FAIR&SHARE: Fast and Fair Multi-Criteria Selections

Kathleen Cachel
Worcester Polytechnic Institute
kcachel@wpi.edu

Elke Rundensteiner
Worcester Polytechnic Institute
rundenst@wpi.edu

ABSTRACT

Traditional multi-criteria selection methods are the leading approach for selecting a set of candidates when multiple criteria determine selection relevancy. For instance, hiring platforms combine candidates' proximity, skills, and years of experience to build shortlists for recruiters. While these methods succeed in efficiently selecting candidates, their chosen set may unfairly affect marginalized candidate groups (e.g., race or gender). Bridging the gap between traditional fairness-unaware multi-criteria selection and contemporary fairness interventions, we characterize the open problem of *fair multi-criteria selection*. We design FAIR&SHARE the first efficient fairness-tunable multi-criteria selection method. FAIR&SHARE supports several fair representation notions. The key to FAIR&SHARE is the design of its group-aware utility objective. FAIR&SHARE uses a novel fairness calibration component to provide a user-friendly tuning mechanism for controlling the balance between selection relevancy (utility) and representation fairness. Our fairness-focused selection policy iteratively builds the result set by prioritizing candidates as aiding either the *fair* representation or the *share*-d overall utility goals. We prove the optimality of FAIR&SHARE, meaning that FAIR&SHARE selects the best possible candidates such that the desired fair representation is achieved. Our experimental study demonstrates that FAIR&SHARE achieves the best fairness and utility performance of state-of-the-art alternatives adapted to this new problem while taking a fraction of the time.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Social and professional topics** → **User characteristics**.

KEYWORDS

Algorithmic Fairness, Group Fairness, Fair Candidate Selection.

ACM Reference Format:
Kathleen Cachel and Elke Rundensteiner. 2023. FAIR&SHARE: Fast and Fair Multi-Criteria Selections. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614874

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614874

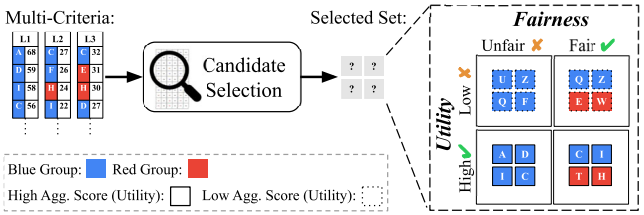


Figure 1: Fair Multi-Criteria Selection. The goal is to select a set of candidates fairly representing groups and having high shared utility (i.e., aggregate scores for selected candidates).

1 INTRODUCTION

Background. Multi-criteria selection sifts through a large volume of potential candidates, each with multiple criteria (features) associated with them, to find a shortlisted set of favorable candidates. After a shortlist has been established, candidates can be prioritized via more resource-intensive means, such as ranking [54] or human subject expertise [58]. Applications from hiring [50, 52], medical care [30, 36], to modern information access (IA) systems [5, 12, 75] follow this pattern. In multi-criteria selection, criteria are modeled as ordered lists of both candidates and each candidate's corresponding score in the given criteria [1, 25, 26, 42, 70, 73]. The blending of multiple criteria is accomplished through an aggregation function [19, 29, 36, 41]. Typically mined [32, 65] or from domain knowledge, this function indicates the relative importance when criteria are combined into a candidate's selection relevancy, i.e., the final aggregate score for each candidate.

Candidate selection is essential for high-stakes decision-making. It is used to determine clinical trial eligibility [64], who is interviewed for a job [52], who profits from product purchases [55], and whose creative content becomes popular [20]. Given their societal influence, selection processes bear the responsibility to not only find relevant candidates but also to ensure their selections are fair and unbiased (illustrated in Figure 1). Specifically, opportunities or resources should not be withheld from protected or marginalized groups, such as race or gender.

State-of-the-Art. Multi-criteria selection has been extensively studied [1, 16, 25, 26, 43, 48, 59, 67, 70, 74, 76]. Conventional approaches focus on efficiently finding the best set of candidates according to an apriori domain-specified aggregation function. Their sole focus has been to maximize set utility, i.e., the total sum of aggregate scores for selected candidates. As seen in Figure 2a, they examine the different lists of criteria directly to select candidates without exhaustively having to combine all the criteria for all candidates first. However, these strategies do not ensure fair representation for groups of candidates who risk being impacted by discriminatory bias.

Recent work has introduced fairness into selection settings, in particular, by developing algorithms [62, 71] and by conducting

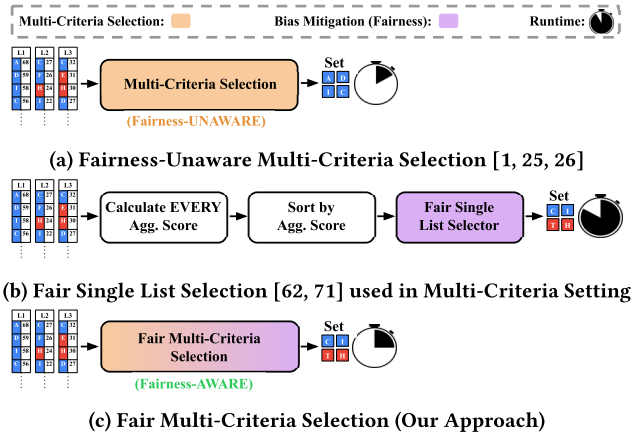
empirical studies [9, 24, 37]. Unfortunately, as shown in Figure 2b, these algorithms are not designed for multi-criteria settings. Instead, they rely on first having a *single* pre-computed score for each candidate, and candidates must be sorted by this score before using the existing fair selection method. For our multi-criteria use case, this many-step solution is prohibitively time-consuming and impractical.

Problem Overview. Our work is the first to consider fair multi-criteria selection, which is an open impactful real-world problem. More concretely, we address fair selection directly from multiple lists representing candidate performance in multiple criteria. Since aggregation functions are specific to their application contexts, we utilize the aggregation function exactly as given by the domain; even in instances, when without fairness intervention, the candidate set is completely biased towards an advantaged group. Further, we exploit core efficiency insights from state-of-the-art (fairness-unaware) multi-criteria selection methods with the aim of bringing high-speed performance to fair selection in the multi-criteria setting. This is a multi-objective problem; with the goals of fairness and utility potentially conflicting. As depicted in Figure 1, the ideal solution ensures a fair representation of candidates while delivering as much utility as possible.

Challenges. Despite its potential for societal impact and practical use, the problem of fair multi-criteria selection remains unaddressed. Challenges to address fair multi-criteria selection include:

- *Dual objectives:* As shown in the quad chart in Figure 1, numerous combinations of fairness and utility exist in selected sets. Conventional approaches fall into the bottom left quadrant - high utility but no fairness guarantees. Simple fairness interventions, such as traversing the lists from top to bottom picking candidates based on their group affiliation, fall into the top right quadrant - fairness but no utility guarantees. In order to be practically useful, fairness-enhanced solutions must also ensure as much utility as possible.
- *Conflicting objectives:* While both fairness and utility are critical goals, they can contradict each other. This is complicated by the fact that the relative importance of each goal may change depending on the task at hand. Solutions must help navigate this tradeoff, even though the severity of this tradeoff is unknown prior to selection and will vary for each candidate pool.
- *Unknown optimal number of candidates per group:* The exact number of candidates to select per group to ensure fairness is often unknown prior to selection. For example, popular fairness notions such as the Rooney-Rule [18, 22] only specify a minimum number of candidates from each group, leaving a portion of the set "unconstrained". Unfortunately, the simple solution of running a fairness-unaware multi-criteria selector [1, 25, 26] once per group, while ignoring candidates not in the current group, is only feasible when the exact number of candidates to select per group is known. Otherwise, the resulting set will provide insufficient utility.

Proposed Method. Addressing the above issues, we propose the multi-criteria fair selection method Fair&Share. The key to Fair&Share is the design of its group-aware utility objective. Addressing the dual and conflicting fairness and utility objectives, we design Fair&Share with an easy-to-use fairness tuning parameter $\delta \in [0, 1]$. While traditional multi-criteria selection methods always pick the utility side of this trade-off, Fair&Share empowers practitioners to seamlessly adjust the balance of fairness and utility in the candidate set. Complete fairness is achieved when $\delta = 0$, and higher δ produces a more minimal fairness intervention, First, the FairCalibrate component translates the user's δ choice and the chosen fairness notion from a variety of supported fairness conceptualizations (such as equal, proportional, Rooney-Rule, and custom groups representations) into one unified model for the FairSelect policy. Next, the FairSelect policy efficiently computes *which candidates* to select, using known constraints per group to inform optimal selections. Specifically, the policy decides whether a candidate is flagged as *fair* and thus helps represent a specific group or as *share* and thus helps to maximize the shared maximal utility objective. We prove the optimality of Fair&Share, meaning, that Fair&Share selects the best possible candidates such that the desired fair representation is achieved.



(a) Fairness-Unaware Multi-Criteria Selection [1, 25, 26]

(b) Fair Single List Selection [62, 71] used in Multi-Criteria Setting

(c) Fair Multi-Criteria Selection (Our Approach)

Figure 2: Related works in Multi-Criteria and Fair Selection.

Advancing beyond prior methods (Figure 2), Fair&-Share ensures a utility-maximizing fair representation of candidates is efficiently selected directly from multiple criteria. We demonstrate the generality and extensibility of Fair&Share's design concepts and algorithmic strategies by applying them to fairness-enhance the seminal multi-criteria selection method - Fagin's algorithm [25].

Contributions. Our contributions are as follows:

- We define the new Fair Multi-Criteria Selection problem, bridging the gap between modern fairness notions and performing selection directly from lists representing multi-criteria.
- Our Fair&Share$\tilde{\text{is}}$ the first solution to this open problem, integrating a user-friendly tuning mechanism balancing of fairness and utility with an efficient fair selection policy.
- Utilizing three synthetic and five real-world datasets, we demonstrate Fair&Share's ability to efficiently find fair sets faster than alternate techniques. We also show the efficacy of Fair&Share for different fairness conceptualizations, along with its versatility in balancing utility and fairness.

2 RELATED WORK

Traditional Multi-Criteria Set Selection. Conventional methods find the best set of candidates according to the provided aggregation function [1, 25, 26, 42, 70, 73]. The main strategy driving the efficiency of these methods is the Threshold Algorithm (TA) procedure which was first introduced in [26] and leveraged in additional methods [1, 42, 73]. In TA, the lists representing the criteria are accessed sequentially (top to bottom), and the aggregate score for each candidate seen is calculated. Then a "threshold" score t is determined by summing the individual scores of the candidates just encountered. Once k candidates have aggregate scores greater than t those candidates are returned, and this set is guaranteed to have maximum utility [26]. TA is why multi-criteria selection is extremely efficient and seldom requires calculating each candidate's aggregate score. The literature also studies alternate problem definitions such as: settings with a partially specified aggregation function [17], probabilistic [53, 61, 67, 69, 74] or XML-based [63] candidate criteria, and privacy-preserving variants [34]. However, to date, no methods ensure fair representation for groups of candidates being selected to avoid discriminatory bias.

Fair Representation in Set Selection. Fairness-enhanced set selection is an active area of research [6, 9, 10, 13, 24, 31, 37, 44–46, 57, 62, 66]. Broadly, this line of work encompasses several problem settings described below. Unlike our work, none of these settings address our problem of representing marginalized groups fairly when selection relevancy combines multiple criteria.

Multi-winner voting produces a subset of the most preferred candidates from either voter rankings or yes/no approvals of candidates. Recent works have designed voting rules and integer programs [6, 10, 46] that enforce constraints on the number of candidates chosen per group. However, they do not select the best-scoring candidates, thus they are not viable for multi-criteria selection.

Clustering partitions an entire set of items into multiple sets (clusters). Many works address the problem of finding a fair clustering [3, 4, 14, 15, 33, 38]. In this problem, items are split into k sets, whereby the sum of the distances from each item to its assigned cluster's center is minimized. This is fundamentally different from our setting since clusters contain similar items, by distance, and clusters have identical demographic group compositions. For a recent survey of fair clustering see [13].

Designing scoring functions is the task of modifying an aggregation function of multi-criteria until a fair set or ranking is produced by this function [2]. Outside the scope of our work, this setting assumes the domain aggregation function can be modified. This requires the function to be re-designed each time the candidate pool changes, e.g. each round of hiring.Instead, our approach utilizes the original aggregation function. Moreover, the computation costs of [2] are polynomial, $n^{2(d-1)}$, with d the number of criteria, whereas candidate set selection is typically at most linear [1, 19, 26, 73].

Candidate Set Selection from Single Criteria is, as mentioned earlier, the line of work most related to our problem. The DivTopK algorithm [62] selects a fair candidate set from a score-sorted list of candidates. While at first sight it can be adapted to our problem, we will show in Section 5 that our proposed approach runs, on average, in 2−7% of DivTopK's runtime. Further, DivTopK [62] does not offer a mechanism for tuning the balance of fairness and utility. This user-friendly feature of our methodology can be directly leveraged by other algorithms, DivTopk included.

Additional variants of fair single criteria selection include: methods for when candidates are selected from a continuous data stream [66], methods for when the group membership of candidates is noisy [44], empirical studies analyzing the effect of the Rooney-Rule a hiring bias intervention, i.e., that the "hired" candidate set include one person from a disadvantaged group [9, 24, 37], metrics for the social notions of diversity and inclusion [45], methods for fair judicial selection [31], and legal analysis of selection-based hiring pipelines [57].

3 PROBLEM FORMULATION

Our setting involves a set $X = \{x_1, x_i, ..., x_n\}$ of n candidates in consideration for selection. By convention [1, 25, 26, 59], X is represented via m **lists** $\mathcal{L} = L_1, ...L_m$ such that every list L_j contains information about all n candidates. The tuple format is $(x_i, s_j(x_i))$, where x_i indicates the candidate and $s_j(x_i)$ is the local score of candidate x_i in that particular criteria list L_j. Lists are ordered by decreasing score, creating different orders among candidates in each criterion. The **aggregate score** of candidate x_i, denoted by $f(x_i)$, is calculated as $f(s_1(x_i), ..., s_m(x_i))$. f is an aggregate scoring function given by the application that we must use exactly as provided. We assume a monotonic function, as is standard [56] and espoused by popular functions such as max, average, and custom linear combinations.

Specific to our setting, candidates have an associated categorical *protected attribute* p (gender, race, or a combination). The set of candidates $x_i \in X$ who have the same value v for attribute p is denoted by *group* $G_{p:v}$. For instance, $G_{race:asian}$ are the candidates of Asian race. The set of all groups associated with X is $\mathcal{G}$.

Our goal of *Fair Multi-Criteria Selection* is to select a set K of k candidates that guarantees all groups are fairly represented while ensuring the selection contains the best possible candidates as measured by their aggregate scores. We also want the selection to be performed as fast as possible. In other words, we seek to efficiently select a set that guarantees a fair representation of candidates while delivering as much utility (the sum of aggregate scores) as possible. What constitutes a fair representation of candidates is a contextual decision dictated by domain knowledge. Meeting this nuance in the nature of fairness, we do not assume one single fairness definition. We support a wide range of popular fairness notions that practitioners can choose from. Table 1 lists the meaning of each of these contemporary fairness definitions we support. The question of how much to intervene in terms of fairness is also contextually specific and best answered by practitioners themselves [21, 60]. Thus, we seek a tunable solution that enables decision-makers to dial-up or dail-down fairness according to the task at hand.

4 OUR METHODOLOGY

4.1 Overview of FAIR&SHARE

We introduce FAIR&SHARE, the first fair multi-criteria selection method. FAIR&SHARE efficiently selects candidate sets guaranteeing maximal utility and satisfying user-provided fair representation criteria. FAIR&SHARE is composed of two unique components. TheFairCalibrate component provides a user-friendly fairness-tuning

Fairness Notion $\mathcal{F}$	Formulation	Intuition
Equal Representation [7, 14]	$\lvert G_{p:i} \cap K\rvert = K\ \lvert\mathcal{G}\rvert^{-1},\ \forall\ G_{p:i} \in \mathcal{G}$	Set contains the same number of candidates per group.
Proportional Representation [23, 51]	$\lvert G_{p:i} \cap K\rvert = \lvert G_{p:i} \cap X\rvert,\ \forall\ G_{p:i} \in \mathcal{G}$	Set contains the same percentage of candidates per group.
Rooney-Rule [18, 22]	$\lvert G_{p:i} \cap K\rvert \geq r,\ \forall\ G_{p:i} \in \mathcal{G}$	Set contains $\geq r$ candidates per group.
Custom (User-provided)	$\lvert G_{p:i} \cap K\rvert \geq g_{p:i},\ \forall\ G_{p:i} \in \mathcal{G}$	Set contains $\geq g_{p:i}$ candidates for $G_{p,i}$ group.

Table 1: Defining fairness in Fair Multi-Criteria Selection. Formulations specify absolute fairness in the respective notion.

mechanism, while the FairSelect policy efficiently selects candidates from $\mathcal{L}$ using our novel *fair* and *share* design.

The rest of this section is organized as follows. We first describe how we reformulate the traditional fairness-unaware multi-criteria utility notion to be fairness-aware. This allows us to guarantee Fair&Share always returns the utility-maximizing set satisfying the given fairness criteria. Then, we describe the architecture of Fair&Share in detail. Finally, we show the extensibility of the Fair&Share design concepts by using them to fairness-enhance the seminal multi-criteria selection method, Fagin's algorithm [25].

4.2 Reformulating Traditional Utility For Group-Fairness Awareness

In the traditional fairness-unaware multi-criteria selection setting [1, 25, 26, 73], selection works by finding the k candidates with the highest aggregate scores from $\mathcal{L}$. In other words, selectors aim to find the set K such that:

$$\arg\max_{\forall K \in \Pi_X} utility(K) = \sum_{\forall x_i \in K} f(x_i), \tag{1}$$

where Π_X is the set of all possible k-sized sets of candidates X.

To do so, the main principle is the Threshold Algorithm (TA) procedure [26] described in Section 2. At every iteration, it calculates a "threshold" score t and keeps the k candidates with the highest aggregate score from those seen-so-far. Then by the monotonicity of $f()$ when $f(x_i) \geq t\ \forall\ x_i \in K$, set K maximizes Eq.1 [26].

However, Equation 1 cannot be used in our setting to ensure that we select the best candidates because it is oblivious to what groups the candidates belong to. Likewise, it ignores candidates' group identities and can only guarantee the best k candidates are selected. Fortunately, we can devise a strategy conserving the TA design principle of using a threshold and exploiting the monotonicity of $f()$ for efficiency in our fairness-aware setting. To do so, we now introduce a new utility maximizing objective, best within-a-group selection, tailored to fairness concerns.

Definition 4.1 (Best Within-a-group Selection). Set K satisfies **best within-a-group selection** if $\forall\ G_{p:j} \in \mathcal{G}$, each candidate x_i with $x_i \in K, x_i \in G_{p:j}$ and each candidate x_h with $x_h \notin K, x_h \in G_{p:j}$: $f(x_i) \geq f(x_h)$ holds.

Next, we establish that a set satisfying best within-a-group selection guarantees that the selected set's utility, Eq. 1, is maximized for the exact representation of groups in the set. For instance, if a set contains 10 candidates from five racial groups and satisfies, best within-a-group selection, then that set has the highest utility of any set with 10 candidates from each group. This is formally stated in Theorem 4.2 below.

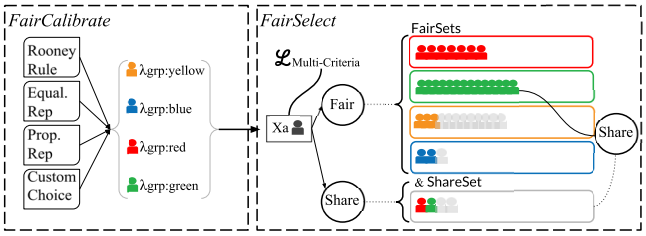


Figure 3: Fair&Share. Given k the size of the returned set, candidates X and their values in protected attribute p, the fairness-utility parameter $\delta \in [0, 1]$, fairness $\mathcal{F}$ and lists $\mathcal{L}$ over set X it returns fair utility-maximizing set K.

Theorem 4.2. *A set K, satisfying best within-a-group selection, has the maximum utility of any set produced from X that has the same representation of groups, i.e., number of candidates per group.*

Proof. Let K be a set satisfying best within-a-group selection (Def. 4.1.) and K' be the set with maximum utility (Eq. 1) such that K' and K have the same number of candidates from each group in $\mathcal{G}$. By contradiction, we show that $K = K'$. Assume that utility(K) < utility(K'). Then the lowest scoring candidate $x_l \in K$ can be swapped with highest scoring candidate $x_h \notin K$ such that $x_l, x_h \in G_{p:v}$, thereby increasing the utility of K. This is a contradiction. By definition of best within-a-group selection, K contains x_h since it is higher scoring than x_l and they belong to the same group. □

Theorem 4.2 allows us to establish a fairness-aware utility maximizing objective for multi-criteria selection. It also serves as the backbone insight for how we (1.) facilitate a tunable solution that balances fairness and utility, and (2.) perform fair selection without knowing the optimal number of candidates per group. For instance, the Rooney-Rule only specifies to select r candidates per group, but $r * |G|$ rarely equals k. The remainder of the set could be filled with candidates from any group. Therefore, each possible set has a different overall utility. Thus, we don't a priori know how many candidates to select per group. Our Fair&Share method embraces this ambiguity and leverages it to our advantage.

4.3 Fair&Share Architecture

As illustrated in Figure 3, Fair&Share features two main components: FairCalibrate and FairSelect. Together they instantiate an efficient fairness-tunable multi-criteria selection method.

First, the FairCalibrate component, detailed in Algorithm 1, translates the chosen variation of supported fairness conceptualizations (e.g. equal, proportional, Rooney-Rule, and custom groups representations) into one unified model for the FairSelect policy. More

Algorithm 1 FAIR&SHARE: FairCalibrate

Input: k size of returned set, X set of candidates and their values in protected attribute p, the fairness-utility parameter $\delta \in [0, 1]$, fairness $\mathcal{F}$, and lists $\mathcal{L}$ over set X.
Output: Set of lower bounds Λ for $\mathcal{G}$.

for each $G_{p:v} \in \mathcal{G}$ **do**
  if $\mathcal{F}$ = Rooney-Rule **then** $Gmin \leftarrow r$
  if $\mathcal{F}$ = custom counts **then** $Gmin \leftarrow$ custom value
  if $\mathcal{F}$ = Prop. Rep. **then** $Gmin \leftarrow ((|G_{p:v}|\backslash|X|) * k)$
  if $\mathcal{F}$ = Equal Rep. **then** $Gmin \leftarrow (k\backslash|\mathcal{G}|)$,
  $\lambda_{p:v} \leftarrow \lfloor (1 - \delta) * Gmin \rfloor$
return $\Lambda = \{\lambda_{G_p:1}, ..., \lambda_{G_p:|\mathcal{G}|}\}$

Algorithm 2 FAIR&SHARE: FairSelect

Input: Λ set of lower bounds per group, lists $\mathcal{L}$ over candidate set X, and k size of returned set.
Output: Set K of k candidates.

Initialize $\mathsf{F} = \{\mathsf{F}_1, ..., \mathsf{F}_{|\mathcal{G}|}\}$ (fair) and S (share) sets.
$t \leftarrow 0$, Initialize `bestpositions` $\leftarrow 0$ //one per $l \in \mathcal{L}$.
while Every $x_j \in \mathsf{F} \cup \mathsf{S}$ has $f(x_j) < t$ and $|\mathsf{F} \cup \mathsf{S}| < k$: **do**
  Access candidates in `bestpositions`.
  for each candidate x_a seen in this iteration **do**
    Directly access x_a in unseen lists to determine $f(x_a)$.
    if $|\mathsf{F}_{G_p:p(x_a)}| < \lambda_{G_p:p(x_a)}$ **then**
      Add x_a to its (fair) set $\mathsf{F}_{G_p:p(x_a)}$
    else
      $x_{min} \leftarrow$ lowest agg. score candidate in $\mathsf{F}_{G_p:p(x_a)}$.
      if $f(x_a) > f(x_{min})$ **then**
        x_a bumps out x_{min} in $\mathsf{F}_{G_p:p(x_a)}$.
        if S is not full **then**
          Add x_{min} to S.
        else if $f(x_{min})$ > min agg. score of S **then**
          x_{min} bumps out cand. with min agg. score.
      if S is not full **then**
        Add x_a to S.
      else if S is full **then**
        if $f(x_a)$ > min agg. score of S **then**
          x_a bumps out cand. with min agg. score.
  Update `bestpostions`, $t = f$(`bestpostions`).
return $K = \mathsf{F} \cup \mathsf{S}$.

importantly, it provides the user with a fairness-utility control mechanism so that they can relax their enforcement of fair representation. This serves to empower practitioners to seamlessly chose the right balance of fairness and utility for their task.

The FairCalibrate step takes as input a fairness notion $\mathcal{F}$ from Table 1 and a *fairness-utility parameter* $\delta \in [0, 1]$. When $\delta = 0$, then $\mathcal{F}$ *is strictly enforced.* As δ increases, $\mathcal{F}$ is relaxed so it is less and less enforced. When $\delta = 1$, then $\mathcal{F}$ *is not enforced.*

As seen in Algorithm 1, FairCalibrate determines how many group members are needed to enforce $\mathcal{F}$ considering how the user has tuned fairness via δ. Here we use lower bounds per group. The sum of the lower bounds rarely sum to k. For instance, in specific notions like the Rooney-Rule and when we relax $\mathcal{F}$ via δ. Therefore, we create an unconstrained portion of the set that we use to select candidates that increase our shared utility objective. Symbol $\lambda_{p:v}$ represents the lower bound cardinality on group $G_{p:v}$, and Λ represents all lower bounds for $\mathcal{G}$. For instances of relaxed fairness, i.e., $\delta < 1$, $Gmin$ is multiplied by $1 - \delta$ to decrease the lower-bound. This is then rounded down to the nearest integer.

The setup of translating fairness notions into a unified representation, along with providing a mechanism to seamlessly tune fairness, is novel for the fair selection literature. Algorithms in applications from clustering to multiwinner voting take as input a user-specified exact number of candidates per group [6, 8, 62] or enforce a single fairness notion strictly [14]. Our FairCalibrate component supports both goals and is enhanced with an easy-to-use mechanism that automatically adjusts the group representation requirement to increase the set utility. Our component could equally be utilized to determine the input for alternate set selection algorithms. In our experimental study in Section 5, we use it along with fair single list selector DivTopK [62].

The FairSelect policy detailed in Algorithm 2 is the second part of FAIR&SHARE. As alluded to earlier (Section 4.2) we use a threshold procedure that accesses candidates in the lists, calculates their score and ensures candidates are the best as long as aggregate scores are greater than the threshold t [1, 26]. Fortunately, we conclude that how we access (and track) $\mathcal{L}$ and compute t does not affect fairness. In our experiments and implementation, we use the bpa2 threshold procedure of [1] as it has demonstrated to have the best efficiency [1] [1]. The best position concept is, for a given $l \in \mathcal{L}$, the greatest ordinal position such that every position between it and the first position has been seen thus far [1]. However, alternate design options [26, 73] could easily be plugged into the FAIR&SHARE architecture.

FAIR&SHARE guarantees fair sets through our selection policy. It chooses candidates based on whether a candidate aids the fair representation objective or aids the shared maximal utility objective. The set returned by FAIR&SHARE is iteratively built through *Fair* sets and a *Share* set. Specifically, there is one fair set per group, and each fair set contains up to λ candidates per group, where λ is determined during the FairCalibrate step. The share set contains $k - \sum$ candidates from any and all groups.

As candidates are initially seen and their aggregate score is calculated, FAIR&SHARE first considers candidates to be assigned to their respective fair set. Candidates are added to the fair set if there are fewer candidates than λ, or if the aggregate score of the just-processed candidate is higher than the aggregate score of any candidate in the fair set. In the case that a candidate "bumps" another candidate, the "bumped" candidate is assessed for inclusion in the share set. Likewise, if an originally processed candidate has a full fair set or its aggregate score is too low to bump another candidate, it is also assessed for inclusion in the share set. Candidates are only added to the share set if the set is unfilled or they have a higher score than the lowest (aggregate) score candidate. Once a candidate is removed from the share set, they are excluded from the result set. Of course, at any time, candidates may have aggregate scores preventing them from inclusion in either set type and, thus from the result set. Once all candidates in the fair and share sets have

[1] In Algorithm 2 we mark the bpa2 threshold procedure pseudocode numbers in brown.

aggregate scores greater than t, Fair&Share returns the union of these sets.

The key invariant of Fair&Share is that at any given iteration, the union of fair and share sets contain the best within-a-group candidates that have been seen thus far. This, as captured in Theorem 4.3, proves that Fair&Share returns the highest utility set for a given (fair) representation of groups.

Theorem 4.3. *If the aggregation function $f()$ is monotonic, then set K returned by the Fair&Share has the highest utility (Eq. 1) of any set with the same fair representation, i.e., the required minimum number of candidates per group.*

Proof. To prove that set K has the highest utility of any set with the same fair representation we need to show Fair&Share satisfies best within-a-group selection. By definition, each fair set $\mathtt{F}_{G_{p:v}}$ returned by Fair&Share is the best of the candidates seen in that group. Likewise, if applicable, the share set $\mathtt{S}$ is the best $k - \sum \Lambda$ candidates not in the fair sets. Since the lists are sorted, if a candidate $x_u \in G_{p:v}$ has not been seen by Fair&Share, then its score in list L_j - $s_j(x_u)$ is at most the lowest local score (from all the lists) of a candidate seen in list L_j. By monotonicity of $f()$, we know the overall score $f(x_u) \leq t_l$, where t_l is the last threshold. Then by the stopping mechanism, t_l is $\leq$ the aggregate scores of candidates in K. Thus, every candidate in K has an aggregate score higher than any candidate not in K belonging to the same group. □

Note that when the desired fairness notion requires each candidate in the selected set to belong to a specific group, i.e., $\sum = k$, then we need not use the share set. Fair&Share would proceed exactly as above without a share set, except when candidates are excluded or removed from a fair set, they directly become obsolete.

4.4 F&S-Fagin: The Fair&Share Fagin Solution

Next, we demonstrate that we can apply the proposed insights underlying Fair&Share into alternate fairness-unaware selection algorithms to enhance them to also select fair sets. Here, we continue with multi-criteria candidate selection as our target problem. Specifically, we leverage the Fair&Share *fair* and *share* design in creating Fair&Share-Fagin, or, in short, F&S-Fagin. Unlike Fagin's algorithm itself [25] and its related versions [17], F&S-Fagin is the first fairness-enhanced Fagin's algorithm guaranteeing the selection of fair and utility-maximizing sets directly from multi-criteria.

The traditional Fagin's algorithm (FA) operates in two steps. First, it sequentially accesses top to bottom $\mathcal{L}$, tracking which candidates have been seen, in what list, and each candidate's score in that list. This step stops once the algorithm has seen k candidates in *all* m criteria lists. At this point, there are potentially many candidates that have only been seen in one or more lists. Thus, in the next step, their aggregate scores are calculated. In this way, the algorithm builds a set Q (where $|Q| >> k$). Lastly, it returns the best k candidates from Q. Since the aggregation function $f()$ is monotonic, the best k candidates are guaranteed to be in Q [25].

We achieve fair multi-criteria selection aligned with the ethos of FA using the design principles presented in Fair&Share. Specifically, F&S-Fagin proceeds by leveraging the FairCalibrate component of Fair&Share. Then to perform fair selection, we introduce the Fagin-aligned selection policy - *FSP*, explained next.

FSP (step 1) sequentially accesses $\mathcal{L}$ to see candidates. Here, F&S-Fagin stops once it has seen k candidates in all $\mathcal{L}$ and the required number of candidates per group as modeled by all $\lambda \in \Lambda$.

FSP (step 2) calculates aggregate scores for each candidate while utilizing *fair* and *share* sets. As in Fair&Share, *fair* sets are "full" once they contain λ candidates from their corresponding group, and the *share* set is full once it has $k - \sum X$ candidates. As aggregate scores are calculated, for candidates seen in *FSP (step 1)*, F&S-Fagin borrows the key ideas from Fair&Share. In brief, candidates are added to their corresponding *fair* set directly or by "bumping" another candidate. Bumped candidates are considered to be added to the *share* set, as well as candidates that are unable to be added to the *fair* set. The invariant property of Fair&Sharethat the union of *fair* and *share* sets contains the best within-a-group candidates seen thus far is also true for F&S-Fagin. Thus, we can easily prove that F&S-Fagin also returns the highest utility set for a given fair representation of groups.

Theorem 4.4. *If the aggregation function $f()$ is monotonic, then set K returned by F&S-Fagin has the highest utility (Eq. 1) of any set with the same fair representation, i.e., the required minimum number of candidates per group.*

Proof. To prove set K has the highest utility of any set with the same fair representation, we must show Fair&Share satisfies best within-a-group selection. Thus, we need to show F&S-Fagin satisfies best within-a-group selection. Proceeding by contradiction, let $x_a \notin K$ be a candidate such that $f(x_a) > f(x_b)$, and $x_a, x_b \in G_{p:v}$ where $x_b \in K$. By the monotonicity of $f()$, the candidate x_a must appear above x_b (i.e., have a higher score) in at least one list. Thus, we have a contradiction since F&S-Fagin would have seen x_a during its first step, computed its aggregate score, and added it into K. □

5 EXPERIMENTS

5.1 Datasets and Metrics

Three synthetic datasets with 10 criteria are created that cover the spectrum of correlation. In each dataset, group A is 20% and group B 80% of the 50,000 total candidates. Namely:

Gauss: is an *independent* dataset with mean = 0 and standard deviation = 1. We set the score for each candidate in each list using this distribution, and then each list is sorted.

Low Corr and ***High Corr***: create datasets of low and high correlation with a list that disadvantages minority group A.

Five real-world datasets we use are described next:

Adult [39]: is comprised of 1994 Census information with five racial groups. Since it exhibits large known racial disparities, we use Adult as a test-bed for the Rooney-Rule.

Bank [47]: holds 41,188 records from a marketing campaign with 7 criteria features. We use the marital status attribute to create "married" and "unmarried" groups.

Credit [68]: dataset contains 29,623 candidates and 10 features related to credit card default for males and females.

IIT [11]: contains the math, physics and chemistry scores for the Joint Entrance Exam (JEE) for 384,977 applicants applying to Indian Institutes of Technology (IIT) for undergraduate admission. Here, the groups are males and females.

Bean [40]: contains 13,611 beans with 17 criteria. We assign three bean types to group A, and the remaining 4 types to group B.

Metrics of efficiency, utility, and fairness utilized are:

Time: reports the average of five runs in seconds.

Utility ratio (UtilR): corresponds to the utility (Eq.1) of the actual set found by the specified technique divided by the utility of the fairness-unaware (maximum utility) size-k set for the same task. It has its maximum (best) value at 1 which means no utility is lost compared to the fairness-unaware utility maximizing set.

Fairness ratio (FairR): captures the fairness of set K. For proportional representation, we use the ratio between the smallest and the largest group selection rates in Eq. 2. For equal representation, the ratio is between the smallest and the largest group proportions of the subset Eq. 3. It has its maximum (best and fairest) value at 1.

$$Fairness\ ratio(prop.) = \frac{\min\{|K \cap G_{p:j}|/|G_{p:j}|\}}{\max\{|K \cap G_{p:i}|/|G_{p:i}|\}}, \forall\ G_{p:j}, G_{p:i} \quad (2)$$

$$Fairness\ ratio(equal) = \frac{\min\{|K \cap G_{p:j}|/|K|\}}{\max\{|K \cap G_{p:i}|/|K|\}}, \forall\ G_{p:j}, G_{p:i} \quad (3)$$

For all datasets, we use the sum of all criteria as the aggregation function $f()$. Our code and experiment implementation is available at https://github.com/KCachel/FairAndShare. [2]

5.2 Compared Methods

We compare against the following:

Max-Util [1]: is the fairness-unaware multi-criteria set selection method known as BPA2. We refer to this approach as "Max-Util" since its sole objective is to maximize utility of the candidate set.

F&S-Fagin: as a baseline, we use our algorithm from Section 4.4.

Group-by-Group (GBG): we construct a baseline that replaces the FairSelect component of FAIR&SHARE with a trivial fairness modification of bpa2 [1]. It runs bpa2 once per group, ignoring candidates not in the group. Then if $\sum \Lambda < k$ it goes over $\mathcal{L}$ again and includes the next seen candidates not already in the set.

Greedy: is a baseline that replaces the FairSelect component of FAIR&SHARE with a fair but greedy selection strategy. It traverses $\mathcal{L}$, top to bottom, and picks the first seen candidates per group to satisfy the fair representation requirement. Then if $\sum \Lambda < k$ it includes the next seen candidates not already in the set. This strategy is simple and fast.

Divtopk [62]: is a *single-list* set selector. It finds the top scoring d candidates per group, from a single utility-sorted list, where d is between a provided lower and upper bound per group. For each fairness notion, we calculate the lower bound needed by using our FAIR&SHARE FairCalibrate component and then set the upper bound to k. This demonstrates the modularity of FAIR&SHARE.

Fa*ir [71]: is a hybrid fair selection and fair ranking algorithm. It creates a ranking of k candidates from a utility-sorted list of $n > k$ candidates such that the "protected group" (disadvantaged relative to equal or proportional representation) is represented near the inputted proportion p. Following [71] we set the significance parameter $\alpha = 0.1$. Note that Fa*ir ranks up to 400 candidates [71, 72], so we cannot use it on the large *IIT* dataset.

[2] All experiments were performed on a Windows 10 machine with 32GB of RAM.

5.3 Experimental Results

We compare FAIR&SHARE with the methods described above with regard to the utility and fairness of the selected set, and time efficiency. We do so for three **fairness concerns: proportional representation, equal representation,** and the **Rooney-Rule.** Recall, that fair multi-criteria selection is a dual-objective problem, with the additional goal of practical efficiency. Thus, the best-performing methods ensure a fair representation exhibited by high FairR scores while maximizing utility exhibited by high UtilR scores, and do so with short runtimes. *Across datasets and fairness concerns, we show FAIR&SHARE consistently achieves the best fairness and utility performance and does so faster than the alternatives.*

5.3.1 Proportional and Equal Representation Results. Tables 2 and 3 present the proportional and equal representation objectives, respectively. Each is broken down into two settings. The first, **strict fairness**, means that proportional or equal representation is exactly satisfied. The second, **relaxed fairness,** means that the proportional or equal presentation fairness intervention is lessened, i.e., relaxed, so that set utility is increased.

Notably, Divtopk, F&S-Fagin, and FAIR&SHARE are the only methods that select utility-maximizing sets with the desired fair representation. This is expected because both F&S-Fagin and FAIR&SHARE find the utility-maxizing fair set by Theorems 4.4 and 4.3, respectively. Additionally, as Divtopk converts multi-criteria selection into single list selection, it also finds fair sets with maximum utility.

The drawback of Divtopk is how slow it is in the multi-criteria setting. This stems from the fact that *every single candidate's* aggregate score has to be calculated to use Divtopk. The observation that Divtopk is affected by the number of candidates is best seen in the *IIT* in Table 3. For strict fairness in Table 3a, *IIT* has 384,977 candidates and Divtopk takes 1551.0427 seconds compared to FAIR&SHARE's 17.1895 seconds. FAIR&SHARE, averaging over all proportional representation results, runs in 19% of F&S-Fagin's runtime, and in 7% of Divtopk's runtime. And for equal representation, it runs in 7% of F&S-Fagin's runtime, and in 2% of Divtopk's runtime. In fact, across all of our experiments, FAIR&SHARE takes at most 27.5% of Divtopk's runtime. This is in the *Gauss* dataset for equal representation in Table 3c, e.g., FAIR&SHARE runs in 25.4241s and Divtopk runs in 92.555.

The next-best performing methods are GBG and Greedy, depending on the metric. Greedy is perhaps the most similar method to FAIR&SHARE in intent, as it aims to enforce that the Λ per-group requirements are met. However, Greedy does so by picking the first candidates encountered in each group, thus Greedy is on par with FAIR&SHARE in terms of time efficiency, but has drastically lower UtilR values. An example of this is the *Gauss* dataset, for equal representation, where Greedy is faster than FAIR&SHARE and comparably fair, but has extremely low utility. Specifically, in Table 3a, Greedy's UtilR value is 0.3320 compared to FAIR&SHARE's 0.9829.

GBG shows that a trivial extension of fairness-unaware bpa2 [1], namely running it once per group, works only when fair representation is strictly enforced, e.g., in Tables 2a or 3. However, when fairness is relaxed, GBG has generally much lower UtilR and less predictable FairR values than FAIR&SHARE, since its utility degrades without an exact count of candidates to select per group.

	Bank $k = 80$			*Credit* $k = 150$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
Max-Util	1.0000	0.6969	0.3793	1.0000	0.5595	0.9520	1.0000	0.9405	24.0050	1.0000	0.7619	0.0494	1.0000	0.0747	0.0636
Fa*ir	1.0000	0.7382	38.0037	0.9990	0.7115	40.2216	1.0000	0.9405	96.0782	1.0000	0.7619	96.7369	0.9018	0.6429	95.8089
GBG	1.0000	0.9784	0.7824	0.9939	0.9872	1.9856	0.9999	1.0000	48.7964	0.9985	1.0000	0.1154	0.8501	1.0000	2.2109
Greedy	0.9526	0.9784	0.0778	0.4309	0.9872	0.2136	0.3320	1.0000	0.1984	0.9984	1.0000	1.5580	0.8371	1.0000	1.1399
Divtopk	1.0000	0.9784	36.0307	0.9939	0.9872	32.9286	0.9999	1.0000	93.4464	0.9985	1.0000	91.7685	0.8501	1.0000	91.6622
F&S-Fagin	1.0000	0.9784	26.8993	0.9939	0.9872	26.2277	0.9999	1.0000	79.2881	0.9985	1.0000	0.7024	0.8501	1.0000	1.5363
FAIR&SHARE	1.0000	0.9784	0.4561	0.9939	0.9872	1.1293	0.9999	1.0000	24.1078	0.9985	1.0000	0.0672	0.8501	1.0000	0.1280

(a) All methods evaluated for strict fairness.

	Bank $k = 80$			*Credit* $k = 150$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
GBG	0.9967	0.9699	0.7918	0.9525	0.9322	1.9648	0.9778	0.9405	48.6802	0.9495	0.8868	1.0234	0.8643	0.7917	0.2753
Greedy	0.9550	0.9290	0.0744	0.4327	0.9852	0.2146	0.3151	0.8864	0.2112	0.9321	0.8868	5.2296	0.8267	0.7917	1.0944
Divtopk	1.0000	0.9199	36.0129	0.9963	0.8825	32.9238	1.0000	0.9405	93.1338	0.9495	0.8868	92.4114	0.8784	0.7917	91.9251
F&S-Fagin	1.0000	0.9199	27.1521	0.9963	0.8825	26.4878	1.0000	0.9405	81.3413	0.9495	0.8868	1.6088	0.8784	0.7917	1.4733
FAIR&SHARE	1.0000	0.9199	0.4679	0.9963	0.8825	1.1363	1.0000	0.9405	24.4989	0.9495	0.8868	0.2345	0.8784	0.7917	0.1308

(b) Methods which model the relaxed fairness of $\delta = 0.05$. In Divtopk we extract Λ from FairCalibrate and use each group's value for d.

	Bank $k = 80$			*Credit* $k = 150$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
GBG	0.9965	0.9784	0.6920	0.9154	0.907	1.8926	0.9446	0.9405	48.0676	0.9546	0.8182	0.9968	0.8915	0.6429	0.2975
Greedy	0.9570	0.8825	0.0818	0.4394	0.9322	0.211	0.332	1.0000	0.194	0.9263	0.8182	5.0397	0.8177	0.6759	1.0356
Divtopk	1.0000	0.8255	36.0321	0.9982	0.771	32.905	1.0000	0.9405	93.8609	0.9546	0.8182	92.3676	0.9018	0.6429	92.6525
F&S-Fagin	1.0000	0.8255	27.0901	0.9982	0.771	26.3769	1.0000	0.9405	82.7384	0.9546	0.8182	1.5698	0.9018	0.6429	1.4728
FAIR&SHARE	1.0000	0.8255	0.4009	0.9982	0.771	1.0606	1.0000	0.9405	24.5055	0.9546	0.8182	0.2237	0.9018	0.6429	0.1198

(c) Methods which model the relaxed fairness of $\delta = 0.10$. In Divtopk we extract Λ from FairCalibrate and use each group's value for d.

Table 2: Results for various levels of proportional representation. Across datasets, strict and relaxed fairness only FAIR&SHARE, F&S-Fagin, and Divtopk find the utility-maximizing fair set. Moreover, FAIR&SHARE, averaging over all equal representation results, finds this set in 7% of F&S-Fagin's runtime (i.e., 7.4 times faster), and in 2% of Divtopk's runtime (i.e., 42.5 times faster).

	Bean $k = 100$			*IIT* $k = 1,000$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
Max-Util	1.0000	0.0000	0.1988	1.0000	0.0661	4.8901	1.0000	0.2658	24.3637	1.0000	0.1905	0.0510	1.0000	0.2987	0.0678
Fa*ir	0.8160	0.6667	14.8363	n/a	n/a	n/a	0.9933	0.6667	96.6851	0.9663	0.6667	96.5867	0.9901	0.6667	96.3776
GBG	0.7643	1.0000	0.9946	0.9679	1.0000	19.4830	0.9829	1.0000	48.1552	0.9414	1.0000	0.2747	0.9746	1.0000	0.1366
Greedy	0.4169	0.6949	0.0980	0.9472	1.0000	29.8275	0.3320	0.2500	0.1882	0.9994	0.2048	1.1127	0.9769	0.8182	0.5037
Divtopk	0.7643	1.0000	10.1369	0.9679	1.0000	1551.0427	0.9829	1.0000	94.2728	0.9414	1.0000	92.3978	0.9746	1.0000	92.3948
F&S-Fagin	0.7643	1.0000	7.7521	0.9679	1.0000	229.2420	0.9829	1.0000	85.1758	0.9414	1.0000	1.6676	0.9746	1.0000	2.3005
FAIR&SHARE	0.7643	1.0000	0.8682	0.9679	1.0000	17.1895	0.9829	1.0000	24.9692	0.9414	1.0000	0.2439	0.9746	1.0000	0.0978

(a) All methods evaluated for strict fairness.

	Bean $k = 100$			*IIT* $k = 1,000$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
GBG	0.7364	0.9231	0.986	0.9689	0.9048	20.2478	0.9578	0.9608	48.1885	0.9991	0.9383	0.1112	0.9706	1.0000	0.5105
Greedy	0.4584	0.9608	0.1264	0.9383	0.938	28.7767	0.3601	0.9608	0.3973	0.995	0.9383	1.2661	0.9614	0.9231	0.5777
Divtopk	0.78	0.8868	10.1209	0.9709	0.9048	1550.554	0.9865	0.8868	94.0706	0.9991	0.9383	91.5996	0.9801	0.8868	92.4624
F&S-Fagin	0.78	0.8868	7.9686	0.9709	0.9048	223.1511	0.9865	0.8868	86.5603	0.9991	0.9383	0.7304	0.9801	0.8868	2.2417
FAIR&SHARE	0.78	0.8868	0.882	0.9709	0.9048	17.0524	0.9865	0.8868	25.5053	0.9991	0.9383	0.0692	0.9801	0.8868	0.0988

(b) Methods which model the relaxed fairness of $\delta = 0.05$. In Divtopk we extract Λ from FairCalibrate and use each group's value for d.

	Bean $k = 100$			*IIT* $k = 1,000$			*Gauss* $k = 100$			*High Corr* $k = 100$			*Low Corr* $k = 100$		
Method	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓	UtilR ↑	FairR ↑	Time (s) ↓
GBG	0.7146	0.8868	0.9704	0.9699	0.8182	19.9121	0.9286	0.9231	47.7748	0.9997	0.878	0.1032	0.9703	1.000	0.5635
Greedy	0.4499	0.9608	0.1254	0.9332	0.8484	27.9631	0.3596	0.8868	0.3965	0.994	0.878	1.2067	0.9626	0.8519	0.5289
Divtopk	0.7903	0.8182	10.1683	0.9738	0.8182	1546.931	0.9887	0.8182	92.555	0.9997	0.878	91.6638	0.9834	0.8182	92.9469
F&S-Fagin	0.7903	0.8182	7.972	0.9738	0.8182	217.3375	0.9887	0.8182	85.9378	0.9997	0.878	0.6708	0.9834	0.8182	2.2219
FAIR&SHARE	0.7903	0.8182	0.8738	0.9738	0.8182	16.0862	0.9887	0.8182	25.4241	0.9997	0.878	0.0594	0.9834	0.8182	0.0898

(c) Methods which model the relaxed fairness of $\delta = 0.10$. In Divtopk we extract Λ from FairCalibrate and use each group's value for d.

Table 3: Results for various levels of equal representation. Across datasets, strict and relaxed fairness only FAIR&SHARE, F&S-Fagin, and Divtopk find the utility-maximizing fair set. Additionally, FAIR&SHARE, averaging over all proportional results, finds this set in 19% of F&S-Fagin's runtime (i.e., 5.2 times faster), and in 7% of Divtopk's runtime (i.e., 13.2 times faster).

	Adult with $r = 5$							*Adult* with $r = 10$						
Method	UtilR ↑	White	Black	Asian	AmIn	Other	Time (s) ↓	UtilR ↑	White	Black	Asian	AmIn	Other	Time (s) ↓
Max-Util	1.0000	91	3	4	0	2	0.2237	1.0000	91	3	4	0	2	0.2237
GbG	0.3964	74	9	7	5	5	3.1031	0.4830	57	12	11	10	10	3.9907
Greedy	0.1242	72	6	8	9	5	0.7560	0.1575	53	14	13	10	10	0.8468
Divtopk	0.9423	80	5	5	5	5	31.1826	0.8525	60	10	10	10	10	31.2126
F&S-Fagin	0.9423	80	5	5	5	5	58.0837	0.8525	60	10	10	10	10	68.6917
Fair&Share	0.9423	80	5	5	5	5	1.7852	0.8525	60	10	10	10	10	1.8830

Table 4: Results for all multi-group methods on the *Adult* dataset with the Rooney-Rule $r = 5$ and $r = 10$ criteria. Fair&Share provides the most efficient selection of the maximum-utility set with the desired fair representation.

This is problematic since the exact number of candidates to select per group is unknown when fairness is slightly relaxed.

Lastly, Fa*ir is the worst performing method because it was neither designed for multi-criteria selection nor explicit fair selection. It exhibits the same time inefficiency as Divtopk. Moreover, its objective is to create a fair top-k ranking. Fairness in ranking is inherently about the ordering of groups, whereas in selection it is about the presence of groups in sets. Thus, when we treat the top-k ranking as the selected set we find that Fa*ir has the lowest FairR values of the fairness-oriented methods. In short, since the fairness concerns are different, the fair ranking approach of Fa*ir does not perform well for multi-criteria set selection.

5.3.2 Rooney-Rule Results. Table 4 presents the results from all methods on the *Adult* dataset with $r = 5$ and $r = 10$. As a result of our prior experiments, we observe the expected behavior that while F&S-Fagin, Divtopk, and Fair&Share all find the utility-maximizing set satisfying the desired fair representation, Fair&Share does so dramatically faster. Specifically, Fair&Share runs in 6% (averaging over $r = 5, 10$) of Divtopk's time.

Interestingly, GbG and Greedy have notably worse performance for the Rooney-Rule compared to equal or proportional representation. The Rooney-Rule is a lighter and more flexible fair representation notion compared to proportional or equal representation since it only asks that r members per group be selected and the remainder of the set is unconstrained. It is extremely popular in practice [27, 49], in part because it has demonstrated significant bias mitigation for such minimal intervention [37]. Unfortunately, the particularly low UtilR values of GbG and Greedy indicate these methods do not perform well for the Rooney-Rule. For instance, when $r = 5$, Fair&Share, GbG and Greedy all ensure ≥ 5 individuals from each group are selected. However, Fair&Share has UtilR of 0.9423, indicating it achieves the fair representation with relatively high preservation of utility compared to the fairness-unaware Max-Util method. In contrast, GbG and Greedy have UtilR values 0.3964 and 0.1242, respectively. Thus, neither are viable solutions when using the Rooney-Rule due to their lack of utility maximization, which is itself often the main driver for the Rooney-Rule fair representation criteria.

6 ETHICAL CONSIDERATIONS

Our work facilitates selecting sets that fairly represent marginalized groups. This research can benefit groups currently disadvantaged by selection systems. While we do not foresee negative outcomes of this work, we are mindful that fairness is a complex sociotechnical concept. We have addressed it from the perspective of prioritizing fair representation, meaning, we are thus limited in producing a set of utmost utility. The fairness-utility trade-off is real in practice regardless of the set selection algorithm. Since traditional multi-criteria selection methods ignore the representation of marginalized groups in their selection, they always pick the utility side of this trade-off. In contrast, algorithms like Fair&Share allow practitioners to balance their own priorities.

7 LIMITATIONS AND FUTURE WORK

We have only begun the study of fair multi-criteria selection and as such, our approach has potential limitations. First, while Fair&Share selects sets guaranteed to maximize utility subject to the corresponding fair representation criteria; we eschew statements that Fair&Share finds fair sets with minimal loss to utility as the tradeoff between fairness and utility is controlled by the dataset (scores of candiates, group composition, etc). That is, the exact loss of utility can only be determined by the data and problem at hand. Future work could study how we might diagnose potential utility loss or introduce ways to bound it in the Fair&Share methodology.

Second, as our work is the initial incorporation of fairness into multi-criteria set selection, we have formulated our fairness-aware problem to closely resemble traditional multi-criteria set selection. Namely, criteria are modeled via m sorted lists, and we assume a monotonic aggregation function. Future work could address how we might relax these assumptions. Third, our approach focuses on three popular notions of group fairness; future work might incorporate additional group fairness notions [28, 51] and or individual fairness [23, 35].

8 CONCLUSION

Our work introduces the Fair Multi-Criteria Selection problem for contexts where selection relevancy combines multiple criteria. For solving this problem, we present Fair&Share. It integrates a user-friendly mechanism for balancing fairness and utility with a novel fair selection policy. Fair&Share selects sets guaranteed to maximize utility, subject to the desired fair representation. We demonstrate that Fair&Share achieves the best fairness and utility performance, and does so faster than existing alternatives from the literature and numerous baseline approaches.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This research was supported in part by the NSF-IIS 2007932 grant.

REFERENCES

[1] Reza Akbarinia, Esther Pacitti, and Patrick Valduriez. 2007. Best position algorithms for top-k queries. In *international conference on Very large data bases (VLDB)*. ACM, 495–506.

[2] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*. 1259–1276.

[3] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable fair clustering. In *International Conference on Machine Learning*. PMLR, 405–413.

[4] Suman K Bera, Deeparnab Chakrabarty, Nicolas J Flores, and Maryam Negahbani. 2019. Fair algorithms for clustering. *arXiv preprint arXiv:1901.02393* (2019).

[5] Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. 2016. CaSMoS: A Framework for Learning Candidate Selection Models over Structured Queries and Documents. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[6] Robert Bredereck, Piotr Faliszewski, Ayumi Igarashi, Martin Lackner, and Piotr Skowron. 2018. Multiwinner elections with diversity constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[7] Kathleen Cachel and Elke A. Rundensteiner. 2022. FINS Auditing Framework: Group Fairness for Subset Selections. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society* (2022).

[8] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. 2016. How to be fair and diverse? *arXiv preprint arXiv:1610.07183* (2016).

[9] L Elisa Celis, Chris Hays, Anay Mehrotra, and Nisheeth K Vishnoi. 2021. The Effect of the Rooney Rule on Implicit Bias in the Long Term. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 678–689.

[10] L Elisa Celis, Lingxiao Huang, and Nisheeth K Vishnoi. 2017. Multiwinner voting with fairness constraints. *arXiv preprint arXiv:1710.10057* (2017).

[11] L Elisa Celis, Anay Mehrotra, and Nisheeth K Vishnoi. 2020. Interventions for ranking in the presence of implicit bias. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 369–380.

[12] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. 2018. Top-K Off-Policy Correction for a REINFORCE Recommender System. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2018).

[13] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. 2021. An overview of fairness in clustering. *IEEE Access* (2021).

[14] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2018. Fair clustering through fairlets. *arXiv preprint arXiv:1802.05733* (2018).

[15] Ashish Chiplunkar, Sagar Kale, and Sivaramakrishnan Natarajan Ramamoorthy. 2020. How to solve fair k-center in massive data models. In *International Conference on Machine Learning*. PMLR, 1877–1886.

[16] Yannis Chronis, Thanh Do, Goetz Graefe, and Keith Peters. 2020. External Merge Sort for Top-K Queries: Eager Input Filtering Guided by Histograms. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2423–2437. https://doi.org/10.1145/3318464.3389729

[17] Paolo Ciaccia and Davide Martinenghi. 2018. FA+ TA< FSA: flexible score aggregation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 57–66.

[18] Brian W Collins. 2007. Tackling unconscious bias in hiring practices: The plight of the Rooney rule. *NYUL Rev.* 82 (2007), 870.

[19] Paul Covington, Jay K. Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems* (2016).

[20] Yashar Deldjoo, Markus Schedl, and Peter Knees. 2021. Content-driven Music Recommendation: Evolution, State of the Art, and Challenges. *ArXiv* abs/2107.11803 (2021).

[21] Wesley Hanwen Deng, Manish Nagireddy, Michelle Seng Ah Lee, Jatinder Singh, Zhiwei Steven Wu, Kenneth Holstein, and Haiyi Zhu. 2022. Exploring How Machine Learning Practitioners (Try To) Use Fairness Toolkits. *2022 ACM Conference on Fairness, Accountability, and Transparency* (2022).

[22] N Jeremi Duru and David Waldstein. 2015. Success and shortfalls in effort to diversify nfl coaching. *The New York Times* (2015).

[23] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[24] Vitalii Emelianov, Nicolas Gast, Krishna P. Gummadi, and Patrick Loiseau. 2020. On Fair Selection in the Presence of Implicit Variance. *Proceedings of the 21st ACM Conference on Economics and Computation* (2020).

[25] Ronald Fagin. 1999. Combining fuzzy information from multiple systems. *Journal of computer and system sciences* 58, 1 (1999), 83–99.

[26] Ronald Fagin. 2002. Combining fuzzy information: an overview. *ACM SIGMOD Record* 31, 2 (2002), 109–118.

[27] Nicholas Fandos. 2020. Senators lead an increasingly diverse nation. their top aides are mostly white. https://www.nytimes.com/2020/08/21/us/politics/top-senate-aides-diversity.html

[28] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016), 3315–3323.

[29] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *International Workshop on Data Mining for Online Advertising*.

[30] Grant D. Huang, Jonca Bull, Kelly Johnston McKee, Elizabeth Mahon, Beth D Harper, and Jamie N. Roberts. 2018. Clinical trials recruitment planning: A proposed framework from the Clinical Trials Transformation Initiative. *Contemporary clinical trials* 66 (2018), 74–79.

[31] Lingxiao Huang, Julia Wei, and L. Elisa Celis. 2020. Towards Just, Fair and Interpretable Methods for Judicial Subset Selection. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020).

[32] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002).

[33] Matthew D. Jones, Huy L. Nguyen, and Thy Nguyen. 2020. Fair k-Centers via Maximum Matching. In *International Conference on Machine Learning*.

[34] Kristján Valur Jónsson, Karl Palmskog, and Ymir Vigfusson. 2012. Secure distributed top-k aggregation. *2012 IEEE International Conference on Communications (ICC)* (2012), 804–809.

[35] Michael Kearns, Aaron Roth, and Zhiwei Steven Wu. 2017. Meritocratic fairness for cross-population selection. In *International Conference on Machine Learning*. PMLR, 1828–1836.

[36] Edward S. Kim, Thomas S. Uldrick, Caroline Schenkel, Suanna Steeby Bruinooge, R. Donald Harvey, Allison Magnuson, Alexander I. Spira, James L. Wade, Mark D. Stewart, Diana Merino Vega, Julia A Beaver, Andrea M. Denicoff, Gwynn Ison, S. Percy Ivy, Suzanne George, Raymond P Perez, Patricia A. Spears, William D. Tap, and Richard L. Schilsky. 2021. Continuing to Broaden Eligibility Criteria to Make Clinical Trials More Representative and Inclusive: ASCO–Friends of Cancer Research Joint Research Statement. *Clinical Cancer Research* 27 (2021), 2394 – 2399.

[37] Jon Kleinberg and Manish Raghavan. 2018. Selection problems in the presence of implicit bias. *arXiv preprint arXiv:1801.03533* (2018).

[38] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. 2019. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*. PMLR, 3458–3467.

[39] Ronny Kohavi and Barry Becker. 1996. Uci machine learning repository: adult data set. *Avaliable: https://archive. ics. uci. edu/ml/machine-learning-databases/adult* (1996).

[40] Murat Koklu and Ilker Ali Ozkan. 2020. Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture* 174 (2020), 105507.

[41] Linchi Kwok, Charlie Adams, and Margret Ann Price. 2011. Factors Influencing Hospitality Recruiters' Hiring Decisions in College Recruiting. *Journal of Human Resources in Hospitality & Tourism* 10 (2011), 372 – 399.

[42] Nikos Mamoulis, Man Lung Yiu, Kit Hung Cheng, and David W Cheung. 2007. Efficient top-k aggregation of ranked inputs. *ACM Transactions on Database Systems (TODS)* 32, 3 (2007), 19–es.

[43] Ankush Mandal, He Jiang, Anshumali Shrivastava, and Vivek Sarkar. 2018. Topkapi: parallel and fast sketches for finding top-K frequent elements. *Advances in Neural Information Processing Systems* 31 (2018).

[44] Anay Mehrotra and L Elisa Celis. 2021. Mitigating Bias in Set Selection with Noisy Protected Attributes. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 237–248.

[45] Margaret Mitchell, Dylan Baker, Nyalleng Moorosi, Emily Denton, Ben Hutchinson, Alex Hanna, Timnit Gebru, and Jamie Morgenstern. 2020. Diversity and inclusion metrics in subset selection. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 117–123.

[46] Farhad Mohsin, Ao Liu, Pin-Yu Chen, Francesca Rossi, and Lirong Xia. 2022. Learning to Design Fair and Private Voting Rules. *J. Artif. Intell. Res.* 75 (2022), 1139–1176.

[47] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[48] HweeHwa Pang, Xuhua Ding, and Baihua Zheng. 2010. Efficient processing of exact top-k queries over disk-resident sorted lists. *The VLDB Journal* 19, 3 (2010), 437–456.

[49] Christina Passariello. 2016. Tech firms borrow football play to increase hiring of women. *Wall Street Journal* 27 (2016).

[50] Andi Peng, Besmira Nushi, Emre Kıcıman, Kori Inkpen Quinn, Siddharth Suri, and Ece Kamar. 2019. What You See Is What You Get? The Impact of Representation Criteria on Human Bias in Hiring. In *AAAI Conference on Human Computation & Crowdsourcing*.

[51] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. *Advances in neural information processing systems* 30 (2017).

[52] M. Raghavan, Solon Barocas, Jon M. Kleinberg, and Karen E. C. Levy. 2019. Mitigating bias in algorithmic hiring: evaluating claims and practices. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (2019).
[53] Christopher Re, Nilesh Dalvi, and Dan Suciu. 2007. Efficient top-k query evaluation on probabilistic data. In *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 886–895.
[54] Stephen E. Robertson. 1997. The probability ranking principle in IR.
[55] David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vasile, and Alexandros Karatzoglou. 2018. RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. *ArXiv* abs/1808.00720 (2018).
[56] Kenneth A. Ross and Yehoshua Sagiv. 1992. Monotonic aggregation in deductive databases. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*.
[57] Jad Salem, Deven R. Desai, and Swati Gupta. 2022. Don't let Ricci v. DeStefano Hold You Back: A Bias-Aware Legal Solution to the Hiring Paradox. *2022 ACM Conference on Fairness, Accountability, and Transparency* (2022).
[58] Clemens Scherer, Stephan Endres, Martin Orban, Stefan Kääb, Steffen Massberg, Alfred Winter, and Matthias Löbe. 2022. Implementation of a clinical trial recruitment support system based on fast healthcare interoperability resources (FHIR) in a cardiology department. *European Heart Journal* 43 (2022).
[59] Anil Shanbhag, Holger Pirk, and Samuel Madden. 2018. Efficient top-k query processing on massively parallel hardware. In *Proceedings of the 2018 International Conference on Management of Data*. 1557–1570.
[60] Jessie J. Smith and Lex Beattie. 2022. RecSys Fairness Metrics: Many to Use But Which One To Choose? *ArXiv* abs/2209.04011 (2022).
[61] Mohamed A Soliman, Ihab F Ilyas, and Kevin Chen-Chuan Chang. 2007. Top-k query processing in uncertain databases. In *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 896–905.
[62] Julia Stoyanovich, Ke Yang, and HV Jagadish. 2018. Online set selection with fairness and diversity constraints. In *Proceedings of the EDBT Conference*.
[63] Martin Theobald, Ralf Schenkel, and Gerhard Weikum. 2005. An efficient and versatile query engine for TopX search. In *Proceedings of the 31st international conference on Very Large Data Bases*. 625–636.
[64] Harald Walach, Catarina Sadaghiani, Cornelia Dehm, and D Bierman. 2005. The therapeutic effect of clinical trials: understanding placebo response rates in clinical trials – A secondary analysis. *BMC Medical Research Methodology* 5 (2005), 26 – 26.
[65] Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Knowledge Discovery and Data Mining*.
[66] Yanhao Wang, Francesco Fabbri, and Michael Mathioudakis. 2021. Fair and Representative Subset Selection from Data Streams. In *Proceedings of the Web Conference 2021*. 1340–1350.
[67] Guoqing Xiao, Kenli Li, Keqin Li, and Xu Zhou. 2015. Efficient top-(k, l) range query processing for uncertain data based on multicore architectures. *Distributed and Parallel Databases* 33, 3 (2015), 381–413.
[68] I-Cheng Yeh and Che-hui Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications* 36, 2 (2009), 2473–2480.
[69] Ke Yi, Feifei Li, George Kollios, and Divesh Srivastava. 2008. Efficient processing of top-k queries in uncertain databases. In *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 1406–1408.
[70] Jing Yuan, Guangzhong Sun, Tao Luo, Defu Lian, and Guoliang Chen. 2012. Efficient processing of top-k queries: selective NRA algorithms. *Journal of Intelligent Information Systems* 39, 3 (2012), 687–710.
[71] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1569–1578.
[72] Meike Zehlike, Tom Sühr, Carlos Castillo, and Ivan Kitanovski. 2019. FairSearch: A Tool For Fairness in Ranked Search Results. *Companion Proceedings of the Web Conference 2020* (2019).
[73] Shile Zhang, Chao Sun, and Zhenying He. 2016. Listmerge: Accelerating top-k aggregation queries over large number of lists. In *International Conference on Database Systems for Advanced Applications*. Springer, 67–81.
[74] Wenjie Zhang, Xuemin Lin, Ying Zhang, Jian Pei, and Wei Wang. 2010. Threshold-based probabilistic top-k dominating queries. *The VLDB Journal* 19, 2 (2010), 283–305.
[75] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed H. Chi. 2019. Recommending what video to watch next: a multitask ranking system. *Proceedings of the 13th ACM Conference on Recommender Systems* (2019).
[76] Vasileios Zois, Vassilis J Tsotras, and Walid A Najjar. 2019. Efficient main-memory top-k selection for multicore architectures. *Proceedings of the VLDB Endowment* 13, 12 (2019).