aware re-ranking. arxiv:1809.02921 [cs.IR]. Retrieved from https://arxiv.org/abs/1809.02921.

[35] 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'18).* 24 pages.

[36] 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.* KDD, 3224–3233.

[37] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, New York, NY, 429–438.

[38] Hervé Moulin. 2004. *Fair Division and Collective Welfare.* MIT Press.

[39] Deirdre K Mulligan, Joshua A. Kroll, Nitin Kohli, and Richmond Y. Wong. 2019. This thing called fairness: Disciplinary confusion realizing a value in technology. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (2019), 1–36.

[40] Safiya Umoja Noble. 2018. *Algorithms of Oppression: How Search Engines Reinforce Racism.* New York University Press.

[41] Cathy O'Neil. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Broadway Books.

[42] Eric Pacuit. 2019. Voting methods. In *The Stanford Encyclopedia of Philosophy (Fall 2019 ed.)*, Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University.

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

[44] Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of the Web Conference 2020.* ACM, New York, NY, 1194–1204.

[45] Gourab K. Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. 2022. Fair ranking: A critical review, challenges, and future directions. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency.* ACM, New York, NY, 1929–1942.

[46] 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.* ACM, New York, NY, 804–812.

[47] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations.* Cambridge University Press.

[48] Jessie J. Smith, Lex Beattie, and Henriette Cramer. 2023. Scoping fairness objectives and identifying fairness metrics for recommender systems: The practitioners' perspective. In *Proceedings of the ACM Web Conference 2023*. ACM, New York, NY, USA, 3648–3659.

[49] Jessie J. Smith, Anas Buhayh, Anushka Kathait, Pradeep Ragothaman, Nicholas Mattei, Robin Burke, and Amy Voida. 2023. The many faces of fairness: Exploring the institutional logics of multistakeholder microlending recommendation. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. ACM, New York, NY, 1652–1663.

[50] Nasim Sonboli, Amanda Aird, and Robin Burke. 2022. *Microlending 2017 Data Set*. University of Colorado, Boulder. https://doi.org/10.25810/PGJK-RR19

[51] 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]. Retrieved from https://arxiv.org/abs/2009.02590

[52] 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 (UMAP'20)*. Association for Computing Machinery, New York, NY, 239–247. https://doi.org/10.1145/3340631.3394846

[53] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, 3082–3092.

[54] William Thomson. 2011. Fair allocation rules. In *Handbook of Social Choice and Welfare*. Vol. 2. Elsevier, 393–506.

[55] T. Nicolaus Tideman. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare* 4 (1987), 185–206.

[56] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. arxiv:1610.03013 [cs.GT]. Retrieved from https://arxiv.orb/abs/1610.03013

[57] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *ACM Trans. Inf. Syst.* 41, 3 (2023), 1–43.

[58] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. 2022. A multi-objective optimization framework for multi-stakeholder fairness-aware recommendation. *ACM Trans. Inf. Syst.* 41, 2 (2022), 1–29.

[59] Yao Wu, Jian Cao, and Guandong Xu. 2023. FASTER: A dynamic fairness-assurance strategy for session-based recommender systems. *ACM Trans. Inf. Syst.* 42, 1 (2023), 1–26.

[60] 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]. Retrieved from https:///arxiv.org/abs/1206.1754

[61] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: Measurement and analysis. In *Proceedings of the 7th International Workshop on Data Mining for Online Advertising*. ACM, New York, NY, 3.

[62] Meike Zehlike, Tom Sühr, Ricardo Baeza-Yates, Francesco Bonchi, Carlos Castillo, and Sara Hajian. 2022. Fair top-k ranking with multiple protected groups. *Inf. Process. Manage.* 59, 1 (2022), 102707.

[63] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in ranking, Part II: Learning-to-rank and recommender systems. *Comput. Surv.* 55, 6 (2022), 1–41.

[64] Dell Zhang and Jun Wang. 2021. Recommendation fairness: From static to dynamic. arxiv:2109.03150 [cs.IR]. Retrieved from https://arxiv.org/abs/2109.03150

[65] 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, New York, NY, 1077–1086.

[66] 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*. ACM, New York, NY, 1153–1162.

[67] William S. Zwicker. 2016. Introduction to the theory of voting. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 23–56. https://doi.org/10.1017/CBO9781107446984.003

Received 3 March 2023; revised 15 July 2024; accepted 8 August 2024