Fairness of Scoring in Online Job Marketplaces

Sihem Amer-Yahia, Shady Elbassuoni, Ahmad Ghizzawi

Fairness of Scoring in Online Job Marketplaces AUTHORS' VERSION

SHADY ELBASSUONI, Computer Science Department, American University of Beirut
SIHEM AMER-YAHIA, CNRS, University of Grenoble Alpes, France
AHMAD GHIZZAWI, Computer Science Department, American University of Beirut

We study fairness of scoring in online job marketplaces. We focus on group fairness and aim to algorithmically explore how a scoring function, through which individuals are ranked for jobs, treats different demographic groups. Previous work on group-level fairness has focused on the case where groups are pre-defined or where they are defined using a single protected attribute (e.g., whites vs blacks or males vs females). In this manuscript, we argue for the need to examine fairness for groups of people defined with any combination of protected attributes (the-so called subgroup fairness). Existing work also assumes the availability of worker's data (i.e., data transparency) and the scoring function (i.e., process transparency). We relax that assumption in this work and run user studies to assess the effect of different data and process transparency settings on the ability to assess fairness.

To quantify the fairness of a scoring of a group of individuals, we formulate an optimization problem to find a partitioning of those individuals on their protected attributes that exhibits the highest unfairness with respect to the scoring function. The scoring function yields one histogram of score distributions per partition and we rely on Earth Mover's Distance, a measure that is commonly used to compare histograms, to quantify unfairness. Since the number of ways to partition individuals is exponential in the number of their protected attributes, we propose a heuristic algorithm to navigate the space of all possible partitionings to identify the one with the highest unfairness. We evaluate our algorithm using a simulation of a crowdsourcing platform and show that it can effectively quantify unfairness of various scoring functions. We additionally run experiments to assess the applicability of our approach in other less-transparent data and process settings. Finally, we demonstrate the effectiveness of our approach in assessing fairness of scoring in a real dataset crawled from the online job marketplace TaskRabbit.

Additional Key Words and Phrases: algorithmic fairness, scoring, discrimination, demographic disparity, transparency, group fairness, virtual marketplaces

ACM Reference Format:
Shady Elbassuoni, Sihem Amer-Yahia, and Ahmad Ghizzawi . 2021. Fairness of Scoring in Online Job Marketplaces AUTHORS' VERSION. 1, 1 (March 2021), 29 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online job marketplaces are gaining popularity as mediums to hire people to perform certain jobs. These marketplaces include freelancing platforms such as Qapa and MisterTemp' in France, and TaskRabbit and Fiverr in the USA. On these platforms, individuals can find temporary jobs in the physical world (e.g., looking for a plumber), or in the form of virtual "micro-gigs" such as "help

Authors' addresses: Shady Elbassuoni Computer Science Department, American University of Beirut, se58@aub.edu.lb; Sihem Amer-Yahia CNRS, University of Grenoble Alpes, France, sihem.amer-yahia@univ-grenoble-alpes.fr; Ahmad Ghizzawi Computer Science Department, American University of Beirut, ahg05@mail.aub.edu.

XXXX-XXXX/2021/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

with HTML, JavaScript, CSS, and JQuery". Crowdsourcing platforms are also a very popular type of online job marketplaces nowadays. These platforms are fully virtual: workers are hired online and tasks are also completed online. Examples of crowdsourcing platforms are FouleFactory, and Prolific Academic in Europe, and Amazon Mechanical Turk and Figure Eight in the USA.

On virtual marketplaces, either people are ranked or jobs. For instance, on Amazon Mechanical Turk, a worker sees a ranked lists of micro-tasks while on TaskRabbit, an employer sees a ranked list of potential employees. In this work, we are interested in studying fairness in the case where *people are ranked*. A person who needs to hire someone for a job on these platforms can formulate a query and is shown a ranked list of people based on some scoring function. The resulting ranking naturally poses the question of fairness. Algorithmic fairness has recently received great attention from the data mining, information retrieval and machine learning communities (See for instance [1–4]). The most common definition of fairness was introduced in [5, 6] as *demographic disparity*, that is the unfair treatment of a person based on *belonging to a certain group of people*. Groups are defined using protected attributes such as gender, age, ethnicity or location. For instance, in the French Criminal Law (Article 225-1), 23 such attributes are listed as discriminatory.[1] We carry these definitions in our work and define unfairness of scoring in online marketplaces as the unequal treatment of people by a scoring process based on their protected attributes. This definition is inline with what is also commonly referred to as *group unfairness* [7].

Fairness Model. Our goal in this manuscript is quantify unfairness of a scoring of individuals in online job marketplaces. We cast our problem into a mathematical formulation based on the one defined in [7] for fairness in decision making in general. In this formulation, the scoring process takes place through three metric spaces: *the construct space*, which contains necessary but unmeasurable attributes to score individuals such as their grit, punctuality and experience, *the observed space*, which is a set of measurable attributes that approximate those in the construct space such as ratings, number of past jobs and qualification scores, and *the decision space*, which contains the outcome of the scoring process. Our goal is then to quantify unfairness in the decision space, which occurs as a consequence of *structural bias*, i.e., the unequal treatment of groups of people based on their protected attributes by the observation process that maps between the construct space and the observed space. In some cases, discriminatory scoring might be desired, which is known as *positive discrimination* [8] where disadvantaged individuals are favored. Nonetheless, unwarranted unfairness can occur within subgroups. For example, consider a requester on a crowdsourcing platform who is looking for *female* annotators only. This is an example of positive discrimination. The scoring function can however still be unfair with respect to subgroups of females such as older females or black ones. Our framework thus aims to identify unfairness at any subgroup level. However, it stops short of fixing any unveiled unfairness, whether positive or negative, and it is up to the user, the job owner or the platform developer, to decide on the right subsequent action.

Most previous work on group-level fairness have either assumed that groups are pre-defined [4] or that they are defined using a single protected attribute (e.g., whites vs blacks or males vs females) [9]. As mentioned above, in this manuscript we consider groups of individuals defined with any combination of protected attributes (the so-called *subgroup fairness* [3]). The scoring function through which individuals are ranked yields one histogram per demographic group as score distributions. We use the the Earth Mover's Distance (EMD) [10], a measure that is commonly used to compare histograms, to quantify distances between groups. The intuition behind this is that if the distribution of scores is significantly different between different groups, the scoring process does not treat the individuals in these groups equally. For instance, consider two groups

[1]https://www.legifrance.gouv.fr/affichCodeArticle.do?cidTexte=LEGITEXT000006070719&idArticle=LEGIARTI000006417828

only, namely young males and young females. Unfairness can be computed as the EMD between the score distributions of these two groups. Note that these groups are disjoint by definition and thus other metrics such as Kendall Tau or Pearson correlation are not applicable. The choice of EMD is also inspired by the mathematical model in [7] where group unfairness is measured using the Wasserstein distance, which is another name for EMD.

To motivate the significance of subgroup fairness, consider the following scenario where a job owner is looking for an event planner in San Francisco on an online marketplace. Assume there are only two binary protected attributes: age = {young, old} and gender = {male, female}. Also assume that the process used to score the individuals on the marketplace includes only males regardless of their age group and *young* females in the top-10 ranking. If one were to examine either protected attribute alone, the scoring function might seem fair since both young and old individuals appear in the top-10 ranking and both males and females appear in the top-10 as well. However, it is only by looking at combinations of these attributes that one can truly unveil that the scoring process is unfair with respect to the subgroup *older females*.

Motivating Examples. To motivate our work, consider the following crowdsourcing scenario where a requester has posted a tweet generation task that requires good knowledge of English and the ability to follow task instructions. To be able to assign this task to workers on the crowdsourcing platform, the requester relies on a language test and the worker approval rate to compute a score for each worker that shows interest in the task. This score can then be used to rank those workers and the requester can pick the top-k ranked workers to finally perform the task. The requester can also use a threshold to assign the task to those workers whose scores surpass the threshold. Such scoring process can be unfair if it systematically disadvantages certain groups of workers based on their protected attributes. For instance, the language test can suffer from what is known as test bias [11]. Similarly, online ratings, such as approval rates in crowdsourcing platforms, are vulnerable to bias [12]. This might in turn cause the scoring function through which workers are ranked to be biased by systematically associating certain groups of workers, say women or older people or people from certain locations with lower scores compared to other comparable groups of workers.

As another example, consider a mounting job in New York City posted on a freelancing platform such as TaskRabbit. The job owner receives a ranked list of individuals on the platform for this job. Such ranking might be considered unfair if it is biased towards certain groups of people, say where white males are consistently ranked above black males or white females. This can commonly happen since such ranking might depend on the ratings of individuals and the number of their past jobs, both of which can perpetuate bias against certain groups of individuals.

Algorithm. Since we do not want to focus only on pre-defined groups, we must exhaust all possible ways of partitioning individuals on their protected attributes to quantify unfairness. We thus define an optimization problem as finding a partitioning of the decision space, i.e., individuals and their scores, that exhibits the highest average EMD between its partitions. Exhaustively enumerating all possible partitionings is exponential in the number of protected attributes. More precisely, given n protected attributes, the number of possible partitionings on those attributes is equal to $2^n - 1$. Even for small values of n, this can grow very fast. For instance, as mentioned above, the french criminal law defines 23 protected attributes, which yields $2^{23} - 1 = 8,388,607$ possible ways of partitioning individuals on those 23 protected attributes.

Therefore, we propose a heuristic algorithm, EMDP, which stands for EMD partitioner. At each step, EMDP greedily splits individuals on the worst attribute, i.e., the one that results in the partitioning with the highest average EMD between score distributions. This local decision is akin to the one made in decision trees using gain functions [13]. The algorithm stops when there are

no further attributes left to split on or when the current partitioning of individuals exhibits more unfairness than it would if its partitions were split further.

Data and Process Transparencies. Our framework assumes the availability of individuals' data (i.e., data transparency) and the scoring function (i.e., process transparency). To assess the validity of such assumptions in real-world online job marketplaces, we ran a series of user studies on Academic Prolific (https://prolific.ac/) to examine the relationship between transparency and fairness of scoring from the perspective of individuals being ranked for jobs. Our first two studies on *data transparency* found that individuals do not object to exposing their *protected attributes*, such as their age and location, to job owners for the purpose of being ranked for jobs. They also prefer job owners to see their *observed attributes* such as their approval rate and language skills, because they believe the scoring process will be more accurate in light of that information. Our third study on *process transparency* found that when asked to choose between a transparent scoring process and an opaque but more fair one, most individuals preferred the transparent one. This last result confirms a previously established hypothesis in Economics [14] where it was shown that people are more accepting of transparent procedures that treat them "with respect and dignity", making it easier to accept outcomes, even if they do not like those outcomes. In our last user study, *human judges*, that play the role of platform auditors, were asked to assess fairness under different *transparency settings*. We found that judges prefer data transparency over process transparency when assessing fairness. The intuitive explanation is that if individuals' data and jobs they qualify for are made available to a judge (data transparency), she can better infer the process through which the individuals are ranked for those jobs, and assess its fairness.

Evaluation. Our user studies show that individuals believe that an approach to quantify unfairness of scoring can operate more effectively if it has access to their protected and observed attributes, and that they are not sensitive about sharing such attributes if it results in a fairer scoring process. They also prefer the scoring process to be transparent, however they do not deem this crucial for quantifying unfairness. These findings motivated us to evaluate our approach in various transparency settings, the first of which is the case of transparent data/transparent process, where our framework is given access to individuals' attributes and the scoring function. We also evaluate our approach in the case of transparent data/opaque process, where our framework is given access to individuals' data and only the outcome of the scoring process. Finally, we also evaluate our approach in the case of opaque data/transparent process, where our approach is given access to *k-anonymized* individuals' data and the full scoring process.

Our evaluation results on a database of workers simulated from Amazon Mechanical Turk show that our proposed approach can efficiently and effectively quantify unfairness in scoring of individuals. They also suggest that our approach is effective in quantifying unfairness regardless of the transparency setting. Naturally, it is most effective when it has access to as much information as possible. This is especially true regarding data transparency. Our approach is most successful in quantifying unfairness when it has access to workers' protected attributes, regardless of whether the scoring process is transparent or opaque. This indeed coincides with our findings from the user studies, in which workers indicated being more successful in assessing fairness when they have access to individuals' attributes, regardless of whether the process is transparent or not.

Finally, to assess the effectiveness of our approach in quantifying unfairness on real data, we run an experiment on a dataset crawled from the online freelancing platform TaskRabbit. Our findings suggest that the majority of jobs on such platform during the time of crawling suffered from systematic bias based on ethnicity followed by gender followed by the combination of both. This is inline with recent studies where such assessment was manually done [9]. Our findings also suggest that jobs related to moving, handyman and yard work exhibited the highest amount of

Table 1. Task examples and their respective Construct and Observed spaces

Task	Construct space CS	Observed space OS
Tweet generation	Language level, General knowledge	Language test, General knowledge test
Audio transcription	Language level, Ability to recognize spoken words	Language test, Approval rate
Text translation	Language level	Language test, Approval rate
Image transcription	Basic language level	Language test, Approval rate
Image description	Written language level, Ability to follow instructions	Language test, Approval rate

unfairness in scoring compared to event staffing and general cleaning. Additionally, we used our approach to identify the locations that exhibited the highest amount of unfairness in scoring in our crawled dataset, which happened to be San Antonio in Texas, Louisville in Kentucky, and Cleveland in Ohio.

Contributions. We summarize our contributions as follows:

(1) We cast the problem of fairness of scoring in online job marketplaces as an optimization problem to find the partitioning of individuals based on their protected attributes that exhibits the highest unfairness.
(2) We develop an efficient algorithm that solves our optimization problem and finds groups of individuals, a.k.a., partitions, described with any combination of their protected attributes.
(3) We run user studies to examine the relationship between transparency and fairness of scoring in online job marketplaces and report findings that align with previous theories.
(4) We run extensive experiments on simulated and real datasets that demonstrate the ability of our approach in quantifying unfairness of scoring in online job marketplaces under different transparency settings.

System. Our approach is implemented as an interactive system to explore fairness of scoring in online job marketplaces [15, 16]. Our system appeals to different users. It can be used by *auditors*, whose role is to monitor the fairness of scoring in a given marketplace. It can be used by a *job owner*, who wants to study the behavior of a scoring function and its variants to understand their impact on the ranking of individuals, and choose the fairest one among them. Finally, it can be used by *the end-user*, who is being ranked, to assess the fairness of jobs on different marketplaces and make an informed decision about which one to target.

Our framework is general as it can be used both to verify hypotheses (given as baselines) and to generate new ones (for which no ground truth exists). New hypotheses can serve as a basis for real-world campaigns where they can be tested.

Outline. The manuscript is organized as follows. In Section 2, we present our data model. In Section 3, we describe our algorithm for quantifying unfairness of scoring and in Section 4, we present our experimental results. Finally, in Section 5, we review related work and conclude and present future directions in Section 6.

2 DATA MODEL

To study fairness of scoring in online job marketplaces, we adopt the mathematical data model for fairness in decision making defined in [7]. In that model, given a set of individuals W associated with a set of protected attributes $A = \{a_1, a_2, ..., a_n\}$, the scoring process takes place through the following three metric spaces over W:

- **Construct Space**: $CS = (W, d_w)$
  A metric space where d_w is the distance measured between individuals in W with respect

to their qualifications for a job. This space contains necessary but *unmeasurable* attributes for scoring individuals. Table 1 shows example jobs (i.e., tasks) and their construct space in a crowdsourcing setting. For instance, for an audio transcription task, $\mathcal{CS}$ contains the language level of a worker.

- **Observed Space**: $\mathcal{OS} = (\hat{W}, \hat{d})$
  A metric space where $\hat{W}$ are entities generated from individuals in W using an observation process $g : W \rightarrow \hat{W}$ and $\hat{d}$ is a distance between those entities. The observation process g serves as a proxy for the construct space and the observed space will thus contain a set of measurable attributes $B = \{b_1, b_2, \ldots, b_m\}$ through which those in the construct space can be inferred. For instance, the language level of a worker is inferred using a language test (see Table 1).
- **Decision Space**: $\mathcal{DS} = (O, d_O)$
  A metric space where O is a space of outcomes and d_O is a metric defined on O that results from a mapping $f : \hat{W} \rightarrow O$, where f is a scoring function that is used for scoring individuals in W. The decision space $\mathcal{DS}$ usually contains the potential of an individual to perform a job computed as a real-valued score using f. It could also simply contain a binary outcome: "an individual qualifies for a job" or "an individual does not qualify for a job", if a threshold is applied to generate individual qualification decisions. It can also simply contain the rank of each individual for the job.

We can now define data and process transparencies as follows, where, unless otherwise stated, by process we mean the scoring process:

- **Data Transparency** is defined as the availability of individuals' protected attributes $A = \{a_1, a_2, ..., a_n\}$ and observed attributes $B = \{b_1, b_2, \ldots, b_m\}$. By definition, the attributes in the construct space, $\mathcal{CS}$ are unmeasurable and hence opaque.
- **Process Transparency** is defined as the availability of the scoring function f that maps between the observed space $\mathcal{OS}$ and the decision space $\mathcal{DS}$.

Our aim is to quantify unfairness in the decision space $\mathcal{DS}$. We rely on the same common underlying assumption as in [7] in which individuals can be partitioned into groups, i.e., a collection of individuals that share a certain set of characteristics or protected attributes such as gender, race, or religion. The assumption is that in the construct space, $\mathcal{CS}$, all groups look essentially the same. In other words, there are no innate differences between groups of individuals defined via those potentially discriminatory characteristics. Unfairness occurs in the decision space when the same groups are treated differently by the scoring process, i.e., the scoring function f would favor certain individuals over others based on their group membership. This happens as a result of *structural bias*, when the construct space $\mathcal{CS}$ is not accurately represented by the observed space $\mathcal{OS}$ due to noise in the transformation between the two spaces.

For example, consider an audio transcription task in a crowdsourcing platform and assume that the construct space consists of the language level of the workers. Also, assume that workers are grouped based on age and gender into four groups: young males, old males, young females and old females. Our assumption is that all four groups have the same distribution of language levels. Now, assume that the observed space is represented using a language test to quantify the language level of workers. If that test suffers from structural bias, i.e., unequal treatment of the four groups, then unfairness might occur when the language test scores are used in scoring the workers for the task.

To formulate fairness of scoring in online job marketplaces, we propose to identify the extent of unfairness that might occur given a set of individuals and a scoring function. To do this, we assume both *data and process transparencies* and we measure unfairness as the *average* distance between partitions of individuals in the decision space. Unlike previous work where partitions were defined

or known a priori (e.g., [9]), in this work we explore the space of all possible groups of individuals defined by a combination of their protected attributes. The goal becomes finding the *most* unfair partitioning of individuals under the scoring function f. This can be viewed as a worst-case scenario, where we want to measure how differently f treats groups under this worst-case partitioning of individuals. We cast this goal as an optimization problem as follows.

Definition 1 (Most Unfair Partitioning Problem). *We are given a set of individuals W, where each individual is associated with a set of protected attributes $A = \{a_1, a_2, ..., a_n\}$ and observed attributes $B = \{b_1, b_2, \ldots, b_m\}$. The protected attributes are inherent properties of the individuals such as gender, age, ethnicity, origin, etc. The observed attributes represent the qualifications of an individual for jobs and could include, for instance, the approval rate of the individual, a language test score and a mathematical test score in case of crowdsourcing or rating and number of past jobs and skills in case of freelancing. We are also given a scoring function $f : W \rightarrow [0, 1]$ through which the individuals are ranked for jobs. Our goal is to* fully *partition the individuals in W into k* disjoint *partitions $P = \{p_1, p_2, \ldots, p_k\}$ based on their protected attributes in A using the following optimization objective:*

$$\begin{aligned} \underset{P}{\text{argmax}} \quad & unfairness(P, f) \\ \textit{subject to} \quad & \forall i, j \; p_i \bigcap p_j = \phi \\ & \bigcup_{i=1}^{k} p_i = W \end{aligned}$$

We now define how to compute the amount of unfairness of a function f for a partitioning P, or $unfairness(P, f)$ in the above optimization problem.

Definition 2 (Earth Mover's Distance Unfairness). *Given a set of individuals W, a full-disjoint partitioning of the individuals $P = \{p_1, p_2, \ldots, p_k\}$ and a scoring function f, the amount of unfairness of f for the partitioning P is measured as the average Earth Mover's Distance (EMD) between the distribution of scores of individuals using f in the different partitions in P, which is computed as follows:*

$$unfairness(P, f) = \underset{i,j}{\text{avg}} \quad EMD(h(p_i, f), h(p_j, f))$$

where $h(p_i, f)$ is a histogram of the scores of individuals in p_i using f.

Note that another aggregation method other than average can be used in the above formulation. For instance, one could use maximum or minimum to obtain the partitioning with the maximum or minimum EMD between the distribution of scores of individuals in the different partitions.

Example. In Table 2, we display a toy example consisting of 10 workers in a crowdsourcing platform, their protected attributes A and their observed attributes B. Consider a scoring function for tweet generation applied to this set of workers which is of the form:

$$f = 0.3 \times LanguageTest + 0.7 \times ApprovalRate$$

Figure 1 shows the partitioning of workers that results in the highest average pairwise EMD induced by the function f. In our toy example, the optimum partitioning is the one resulting from splitting the workers based on *Gender* first then *YearOfBirth* to get to the partitioning shown in the leaf nodes. To arrive to this partitioning, one must exhaust all possible *full disjoint* partitionings of workers based on any combination of their protected attributes A and for each possible partitioning

Table 2. Toy example consisting of 10 workers and a scoring function for tweet generation

Worker	Gender	Country	YearOfBirth	Language	Ethnicity	Experience	LanguageTest	ApprovalRate	f(w)
w1	Female	India	2004	English	Indian	0	0.50	0.20	0.29
w2	Male	America	1976	English	White	14	0.89	0.92	0.911
w3	Male	India	1976	Indian	White	6	0.35	0.65	0.560
w4	Male	Other	1963	Other	Indian	18	0.64	0.76	0.724
w5	Female	India	1963	Indian	Indian	21	0.85	0.90	0.885
w6	Male	America	1995	English	African-American	2	0.42	0.20	0.266
w7	Female	America	1982	English	African-American	16	0.95	0.98	0.971
w8	Male	Other	2008	English	Other	0	0.30	0.15	0.195
w9	Male	Other	1992	English	White	2	0.32	0.25	0.271
w10	Female	America	2000	English	White	5	0.76	0.56	0.620

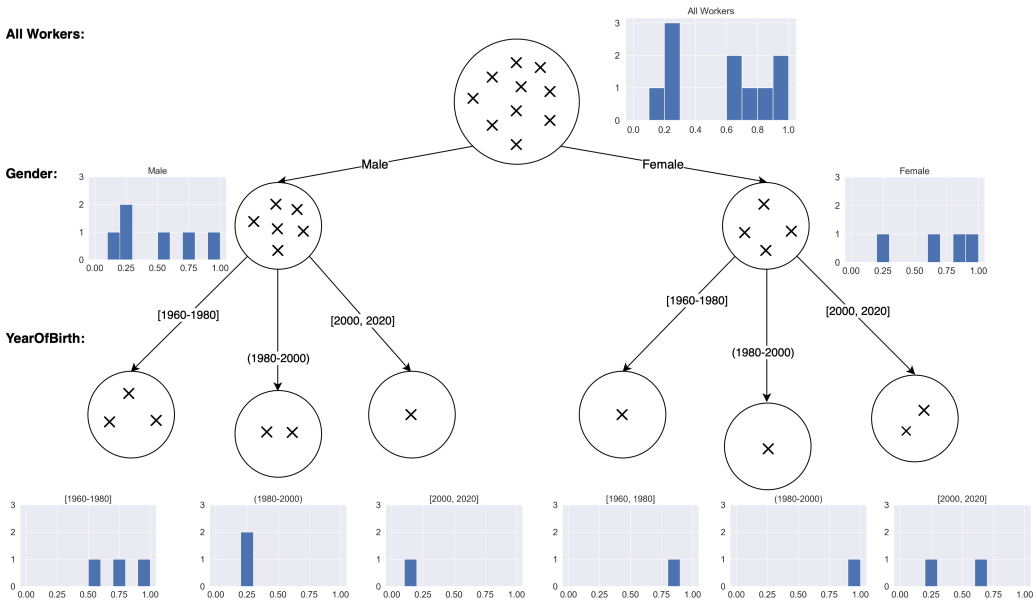


Fig. 1. Optimum partitioning of the toy example data

compute the average EMD between all pairs of partitions. To do that, we generate a histogram for each partition as indicated in Figure 1 based on the function scores by creating equal bins over the range of f and counting the number of workers whose function values $f(w)$ fall in each bin. Once the partitioning with highest average pairwise EMD has been identified, it is up to the user, requester or crowdsourcing platform developer, to decide on the right subsequent action. *Note that since the partitions are disjoint, other metrics such as Kendall Tau or Pearson correlation will not be applicable.*

3 ALGORITHM

Our optimization problem for finding the most unfair partitioning is hard since the number of possible partitionings is exponential in the number of protected attributes. More precisely, assuming n protected attributes, then there are $O(2^n)$ possible partitionings of users based on those n protected attributes. This can be very large even for relatively small values of n. For this reason, we propose a

ALGORITHM 1: EMDP (W: a set of individuals, f: a scoring function, A: a set of attributes)

```
a = worstAttribute(W, f, A)
A = A − a
current = split(W, a)
currentAvg = averageEMD(current, f)
while A ≠ ∅ do
    a = worstAttribute(current, f, A)
    A = A − a
    children = split(current, a)
    childrenAvg = averageEMD(children, f)
    if currentAvg ≥ childrenAvg then
        break
    else
        current = children
        currentAvg = childrenAvg
    end if
end while
return current
```

heuristics-based algorithm to identify a partitioning of individuals with respect to our optimization objective within reasonable time.

Algorithm 1 shows the pseudocode of our EMD partitioning algorithm EMDP. It generates a partitioning of the individuals in a greedy manner using the EMD of the partitions. EMDP is based on decision trees with EMD as utility [13]. It starts by splitting the individuals on the *worst* attribute with respect to EMD. This is done by trying out all possible attributes one at a time, and associating to each attribute-value partition, one histogram of the scores of all the individuals it contains. For each candidate attribute, EMDP computes the average pairwise EMD over histograms associated to the partitions obtained with the values of that attribute. It then returns the attribute with the highest average pairwise EMD and splits on that attribute. In the subsequent splitting steps, EMDP iteratively partitions the individuals using the other attributes in the same manner and only stops when the average pairwise EMD of the current partitioning is greater than that of the next candidate partitioning.

Algorithm EMDP has a complexity of $O(n^2)$ in the worst case, where n is the number of protected attributes. At first the algorithm tries out n possible partitionings using a single attribute, and then it tries $n-1$ partitionings corresponding to the remaining $n-1$ attributes and so on until there are no more attributes left. Hence, it will examine a total of $n + (n-1) + \cdots + 1 = O(n^2)$ partitionings in the worst case (i.e., if the termination condition was ever met).

Note that in case of positive discrimination, i.e., when only certain individuals are considered for jobs based on some of their protected attributes, one can seamlessly run EMDP on those individuals only and exclude those protected attributes necessary for the job when quantifying subgroup unfairness.

4 EXPERIMENTS

Our algorithm described in the previous section assumes both data and process transparencies, i.e., the availability of individuals' protected and observed attributes as well as the scoring function. Before we set out to evaluate its effectiveness in quantifying unfairness of scoring in online job marketplaces, we first run a series of user studies to assess the feasibility of such assumptions and to gauge individuals' perceived fairness of scoring. We then evaluate our algorithm and compare it to

a series of baselines under various transparency settings, namely transparent data and transparent process, transparent data and opaque process, and opaque data and transparent process. Finally, we apply our algorithm on real data crawled from the freelancing platform TaskRabbit and report findings on fairness of scoring for various jobs on this platform.

4.1 User Studies on Transparency Settings

Our user studies have two goals. The first goal is to verify how sensitive individuals on online job marketplaces are to making their data transparent. The second goal is to examine the impact of data and process transparencies on the ability of judging fairness of scoring on these platforms from the individuals' perspective. Without loss of generality, we focus on crowdsourcing as one example of such online marketplaces in our user studies. However, we believe our findings to be applicable to other types of online job marketplaces.

We designed different surveys to address the questions:

(1) How do workers in crowdsourcing platforms feel about sharing their protected attributes when it comes to scoring them for tasks?
(2) How do workers feel about sharing their observed attributes?
(3) Do workers prefer a transparent scoring process or an opaque one?
(4) From the workers point of view, how does the interplay between data and process transparencies affect their ability to judge fairness of scoring?

4.1.1 Recruitment. We recruited our subjects from Prolific Academic[2], a crowdsourcing platform with a workforce accustomed to completing surveys. Recruited subjects were redirected to *esurv.org* to fill out forms we designed for each user study. We based our choice of sample size on the following formula, that is estimated using the central limit theorem [17]:

$$\text{Sample size} = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1+(\frac{z^2 \times p(1-p)}{e^2 N})}$$

where:

- N = 22378 is the population size:, the Prolific Academic workforce eligible for our survey completion task,
- e = 8% is the margin of error: the percentage of deviation in result in the sample size compared with the total population,
- z = 90% is the confidence level: if the task is repeated 100 times, 90 times out of 100 the result would lie within the margin of error, and
- p = 50% is the percentage value: it is the expected result value of the experiment. It is advised to put it at 50% when the result is not known.

Our sample size rounded up at 105 subjects per user study.

4.1.2 Survey Design and Results. We designed four surveys to answer the questions stated in the beginning of this section on perceived fairness by workers and their preference for data and process transparencies. In all four, we assumed an audio transcription task in English. We also assumed that there are five workers, some of which are qualified for the task and some are not. Each worker has only one single protected attribute, namely *nationality*, which is either English/American or non-English/American. Similarly, each worker has one observed attribute, namely *qualification*, which is assumed to be a combination of English level and approval rate. The requester gets to choose which workers are qualified for the task based on a scoring process that can use a combination of the workers' protected and observed attributes.

[2]https://www.prolific.ac/

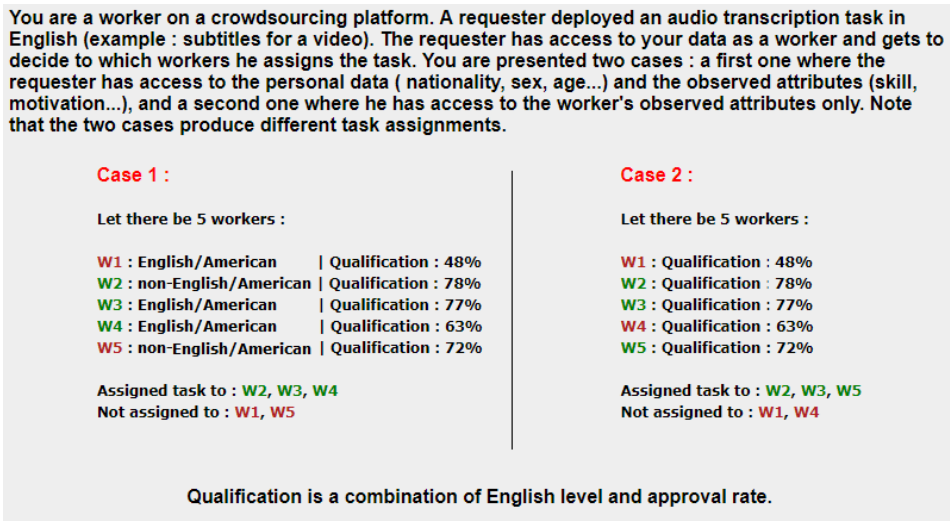


Fig. 2. Survey to assess how workers feel about sharing their protected attributes

Our first survey shown in Figure 2, was used to assess how workers feel about sharing their *protected* attributes. The workers were presented with two scenarios, one where the requester of the task had access to both protected and observed attributes of workers, and hence might use both to score the workers, and another where the requester had only access to observed attributes. The workers were asked to choose which scenario they prefer. The two scenarios received equal votes implying that workers are not sensitive about sharing their protected attributes even if they are used to score them. Those who preferred not to share their protected attributes raised concerns about fairness if such attributes were made available. Note that in both cases we did not provide the scoring function itself (i.e., we assumed an opaque scoring process).

Our second survey is very similar to the first one (see Figure 3) and was used to assess how workers feel about sharing their *observed* attributes. The workers were presented with two scenarios, one where the requester had access to both protected and observed attributes and another where the requester had only access to protected attributes. The workers were asked to indicate which scenario they prefer and overwhelmingly, 83.65% of them, chose the first scenario that makes observed attributes transparent to requesters. They indicated that requesters will do a better job at scoring them for tasks, if they have access to their observed attributes such as English proficiency.

Our third survey was used to assess whether workers prefer a transparent scoring process or an opaque one (see Figure 4). The workers were again given two scenarios to choose from. The first represented the case of a transparent scoring process, where the workers were shown the scoring function by which the requester scores the workers for the tasks. The second scenario represented the case of an opaque process where the workers were only presented with the scoring process decisions (i.e., which workers were qualified for the task and which were not). In both cases, all workers' protected and observed attributes were available (i.e., transparent data). The majority of workers, 62.85%, preferred the transparent process scenario because they believed it was more fair. This confirms a previous result in Economics [14, 18] that people are more accepting of procedures that treat them with respect and dignity, making it easier to accept outcomes, even if they do not like those outcomes.

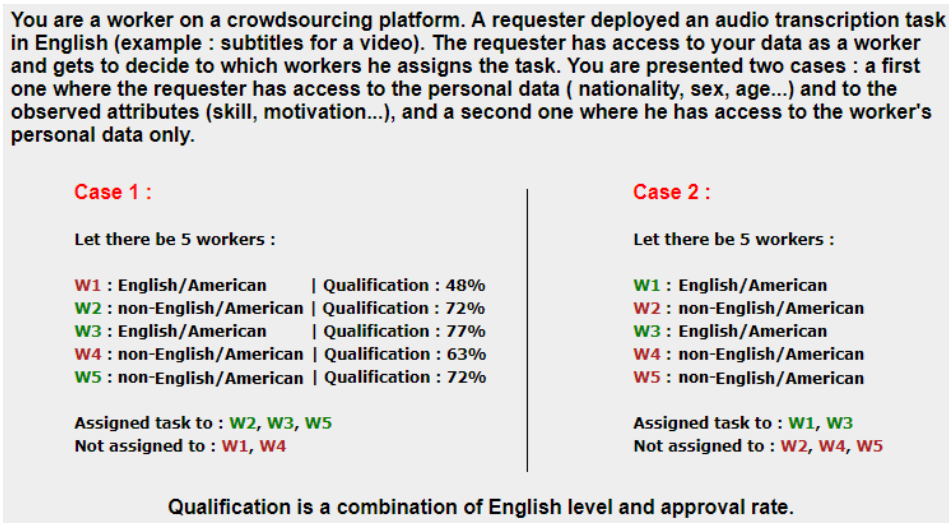


Fig. 3. Survey to assess how workers feel about sharing their observed attributes

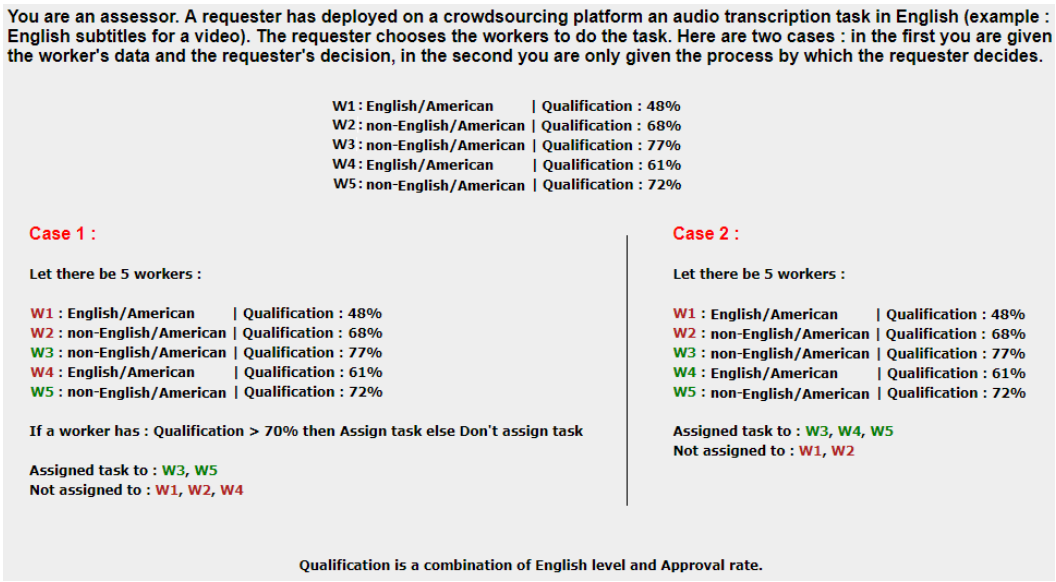


Fig. 4. Survey to assess whether workers prefer a transparent scoring process or an opaque one

Finally, to assess the impact of data and process transparencies on workers' ability to judge fairness in scoring, we designed **a fourth survey**, again about audio transcription (see Figure 5). The survey consisted of two scenarios, the first representing the setting of a transparent data and opaque process and the second representing an opaque data and transparent process setting. In the first scenario, only the workers' protected and observed attributes were shown. In the second scenario, only the scoring function was shown. In both scenarios, the workers who participated in

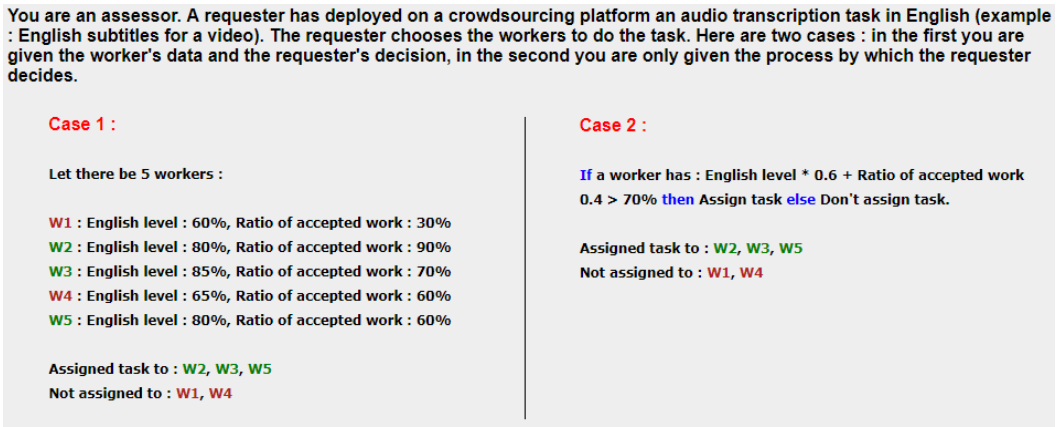


Fig. 5. Survey to assess the impact of data and process transparencies on workers' ability to judge fairness

the survey were asked to pretend they are auditors judging the fairness of the scoring process and were asked to choose which scenario made it easier for them to judge. 66.67% of the workers were in favor of scenario 1 and 33.33% of scenario 2. A further examination of the comments from the workers showed that they preferred the first scenario because the transparent data (both worker data and decisions) made it easier for them to judge the fairness of the scoring process, even though the process itself was not shown. Note that we did not consider the scenario where both data and process are opaque since it would be impossible for the workers to judge fairness in that case. We also did not consider the scenario where both data and process are transparent since it was already considered in our third survey, which has shown that in the case where data is transparent, workers prefer transparent scoring process as well, perceiving it as more fair.

To summarize, our user studies have shown that workers do not mind sharing their protected and observed attributes on crowdsourcing platforms and generally prefer participating in transparent scoring processes. They also believe that in order to judge fairness in scoring on these platforms, access to protected and observed attributes of workers is necessary even if the scoring process itself was not available for judging. Our approach and algorithm have thus far assumed the availability of both individuals' data and the scoring process. In the next experiment, we assess their effectiveness in this setting and then in our next two experiments, we relax this assumption and evaluate our algorithm in the case where the scoring process is not available, as well as when the individuals' data is k-anonymized [19].

4.2 Transparent Data - Transparent Process

In this experiment, we evaluate the effectiveness of our approach in quantifying unfairness in the case of transparent data and process. We run a simulation of a crowdsourcing platform using various sets of *active* workers and various scoring functions to compare our proposed algorithm EMDP against a set of baselines.

4.2.1 Setting. We generate three sets of active workers $\mathcal{W}$ of different sizes: 50, 500 and 7300 (the estimated number of Amazon Mechanical Turk workers who are active at any time [20]). Each w in $\mathcal{W}$ has six protected attributes:

- Gender = {Male, Female},
- Country = {America, India, Other},

Table 3. Average EMD and runtime for 50 workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
EMDP	**0.185**	0.184	**0.167**	0.264	0.241	2.348	2.211	2.34	2.305	3.453
R-EMDP	0.180	0.182	0.159	0.263	0.243	0.89	0.796	0.785	1.031	0.952
FULL	0.174	0.179	0.158	0.260	0.239	**0.417**	**0.462**	**0.462**	**0.463**	**0.465**
OPTIMUM	**0.185**	**0.192**	**0.167**	**0.272**	**0.257**	11.408	11.405	11.583	12.5	13.064

Table 4. Average EMD and runtime for 500 workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
EMDP	**0.196**	**0.194**	**0.177**	**0.246**	**0.253**	163.036	180.181	181.441	181.88	181.017
R-EMDP	0.195	0.193	**0.177**	**0.246**	**0.253**	86.697	75.017	89.363	91.35	76.096
FULL	0.195	0.193	**0.177**	**0.246**	**0.253**	**41.588**	**41.355**	**41.16**	**40.931**	**41.274**
OPTIMUM	**0.196**	**0.194**	**0.177**	**0.246**	**0.253**	363.866	367.177	367.012	413.061	367.828

- YearOfBirth = [1950, 2009],
- Language = {English, Indian, Other},
- Ethnicity = {White, African-American , Indian, Other}, and
- YearsOfExperience = [0,30],

and two observed attributes:

- LanguageTest = [25,100] and
- ApprovalRate = [25,100].

The values of those attributes are populated randomly so as to avoid injecting any bias in the data ourselves. Moreover, we define five different task qualification functions:

- $f_1 = 0.3 \times LanguageTest + 0.7 \times ApprovalRate$
- $f_2 = 0.7 \times LanguageTest + 0.3 \times ApprovalRate$
- $f_3 = 0.5 \times LanguageTest + 0.5 \times ApprovalRate$
- $f_4 = 1 \times LanguageTest + 0 \times ApprovalRate$
- $f_5 = 0 \times LanguageTest + 1 \times ApprovalRate$

We compare our proposed algorithm EMDP to two baselines. The first baseline, which we refer to as R-EMDP, is a copy of our algorithm EMDP that uses a random attribute instead of the worst attribute to split the workers at each step. This baseline was used to attest the validity of our greedy heuristic of choosing the worst attribute to split on. The second baseline, which we refer to as FULL, is an algorithm that splits the workers based on *all* their protected attributes resulting in a full partitioning. This baseline was used to validate the effectiveness of our stopping condition. We also compare our algorithm EMDP to an optimum algorithm that solves our optimization problem exactly by exhaustively examining every possible partitioning of the workers using any combination of their protected attributes and returning the one with the highest average pairwise EMD between partitions.

4.2.2 Simulation Results. Tables 3, 4 and 5 show the average EMD obtained by the different algorithms for the three datasets and their runtimes. Our first observation from the three tables is that for all datasets, functions f_4 and f_5 exhibit the highest unfairness as measured by the average pairwise EMD for all the partitions retrieved by each algorithm. Recall that these two

Table 5. Average EMD and runtime for 7300 workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
EMDP	**0.163**	**0.163**	**0.151**	**0.210**	**0.211**	2343.189	2333.634	1894.409	1911.65	1910.175
R-EMDP	**0.163**	**0.163**	**0.151**	**0.210**	0.170	1878.688	1735.444	**1413.876**	1360.265	**1105.924**
FULL	**0.163**	**0.163**	**0.151**	**0.210**	**0.211**	**1453.626**	**1449.466**	1450.712	**469.839**	1467.606
OPTIMUM	**0.163**	**0.163**	**0.151**	**0.210**	**0.211**	2944.879	2619.262	2364.807	2409.187	2413.059

functions are the ones that rely on one observed attribute only (LanguageTest in case of f_4 and ApprovalRate in case of f_5). *This indicates that when the scoring process uses fewer observed attributes to score individuals, the chance of unfairness increases. In our simulation, since the attribute values were generated at random, there is a higher chance that the scoring function correlates with a single protected attribute than with multiple attributes.*

Second, we observe that our algorithm EMDP consistently outperforms or do as good as both baselines R-EMDP and FULL for almost all datasets and functions. Particularly, EMDP beats both its baselines in 6 cases out of 15, and ties with one of them or both in 8 cases and loses to one of them (random counterpart) in just one case. On the other hand, EMDP achieves the optimum partitioning in 12 cases out of 15. Note that in the case of 7300 workers, all algorithms returned the full partitioning tree, i.e., using all protected attributes, which is the same as the partitioning returned by the FULL algorithm. *We conjecture that it is due to the random values of all attributes.*

In terms of efficiency, FULL is obviously the fastest algorithm since it does not involve any checks and partitions all the individuals upfront using all attributes. OPTIMUM takes the most time to terminate since it has to exhaustively examine an exponential number of partitionings. On the other hand, our algorithm EMDP incurs additional time compared to its random counterpart since at each splitting step, it needs to examine all remaining attributes to determine the worst one (i.e., the one which might result in the highest average EMD). All these factors contributed to the increased time to execute EMDP compared to R-EMDP and FULL. Finally, we observe that the larger the dataset is, the more time it took for all algorithms to finish. This is very intuitive given that the larger the dataset is, the larger the individual histograms are and the more time it takes to compute the pairwise EMD between them. Moreover, the deeper the partitioning tree, the larger the number of histograms that need to be compared.

4.2.3 Qualitative Results. In addition to our simulation where we used a set of *random* scoring functions, we also ran our algorithm on the following set of carefully-constructed functions, which are explicitly unfair by design:

- f_6: this function is unfair against females by setting the scoring function of workers as follows: $f_6(w) > 0.8$ if w is male and $f_6(w) < 0.2$ if w is female.
- f_7: this function sets the score of workers in a biased manner based on their gender and nationality as follows: $f_7(w) > 0.8$ if w is male and American, $f_7(w) < 0.2$ if w is female and American, $0.5 < f_7(w) < 0.7$ if w is Indian, either male or female, $f_7(w) > 0.8$ if w is female with any other nationality, and $f_7(w) < 0.2$ if w is male with any other nationality.
- f_8 : this function was designed as follows: $f_8(w) > 0.8$ if w is female and American, $0.5 < f_8(w) < 0.8$ if w is female and Indian and $f_8(w) < 0.2$ if w is female with another nationality.
- f_9: this function was designed to correlate with three protected attributes, namely: ethnicity, language and year of birth in a similar manner as the previous ones.

As can be seen from Table 6 for the case of 7300 workers, EMDP retrieves the optimal partitionings with the highest possible average EMD. In addition, the resulting partitionings are the ones expected,

Table 6. Average EMD for 7300 workers and biased functions

Algorithm	Average EMD			
	f_6	f_7	f_8	f_9
EMDP	**0.800**	**0.426**	**0.456**	**0.358**
R-EMDP	0.492	0.342	0.295	0.280
FULL	0.421	0.368	0.338	**0.358**
OPTIMUM	**0.800**	**0.426**	**0.456**	**0.358**

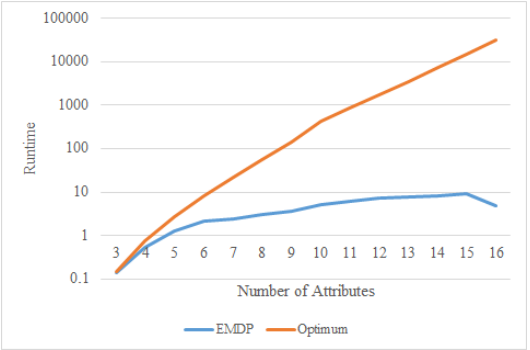


Fig. 6. Runtime of EMDP and Optimum when varying the number of attributes

i.e., using the attributes for which the functions were designed to correlate with. For example, for f_6, EMDP partitions the workers on only gender. Similarly, for f_7, it partitions the workers on both gender and country. Finally, we observe that *overall for all functions and algorithms, the average EMD is much higher compared to the functions used in our simulation experiment, which indicates that our optimization problem is indeed effective in capturing unfairness in scoring as conjectured.*

Finally, to test the scalability of our algorithm EMDP, we ran a simulation of a new dataset consisting of 50 workers and varying the number of protected attributes from 3 to 16. Figure 6 shows the runtime of each algorithm in log-scale versus the number of protected attributes. Note that the optimum algorithm did not finish running in 24 hours when the number of attributes exceeded 16. Also note that for 16 attributes, EMDP ended up splitting on only 3 attributes after which the stopping condition was met, which explains the smaller running time compared to the case of 15 attributes.

4.3 Transparent Data - Opaque Process

In this experiment, we run our simulation again from the previous experiment, however, we assume an opaque process. Our algorithms are therefore given access only to the workers' data (both protected and observed attributes) and to the scoring process decisions, which are binary values indicating whether a worker is qualified for the task or not. Worker scores are not available since the process being assessed is opaque. To do so, we assign a value of 1 to 10%, 30% and 50% of the workers at random and the rest are assigned the value 0. We do not consider any other cases such as assigning a value of 1 to 70% or 90% of the workers since they are redundant because we only have binary decisions. We then run our algorithm EMDP as well as our baselines R-EMDP and FULL, and the optimum algorithm OPTIMUM over the three different worker datasets. Note that in this setting, the only change that will take place in any of the algorithms is the way the histograms

Table 7. Average EMD and runtime for 50 workers and random binary decisions

Algorithm	Average EMD			time (in secs)		
	10%	30%	50%	10%	30%	50%
EMDP	**0.067**	0.218	**0.255**	1.783	2.73	3.006
R-EMDP	0.041	0.204	**0.255**	0.389	0.692	1.117
FULL	0.020	0.204	**0.255**	**0.347**	**0.347**	**0.349**
OPTIMUM	**0.067**	**0.221**	**0.255**	10.481	10.542	10.515

Table 8. Average EMD and runtime for 500 workers and random binary decisions

Algorithm	Average EMD			time (in secs)		
	10%	30%	50%	10%	30%	50%
EMDP	0.099	**0.217**	**0.249**	101.623	134.529	134.446
R-EMDP	0.099	**0.217**	**0.249**	51.53	62.157	55.484
FULL	0.099	**0.217**	**0.249**	**31.418**	**31.354**	**31.367**
OPTIMUM	**0.102**	**0.217**	**0.249**	269.767	268.852	268.183

of workers per partition are generated. Instead of building a histogram to represent the distribution of the function score values, the histograms will represent the distribution of qualification decisions per partition. That is, each histogram will consist of two bins, one representing those workers who did not qualify for the task (value of 0) and the other those who qualified for the task (value of 1).

4.3.1 Simulation Results. As can be seen from Tables 7, 8 and 9, EMDP retrieves the optimum partitioning in most of the cases (7 out of 9). It also outperforms or do as well as both its baselines in all the cases. Moreover, we observe that as we increase the percentage of workers who are considered qualified (i.e., assigned the value 1), the amount of unfairness as measured by the average EMD increases. This is consistent across all algorithms and datasets. Since we randomly picked the workers to be considered qualified in this experiment, it is intuitive that the fewer such workers are, the less likely is their chance of correlating with protected attributes. That is, one would expect that qualified workers would fall in random partitions across the partitioning space. As the number of qualified workers increases, their chance of correlating with the protected attributes increases too. In terms of efficiency, again similar to the case of transparent data and process, OPTIMUM is the most time-consuming, followed by EMDP, then R-EMDP and finally FULL is the least time-consuming.

4.3.2 Qualitative Results. Similar to the case of the transparent data and process, we constructed two opaque scoring processes that were unfair by design. The first such process qualifies only males (i.e., assigns them a value of 1) and the second qualifies workers based on their country and gender. Table 10 shows the average EMD for each of the algorithms for our various datasets. As can be seen from the table, both EMDP and OPTIMUM retrieve the partitionings with the highest average EMD compared to all other algorithms for the case of 7300 workers. Moreover, they partition the workers as expected based on the attributes that correlate with the scoring process, namely gender in the first case and gender and country in the second case. Finally, it can be observed that for all the algorithms, the average EMD is higher than those retrieved in the case of the random scoring processes shown in Tables 7, 8 and 9.

Table 9. Average EMD and runtime for 7300 workers and random binary decisions

Algorithm	Average EMD			time (in secs)		
	10%	30%	50%	10%	30%	50%
EMDP	**0.083**	**0.169**	**0.197**	1445.327	1659.874	1676.844
R-EMDP	0.053	**0.169**	**0.197**	**738.181**	1245.849	1152.123
FULL	**0.083**	**0.169**	**0.197**	1007.394	**1021.390**	**1010.206**
OPTIMUM	**0.083**	**0.169**	**0.197**	2100.375	2099.085	2102.848

Table 10. Average EMD for 7300 workers and biased binary decisions

Algorithm	Average EMD	
	gender	gender & country
EMDP	**0.500**	**0.300**
R-EMDP	0.306	0.225
FULL	0.250	0.250
OPTIMUM	**0.500**	**0.300**

4.4 Opaque Data - Transparent Process

In our third experiment, we run our simulation again from our first experiment, however, in this case we assume that our algorithms are given access to workers k-anonymized data [21], in addition to the scoring function through which the workers are ranked. This represents a setting where we have *partially opaque* data and a transparent process. In our context, the data is k-anonymized with respect to the workers' protected attributes if there are at least k workers whose protected attribute values are the same. In case the k-anonymity condition does not hold, the k-anonymization approach generalizes some of the protected attributes to satisfy the k-anonymity property. We rely on the ARX data anonymization tool[3]. The tool takes as input a set of quasi-identifiers, which we assume to be all the protected attributes in our case, a value of k, and a generalization hierarchy for each attribute. For year of birth and years of experience, we define four levels of generalization, which will replace individual values of such attributes with a broader range. For example, to anonymize year of birth, the tool can generalize the exact values to ranges such as $[1960-1070]$, $[1980-1990]$, $[1990-2000]$, etc. In case this is not sufficient to achieve k-anonymity, a higher level of generalization is used, for instance by generalizing the year of birth to the ranges $[1960-1980]$, $[1980-2000]$, etc. For the attributes gender, ethnicity, language and country, which have only a hand-full of values in our datasets, we only define one level of generalization, which basically results in anonymizing the whole attribute. This in turn entails that these fully anonymized attributes are discarded by our algorithms when partitioning workers.

We set out to investigate the applicability of our algorithms in quantifying unfairness of scoring even in the case when data is k-anonymized. We run the k-anonymization ARX tool on our three simulation datasets, setting $k = 10$ and then passing the resulting k-anonymized datasets as input to our algorithm EMDP. We set $k = 10$ since our data is randomly generated, and thus most of the workers have unique tuples in terms of their protected attributes (i.e., the data is highly un-anonymized). For the case of 50 workers, the k-anonymization approach completely anonymized the attributes country, year of birth, language and ethnicity and generalized the years of experience to 20 years spans. For the case of 500 workers, the k-anonymization approach completely anonymized the

[3]https://arx.deidentifier.org/

Table 11. Average EMD and runtime for 50 k-anonymized workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
50 workers										
EMDP	**0.070**	**0.060**	**0.053**	**0.074**	**0.092**	0.005	0.005	0.004	0.004	0.004
R-EMDP	**0.070**	0.058	**0.053**	**0.074**	**0.092**	0.004	0.004	0.004	0.004	0.004
FULL	**0.070**	0.057	**0.053**	**0.074**	**0.092**	**0.003**	**0.003**	**0.003**	**0.003**	**0.003**
OPTIMUM	**0.070**	**0.060**	**0.053**	**0.074**	**0.092**	0.004	0.004	0.004	0.004	0.005

Table 12. Average EMD and runtime for 500 k-anonymized workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
500 workers										
EMDP	**0.063**	**0.071**	**0.065**	**0.089**	**0.064**	0.205	0.203	0.201	0.204	0.204
R-EMDP	**0.063**	**0.071**	**0.065**	**0.089**	**0.064**	0.164	0.179	0.163	0.162	0.165
FULL	**0.063**	**0.071**	**0.065**	**0.089**	**0.064**	**0.139**	**0.140**	**0.138**	**0.142**	**0.141**
OPTIMUM	**0.063**	**0.071**	**0.065**	**0.089**	**0.064**	0.211	0.211	0.21	0.213	0.211

Table 13. Average EMD and runtime for 7300 k-anonymized workers and random functions

Algorithm	Highest EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
EMDP	**0.042**	**0.040**	**0.038**	**0.047**	**0.049**	8.763	9.978	8.742	8.902	8.834
R-EMDP	**0.042**	**0.040**	**0.038**	**0.047**	**0.049**	6.663	7.010	6.863	7.159	7.042
FULL	**0.042**	**0.040**	**0.038**	**0.047**	**0.049**	**5.647**	**5.623**	**5.636**	**5.663**	**5.671**
OPTIMUM	**0.042**	**0.040**	**0.038**	**0.047**	**0.049**	13.44	13.212	13.33	13.307	13.278

year of birth, language, ethnicity and years of experience. Finally, for the case of 7300 workers, the k-anonymization approach completely anonymized the year of birth and ethnicity, and generalized the years of experience to 20 years spans.

4.4.1 Results. Tables 11, 12 and 13 display the average EMD and the runtime of our algorithm EMDP, OPTIMUM and the baselines with the k-anonymized instances of our three datasets. We observe that overall, the average EMD values returned are all less than the case when non-anonymized data was available, regardless of the algorithm, scoring function or dataset. This is intuitive given that the algorithms have fewer attributes and attribute values to split on. However, we also observe that for the cases of 50 and 7300 workers, the obtained average EMD values are higher for f_4 and f_5 compared to the first three functions, which is inline with the results obtained for the case of transparent data/transparent process from our first experiment. This is also true for f_4 in the case of 500 workers. We observe that in most cases, the results of all algorithms were very similar, which again can be attributed to the reduction in the partitioning space induced by the k-anonymization process. In terms of efficiency, again EMDP is the slowest after OPTIMUM, however, in the case of k-anonymization, we also observe that the runtimes of all algorithms are reduced since the number of possible partitions is reduced and hence the algorithms compare fewer histograms to compute the average EMD.

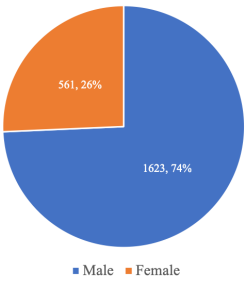


Fig. 7. Gender breakdown

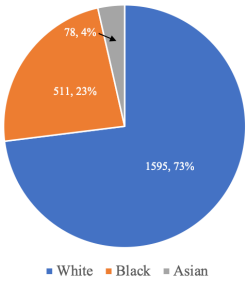


Fig. 8. Ethnic breakdown

We conclude by pointing out that our algorithm has shown to be effective in quantifying unfairness in scoring of individuals regardless of the transparency setting. Naturally, it is most effective when it has access to as much information as possible. This is especially true regarding data transparency. Our algorithm is most successful in quantifying unfairness when it has access to workers' protected attributes, regardless of whether the scoring process is transparent or opaque. This coincides with our finding from the user studies, in which workers indicated that they were more successful in assessing fairness when they have access to workers' attributes, regardless of whether the scoring process is transparent or not.

4.5 TaskRabbit

In this final experiment, we demonstrate the effectiveness of our approach on real data. More precisely, we run our algorithm EMDP on a dataset crawled from the online freelancing platform TaskRabbit[4] and study fairness of scoring for various services (i.e., job types) and in various locations on this platform.

4.5.1 Setting. TaskRabbit is an online marketplace that matches freelance labor with local demand, allowing consumers to find immediate help with everyday tasks, including cleaning, moving, delivery and handyman work. TaskRabbit is supported in 45 different cities mostly in the US. For each one of the 45 locations available on TaskRabbit, we retrieved up to 20 of the most popular services offered in that particular location. We thus generated a total of 278 queries, where each query was a combination of a location and a service, e.g., *Home Cleaning* in *New York*. Note that all queries were generated with the service date set one week in the future relative to the crawl date.

For each query, we extracted the rank of each tasker per query, their badges, reviews, profile pictures, and hourly rates, where the number of taskers returned per query was limited to 50. Since the demographics of the taskers were not readily available on the platform, we crowdsourced the taskers' profile pictures to obtain such demographics. We asked contributors on Figure Eight[5] to indicate the gender and ethnicity of the TaskRabbit taskers based on their profile pictures. Each profile picture was labeled by three different contributors on Figure Eight and a majority vote was employed to obtain a final label per demographic attribute for each tasker. The contributors were given pre-defined categories for gender = {male, female} and ethnicity = {asian, black,white}. We also computed the Kappa coefficient to report agreement between contributors. The values for gender and ethnicity are 0.934 and 0.871, respectively, which shows significant agreement. We also ran Face++[6], a tool that infers gender automatically from images and we measured the agreement

[4]https://www.taskrabbit.com/
[5]https://www.figure-eight.com/
[6]https://www.faceplusplus.com/face-based-identification/

Table 14. Average and standard deviation of the number of taskers per partition across all queries per service.

Service	Average	STD
Run Errands	34.38	11.96
Event Staffing	28.44	13.20
General Cleaning	28.28	16.22
Furniture Assembly	26.80	16.19
Delivery	24.76	16.29
Moving	15.94	13.27
Handyman	15.64	13.58
Yard Work	10.00	10.98

between crowdsourced annotations and the results of Face++ using Kappa coefficient. The value fore gender is 0.954.

The gender and ethnic breakdowns of the taskers in our dataset are shown in Figures 7 and 8. Overall, we had a total of 2184 unique taskers in our crawled dataset, the majority of which were male ($\approx$ 74%) and white ($\approx$ 73%).

4.5.2 Results. We ran our algorithm EMDP on all 278 queries to quantify the unfairness of scoring for these queries. Since we do not have access to the scoring function through which the taskers are ranked, we assume an *opaque* scoring process setting and we utilize the taskers' ranks to build the score distributions. That is, for a given query q and a tasker w, we set the score $f(w, q)$ of tasker w with respect to the query q as follows:

$$f(w, q) = 1 - \frac{rank(w, q)}{|q|}$$

where $rank(w, q)$ is the rank of tasker w for query q and $|q|$ is the number of taskers returned per query (maximum of 50).

We then partition the taskers using our algorithm EMDP to identify the most unfair partitioning of taskers for each query q that we crawled. Since we only had two protected attributes in our dataset, EMDP resulted in three different types of partitionings, one based on ethnicity only, one based on gender only and one that used both attributes (i.e., a full partitioning).

Figure 9 displays the number of queries for which the most unfair partitioning identified by our algorithm EMDP was based on ethnicity only, on gender only or on both attributes. As can be seen in the figure, for the majority of the queries (112 out of 278), our algorithm EMDP partitioned the taskers only on ethnicity indicating that it was the attribute for which most unfairness in scoring occurred. In other words, for those 112 queries, the highest disparity in scoring occurred between whites, black and Asians. This was followed by gender in 89 out of the 278 queries in our crawled dataset. Finally, in 77 queries out of the 278 we had, the highest unfairness unveiled by our EMDP algorithm was when we split the taskers on both ethnicity and gender. The average number of taskers per partition returned by EMDP across all queries is 7.28 and its standard deviation is 9.28. Table 14 shows the average and standard deviation of the number of taskers per partition across all queries per service.

In Figure 10, we show the number of queries per ethnic group. Recall that out of the 278 queries in our dataset, the most unfair partitioning for 112 of those was achieved when partitioning the taskers using ethnicity only. As can be seen from Figure 10, all of these queries consisted of white taskers in the top 50 rankings, whereas only 103 of those consisted of black taskers, and only 38

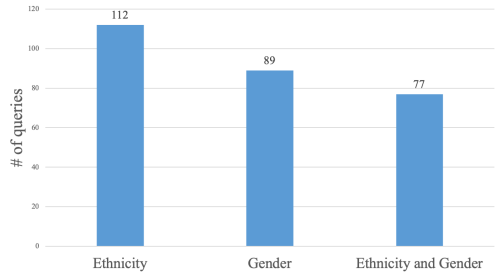


Fig. 9. Number of queries for different partitionings

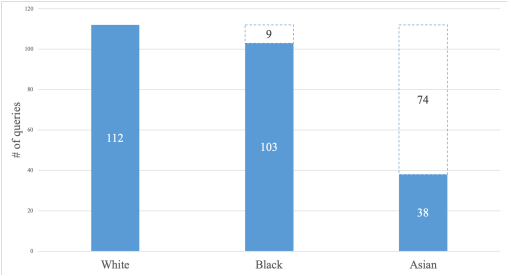


Fig. 10. Number of queries per ethnic group

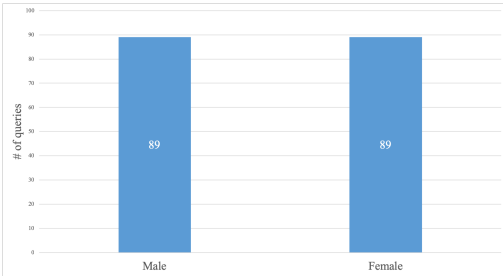


Fig. 11. Number of queries per gender group

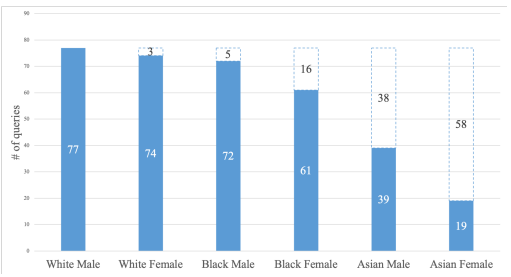


Fig. 12. Number of queries per ethnic-gender group

queries consisted of Asian taskers. This demonstrates that on TaskRabbit, black taskers are not included at all in the top-50 rankings for some services in certain locations. This becomes even more severe in the case of Asians, where for the majority of the queries (74 out of 112), no Asian taskers were included at all in the top-50 rankings. This conforms with the general distribution of taskers overall queries, where 73% of taskers were white, 23% were black and 4% were Asians. When breaking down the taskers for the 112 queries partitioned on ethnicity, the distribution of taskers per ethnicity and per gender followed closely that of the whole dataset. This indicates that the majority of taskers ranked in the top-50 were males and whites, compared to other ethnic and gender groups.

When looking at the number of queries per gender group (Figure 11), we observe that for the 89 queries for which the most unfairness occurred based on gender as revealed by our algorithm, all such queries consisted of both male and female taskers. This means that unlike ethnicity, all 89 queries included both male and female taskers in the top 50 rankings. Nonetheless, since our algorithm partitioned the taskers on only gender, this indicates that the rank distributions between the male and the female taskers were significantly different as measured by the average EMD. By breaking down those taskers in the top-50 rankings for those 89 queries, we observe that 83% of the top-ranked taskers were male compared to only 17% female taskers. This is a deviation from the overall distribution of taskers per gender in the whole dataset. Our algorithm was thus successful in flagging queries for which significantly fewer number of female taskers were in the top-rankings compared to male taskers. On the other hand, when breaking down the taskers based on ethnicity for those flagged 89 queries, they only marginally differed from the overall distribution per ethnicity group as in the whole dataset, where the majority of the taskers were white (75%), followed by blacks (23%) and finally Asians (only 2%).

Table 15. Average EMD and standard deviation over all queries for different types of partitioning and the two top cities and services for each group of queries

Attribute	Average EMD ± STD	Top 2 Cities	Top 2 Services
Ethnicity	0.341 ± 0.109	Memphis, TN, Indianapolis, IN	Handyman, Moving
Gender	0.317 ± 0.109	Minneapolis, MN, Columbus, OH	Handyman, Moving
Ethnicity-Gender	0.318 ± 0.096	San Francisco Bay Area, CA , Baltimore, MD	Handyman, Delivery

Table 16. Average pairwise EMD between each ethnic group and all other ethnic groups

Group	Average Pairwise EMD
Asian	0.337
Black	0.285
White	0.278

Finally, when looking at the queries for which our algorithm partitioned the taskers on both ethnicity and gender (77 queries) as shown in Figure 12, we observe that all those queries included white males in the top rankings, and almost all, with the exception of three queries included female white taskers. On the other hand, five queries did not include any black male taskers compared to 16 which did not include any black females. Finally, only 39 queries (i.e., almost half) included Asian male taskers in the top rankings and only 19 queries included Asian female taskers in the top rankings. When breaking down the taskers for the 77 queries, we observe that the distribution of taskers per gender group and ethnic group closely followed that of the overall distribution of taskers in the whole dataset.

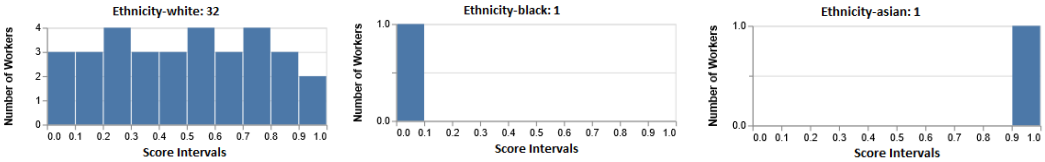
Table 15 shows the average EMD and its standard deviation over all queries for the different types of partitioning by EMDP (i.e., based on ethnicity alone, on gender alone or on both). It also displays the top cities and services among each group of queries for each case. As can be seen from the table, all three types of partitionings had relatively similar amount of unfairness as measured by the average EMD between their partitions. On the other hand, the service Handyman seemed to be the one which is most unfair on the basis of ethnicity, gender, or both. Moving seemed to be also one of the most unfair services when it comes to ethnicity alone or gender alone, whereas delivery was the second most unfair service on the basis of both protected attributes. When it comes to locations, Memphis, TN and Indianapolis, IN were the two most unfair locations on the basis of ethnicity, Minneapolis, MN and Columbus, OH for gender, and San Francisco Bay Area, CA and Baltimore, MD for both protected attributes.

Table 16 displays the average pairwise EMD values between each ethnic group and all other ethnic groups for the 112 queries that were partitioned on ethnicity only. As can be seen from the table, Asian taskers have the highest average pairwise EMD with all other ethnic groups, followed by blacks and whites. This indicates that for those queries that contained Asian taskers in the top rankings, the score distributions between those Asian taskers and all other taskers were significantly different. In other words, the ranking tend to either favor or disfavor Asians compared to all other ethnic groups. On the other hand, for those queries that contained white taskers with other types of taskers, the average pairwise EMD was less than that of Asians or blacks, indicating that overall white taskers were not particularly treated unfairly compared to other ethnic groups.

Table 17 shows the average pairwise EMD for each ethnic-gender group with respect to all other groups based on gender and ethnicity. As can be seen from the table, Asian males tend to have the highest average pairwise EMD followed by black females followed by Asian females followed by

Table 17. Average pairwise EMD between ethnic-gender group and all other ethnic-gender groups

Group	Average Pairwise EMD
Asian Male	0.321
Black Female	0.320
Asian Female	0.297
Black Male	0.277
White Female	0.277
White Male	0.248



(a) (b) (c)

Fig. 13. Histograms for the query with the highest Average EMD

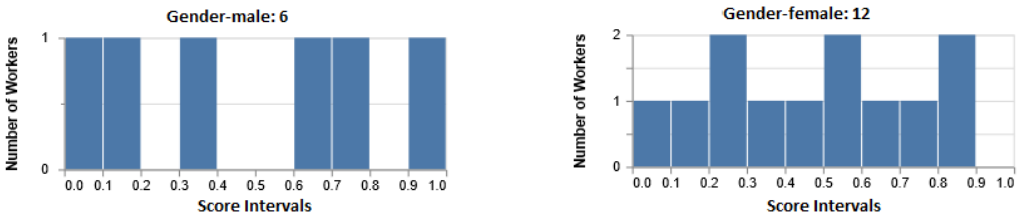
black males followed by white females and finally white males. This is inline with findings in a similar study on ranking in TaskRabbit [9], where the authors found the following correlations between demographics of taskers and their ranks in queries:

- Black taskers tend to be ranked lower compared to white taskers.
- Asian taskers tend to be ranked higher regardless of their gender.
- White women and black men tend to be ranked lower.

To summarize, our algorithm EMDP revealed that out of the 278 queries in our TaskRabbit dataset, the majority of these queries were considered unfair to taskers based on their ethnicity, followed by gender, then followed by both protected attributes. Upon further investigation of each group of queries separately, it turns out that white workers are the most favored in the top rankings, compared to blacks and Asians, and that males are also favored compared to females.

Figure 13 displays the histogram of scores for the query with the highest Average EMD value (0.600) as measured by our EMDP algorithm. The query was *Run Errands* in *Seattle, WA* and it had a total of 34 taskers (32 whites, 1 Asian, 1 black). The algorithm split those 34 taskers based on ethnicity only. Recall that the higher the score value, the higher the rank of the tasker for the query. We observe that since most of the taskers for that query were white, their scores and hence ranks are distributed uniformly across the whole range. That is, we observe that this query had white workers in almost all ranks. On the other hand, the only Asian tasker was ranked in the top 10% while the only black tasker was ranked in the bottom 10%. This is again an illustrative example that Asian taskers tend to be ranked higher, whereas black taskers tend to be ranked lower.

Figure 14 displays the histograms of scores for the query with the lowest Average EMD value (0.075) as measured by EMDP. The query was *Deep Cleaning* in *Portland, OR* and it had a total of 18 taskers. Upon investigating all the taskers returned for this query, we noticed that the all taskers were white, and that the algorithm split them based on gender only. As can be seen from the figure, 12 of the taskers were females compared to only 6 males. However, in both cases, the



(a) (b)

Fig. 14. Histograms for the query with the lowest Average EMD

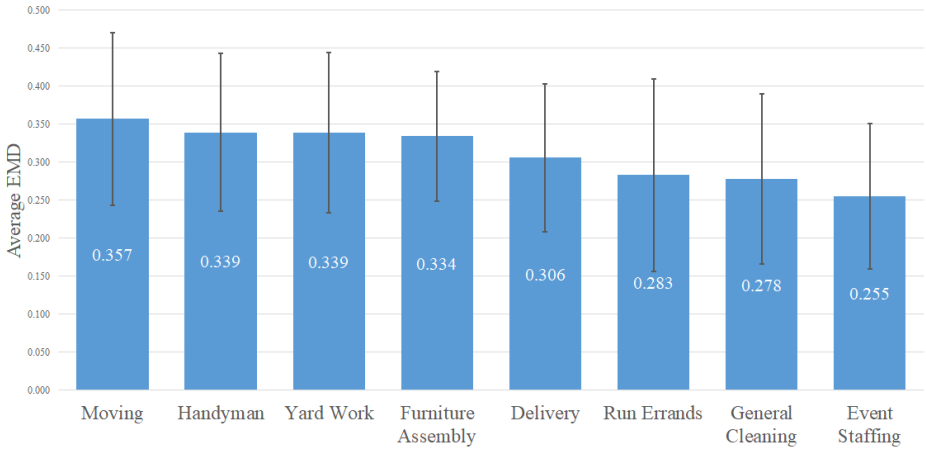


Fig. 15. Average EMD across all queries per service

taskers were spread across the rankings regardless of their gender. That is, female and male taskers were interchanged as they were ranked for the query.

Finally, Figures 15 and 16 display the average EMD and standard deviation across all queries per service and location, respectively. As can be seen from Figure 15, the services which were deemed the most unfair by our algorithm were Moving, Handyman and Yard Work and those deemed least unfair were Run Errands, General Cleaning and Event Staffing. When it comes to locations, the most unfair locations identified by our algorithm were San Antonio TX, Louisville KY, and Cleveland OH, and the least unfair locations were Chicago IL, Atlanta GA, and Phoenix AZ. Such information can be for instance used by the platform developers or by an auditor to investigate further and intervene to reduce unfairness for certain services or locations by adjusting the ranking algorithm used to rank taskers.

5 RELATED WORK

5.1 Algorithmic Fairness

Fairness has been trending in research for the last few years as we increasingly rely on algorithms for decision making. Bias has been identified as a major risk in algorithmic decision making [22]. One algorithmic solution is based on the formalization in [22] to quantify unfairness. Various definitions of unfairness exist [5, 6]. Examples of unfairness were studied in [23]. To detect unfairness in

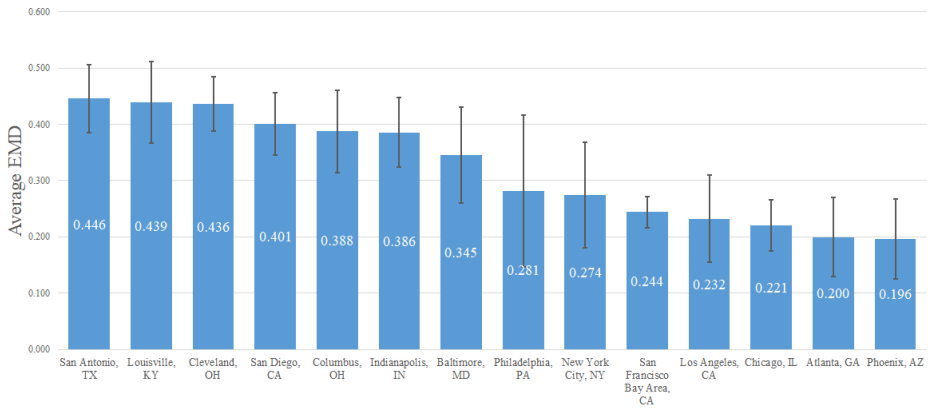


Fig. 16. Average EMD across all queries per location

algorithms, a framework [24] for "unwarranted associations" was designed to identify associations between a protected attribute, such as a person's race, and the algorithmic output using the FairTest tool. *In FairTest, these associations are typically assumed to be on a single-attribute level, which makes it different from our work where the goal is to quantify the relationship between a scoring process and multiple protected attributes.*

Transparency has been identified as a key requirement for informed decision-making. The authors in [25] propose four reference models which are meant to form a holistic conceptual baseline for transparency requirements in information systems. In [7], the notion of unfairness was defined as a disparity in treatment between different groups of people based on their protected attributes (i.e., what is commonly referred to as *group unfairness*). In this context, to assess unfairness mathematically, one needs to compare distributions of decisions across different groups of people. In our work, we adapt the definition of unfairness in [7] and partition individuals into groups using their protected attributes to quantify unfairness. However, rather than trying to fix it, the goal of our work is to just *reveal* any unfairness by the scoring process, which in some cases might be *positive* discrimination [8] where certain disadvantaged individuals are favored based on their protected attributes. To partition individuals based on their protected attributes, we rely on greedy algorithms that mimic decision trees algorithms. However, many other techniques for partitioning also exist, one of which was stated in [26] where they cluster points by separating them in a way to minimize the maximum inter-cluster distance.

Most previous work on studying fairness for groups of individuals has focused on groups defined using one protected attribute at a time. In [3], the authors introduce *subgroup fairness* and formalize the problem of auditing and learning classifiers for a rich class of subgroups. Our work differs in many ways: we are interested in scoring individuals and not classifying them and we seek to quantify the highest unfairness of a scoring function used to rank those individuals.

5.2 Fairness in Online Marketplaces

There is a wealth of work that empirically assessed fairness in online markets such as crowdsourcing or freelancing platforms. For instance, the authors in [27] analyzes ten categories of design and policy choices through which platforms may make themselves more or less conducive to discrimination by users. In [9], the authors found evidence of bias in two prominent online freelance marketplace, TaskRabbit and Fiverr. Precisely, in both marketplaces, they found that gender and

race are significantly correlated with worker evaluations, which could harm the employment opportunities afforded to the workers on these platforms. The work in [28] studies the Uber platform to explore how bias may creep into evaluations of drivers through consumer-sourced rating systems. Finally, discrimination in Airbnb was studied in [29] and high evidence of discrimination against African American guests was reported.

In [30], the authors study ethics in crowd work in general. They analyze recent crowdsourcing literature and extract ethical issues by following the PAPA (privacy, accuracy, property, accessibility of information) concept, a well-established approach in information systems. The review focuses on the individual perspective of crowd workers, which addresses their working conditions and benefits.

Several discrimination scenarios in task qualification and algorithmic task assignment were defined in [31]. Discrimination in crowdsourcing can be defined for different processes. In this work, we focus on one process, namely task qualification which is assumed to be achieved through a scoring process of workers.

In [32], the authors suggest that in order to reduce unfairness in virtual marketplaces, two principles must be adapted: 1) platforms should track the composition of their population to shed light on groups being discriminated against; and 2) platforms should experiment on their algorithms and data-sets in a timely manner to check for discrimination. Discrimination and transparency might be highly correlated but their correlation has yet to be studied profoundly. In [31], transparency plug-ins are reviewed. Those plug-ins disclose computed information, from worker's performance to requester's ratings such as TurkBench [33], and Crowd-Workers [34]. Such plug-ins might be helpful in a more detailed study of the effect of transparency on fairness.

5.3 Fairness of Ranking

There is a wealth of work on addressing fairness of ranking in general (for example [35–38]). Unlike our work, the majority of these works that focus on group fairness either assume the presence of predefined groups based on protected attributes of users, or the presence of ranking constraints that bound the number of users per protected attribute value in the top-k ranking. On the other hand, the work in [39] focuses on addressing amortized individual fairness in a series of rankings. To the best of our knowledge, our work is the first that proposes a general algorithmic approach to quantify unfairness of scoring in online job marketplaces that can reveal such unfairness even when it is based on any combination of protected attributes and under various transparency settings.

6 SUMMARY AND FUTURE WORK

We set out to examine fairness of scoring in online job marketplaces. To do this, we defined an optimization problem to find a partitioning of the individuals being ranked based on their protected attributes that exhibits the highest unfairness by a given scoring function. We used the Earth Mover's Distance between score distributions as a measure of unfairness. Unlike previous work, we did not assume a pre-defined partitioning of individuals and instead proposed a greedy decision-tree-style algorithm that efficiently partitions the individuals without exploring the full space of partitionings. We evaluated our algorithm on various datasets simulated from Amazon Mechanical Turk and crawled from TaskRabbit and showed its effectiveness under various data and process transparency settings. We also ran a series of user studies on Prolific Academic to examine the relationship between transparency and scoring fairness from the perspective of individuals being ranked. Our findings from the user studies verify those from our experiments and are in line with previous theories related to fairness.

We are currently investigating other formulations and metrics for fairness instead of the Earth Mover's Distance proposed in this paper. We also aim to study ways of "repairing" unfairness or

bias by recalibrating individuals' scores in different partitions. We are currently experimenting with different semantics of bias repair in scoring. Moreover, we assumed in this manuscript that protected attributes are uncorrelated. In future work, we plan to relax this assumption and modify our algorithm to deal with groups of correlated attributes. We would also like to study the effect of compensation on data and process transparency settings using approaches from game theory. Finally, our approach is implemented as a stand-alone system, and we plan to use it in other case studies that deal with scoring of people in different contexts such as human-powered data acquisition and job ranking in search engines, to name a few.

7 ACKNOWLEDGMENT

This work is supported by the American University of Beirut Research Board (URB), award number 103603.

REFERENCES

[1] S. Hajian, F. Bonchi, C. Castillo, Algorithmic bias: From discrimination discovery to fairness-aware data mining, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 2125–2126 (2016).

[2] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, K. P. Gummadi, Fairness constraints: Mechanisms for fair classification, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, 2017, pp. 962–970 (2017).

[3] M. J. Kearns, S. Neel, A. Roth, Z. S. Wu, Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, 2018, pp. 2569–2577 (2018).

[4] A. Singh, T. Joachims, Fairness of exposure in rankings, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, 2018, pp. 2219–2228 (2018).

[5] T. Calders, S. Verwer, Three naive bayes approaches for discrimination-free classification, Data Mining and Knowledge Discovery 21 (2) (2010) 277–292 (Sep 2010). doi:10.1007/s10618-010-0190-x.
URL https://doi.org/10.1007/s10618-010-0190-x

[6] I. Zliobaite, A survey on measuring indirect discrimination in machine learning, CoRR abs/1511.00148 (2015).
URL http://arxiv.org/abs/1511.00148

[7] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, On the (im)possibility of fairness, CoRR abs/1609.07236 (2016).
URL http://arxiv.org/abs/1609.07236

[8] M. Noon, The shackled runner: time to rethink positive discrimination?, Work, Employment and Society 24 (4) (2010) 728–739 (2010).

[9] A. Hannak, C. Wagner, D. Garcia, A. Mislove, M. Strohmaier, C. Wilson, Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr, in: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW 2017, Portland, OR, USA, February 25 - March 1, 2017, 2017, pp. 1914–1933 (2017).

[10] O. Pele, M. Werman, Fast and robust earth mover's distances, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 460–467 (September 2009).

[11] L. D. Ross, T. M. Amabile, J. L. Steinmetz, Social roles, social control, and biases in social-perception processes., Journal of personality and social psychology 35 (7) (1977) 485 (1977).

[12] C. Dellarocas, Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior, in: Proceedings of the 2nd ACM conference on Electronic commerce, ACM, 2000, pp. 150–157 (2000).

[13] S. K. Murthy, Automatic construction of decision trees from data: A multi-disciplinary survey, Data mining and knowledge discovery 2 (4) (1998) 345–389 (1998).

[14] G. Bolton, J. Brandts, A. Ockenfels, Fair procedures: Evidence from games involving lotteries, Economic Journal 115 (506) (2005) 1054–1076 (2005).

[15] A. Ghizzawi, J. Marinescu, S. Elbassuoni, S. Amer-Yahia, G. Bisson, Fairank: An interactive system to explore fairness of ranking in online job marketplaces, in: EDBT, 2019 (2019).

[16] S. Elbassuoni, S. Amer-Yahia, A. Ghizzawi, C. El Atie, Exploring fairness of ranking in online job marketplaces, in: EDBT, 2019 (2019).

[17] SurveyMonkey, Calculating the number of respondents you need,
https://help.surveymonkey.com/articles/en_US/kb/How-many-respondents-do-I-need.

[18] J. Sethuraman, C.-P. Teo, L. Qian, Many-to-one stable matching: Geometry and fairness, Math. Oper. Res. 31 (3) (2006) 581–596 (Aug. 2006).
[19] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (05) (2002) 557–570 (2002).
[20] N. Stewart, C. Ungemach, A. J. Harris, D. M. Bartels, B. R. Newell, G. Paolacci, J. Chandler, et al., The average laboratory samples a population of 7,300 amazon mechanical turk workers, Judgment and Decision making 10 (5) (2015) 479–491 (2015).
[21] P. Samarati, Protecting respondents identities in microdata release, IEEE transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027 (2001).
[22] K. Kirkpatrick, Battling algorithmic bias: how do we ensure algorithms treat us fairly?, Commun. ACM 59 (2016) 16–17 (2016).
[23] L. Sweeney, Discrimination in online ad delivery, CoRR abs/1301.6822 (2013).
URL http://arxiv.org/abs/1301.6822
[24] F. Tramèr, V. Atlidakis, R. Geambasu, D. J. Hsu, J. Hubaux, M. Humbert, A. Juels, H. Lin, Discovering unwarranted associations in data-driven applications with the fairtest testing toolkit, CoRR abs/1510.02377 (2015).
URL http://arxiv.org/abs/1510.02377
[25] M. Hosseini, A. Shahri, K. Phalp, R. Ali, Four reference models for transparency requirements in information systems, Requirements Engineering (2017) 1–25 (2017).
[26] T. F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoretical Computer Science 38 (1985) 293–306 (1985). doi:10.1016/0304-3975(85)90224-5.
[27] K. Levy, S. Barocas, Designing against discrimination in online markets, Berkeley Tech. LJ 32 (2017) 1183 (2017).
[28] A. Rosenblat, K. E. Levy, S. Barocas, T. Hwang, Discriminating tastes: Uber's customer ratings as vehicles for workplace discrimination, Policy & Internet 9 (3) (2017) 256–279 (2017).
[29] B. Edelman, M. Luca, D. Svirsky, Racial discrimination in the sharing economy: Evidence from a field experiment, American Economic Journal: Applied Economics 9 (2) (2017) 1–22 (2017).
[30] D. Durward, I. Blohm, J. M. Leimeister, Is there papa in crowd work?: A literature review on ethical dimensions in crowdsourcing, in: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences, IEEE, 2016, pp. 823–832 (2016).
[31] R. M. Borromeo, T. Laurent, M. Toyama, S. Amer-Yahia, Fairness and transparency in crowdsourcing, in: Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017., 2017, pp. 466–469 (2017). doi:10.5441/002/edbt.2017.46.
URL https://doi.org/10.5441/002/edbt.2017.46
[32] M. Luca, R. Fisman, Fixing discrimination in online marketplaces, Harvard Business Review (Dec 2016).
URL https://hbr.org/product/fixing-discrimination-in-online-marketplaces/R1612G-PDF-ENG
[33] B. V. Hanrahan, J. K. Willamowski, S. Swaminathan, D. B. Martin, Turkbench: Rendering the market for turkers., in: B. Begole, J. Kim, K. Inkpen, W. Woo (Eds.), CHI, ACM, 2015, pp. 1613–1616 (2015).
URL http://dblp.uni-trier.de/db/conf/chi/chi2015.html#HanrahanWSM15
[34] C. Callison-Burch, Crowd-workers: Aggregating information across turkers to help them find higher paying work, in: The Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP-2014), 2014 (November 2014).
URL http://cis.upenn.edu/~ccb/publications/crowd-workers.pdf
[35] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, R. Baeza-Yates, Fa* ir: A fair top-k ranking algorithm, in: CIKM, 2017, pp. 1569–1578 (2017).
[36] L. E. Celis, D. Straszak, N. K. Vishnoi, Ranking with fairness constraints, arXiv preprint arXiv:1704.06840 (2017).
[37] K. Yang, J. Stoyanovich, Measuring fairness in ranked outputs, in: SSDM, 2017, p. 22 (2017).
[38] A. Singh, T. Joachims, Fairness of exposure in rankings, arXiv preprint arXiv:1802.07281 (2018).
[39] A. J. Biega, K. P. Gummadi, G. Weikum, Equity of attention: Amortizing individual fairness in rankings, arXiv preprint arXiv:1805.01788 (2018).