IE529 Project: Fair Algorithms for Clustering

Rashid Anzoom, Akash Govind Kuttikad, Thihan Moe Kyaw

December 2021

1 Introduction

1.1 Paper Information

For this project, we have chosen to study the paper titled **Fair Algortihms for Clustering** authored by Suman K Bera, Deeparnab Chakrabarty, Nicolas J. Flores and Maryam Negahbani. It was published at the Advances in Neural Information Processing Systems 32 (NeurIPS 2019).

1.2 Motivation

Computers have been an indispensable part in all businesses for the past few decades and is expected to retain this position in the foreseeable future. Recent advancements in machine learning has allowed private and public organizations save time and money. Tasks that were impossible decades ago are now achievable due to these advancements. However, research on machine learning algorithms has heavily leaned toward achieving optimal efficiency, often ignoring the biases associated with the output. As a result, widespread application of these algorithms as a black-box model has resulted in outputs that has failed to maintain fairness across different groups. One significant example is the alleged racism inherent in different automation processes. In U.S hospitals, only 17.7% of the total U.S. patients referred for extra care were black whereas the actual number should be closer to 46% [Ledford, 2019]. Another glaring example is the difference in mortgage approval between majortiy and minority groups. Heaven blamed the disparate data availability for creation of such bias. These examples indicate that there is need for research in creation and modification of algorithms for obtaining a more fair solution. Such realization created a surge in research at this direction over the last five years [Kearns, 2017, Wilson et al., 2021, Lum and Johndrow, 2016, Agarwal et al., 2019]. One of the problems that they have tried to incorporate fairness in is the clustering problem.

The primary purpose of the paper we are reviewing is to design an adaptive algorithm for the fair clustering problem. This work is primarily motivated by previous works by Chierichetti et al. [2018] and Rösner and Schmidt [2018]. The former first brought forward the issue of fairness in clustering, while the second one provided generalization of the concepts. However, the literature lacked several shortcomings, including lack of flexible constraints, non-overlap rule for groups, and limited applicability,i.e., algorithm is specific to k-center problem. The motivation of the authors to alleviate these issues eventually led to the creation of this work.

In real life, clustering may cause bias towards many marginalized groups (not necessarily). For example, while clustering a set of population into different groups corresponding to credit scores, one might put too many/less people from a particular community to a single group/cluster. However, this might hinder/accelerate their progress. To avoid that, we need a mean to ensure that the clustering is fair to all categories. Chierichetti et al. [2018]

As the world's use on machine learning grows exponentially, clustering techniques will be used exponentially too. As the technology becomes increasingly accessible and streamlined for more people to be used, there might be more carelessness present in handling the input data. Such carelessness can lead to unintentional discrimination towards one group over another. A way to prevent this is by implementing additional steps in the already existing algorithms. The paper provides a foundation for the users of clustering algorithms to assign minimum ratios of each protected groups. This can make the data cleaning process a lot less time consuming and provides a safeguard for potential biases in the data cleaning process.

1.3 Outline

The outline of this report is as follows. In section 2, we discuss the technical concepts and terminologies required to understand the paper. In section 3, we discuss the solution concept of the algorithm and its theoretical result. In section 4. we demonstrate the application of the algorithm to a particular database and analyze its results. Finally, we conclude our discussion with summarizing remarks.

2 Technical Background

The paper aims for a fair clustering assignment of individual groups based on the disparate impact doctrine.

They give a much more generalized and tunable notion of fairness in clustering than that in [19, 50]. Our main result is that any solution for a wide suite of vanilla clustering objectives can be transformed into fair solutions in our notion with only a slight loss in quality by a simple algorithm. Here, the authors are interested in conducting clustering in a manner that is fair to different groups. The model that they propose is motivated by the disparate impact (DI) doctrine, which posits that any 'protected' class must have approximately equal representation in a clustering scenario. Although the DI doctrine is a law in the United States, violating the doctrine is nit illegal unless the violation cannot be justified by the decision maker. Here, we claim a clustering solution to be fair if it satisfies two conditions: restricted dominance (RD), that ensures the proportion of any group i is no more than α_i, and minority protection (MP) that ensures the proportion of any group i is not less than β_i.

The flexibility of our model comes from the fact that we allow α_i and β_i to be user specified parameters for each group i, and we also allow overlaps between different groups, i.e. one person can belong to more than one protected group.

2.1 Definitions

In this section, we define the variables/notations used in this report, the distance metrics used for clustering, and we also provide the mathematical definitions of the clustering techniques that we employ.

Let C be the set of all v points with dimension d that we want to cluster in a metric space χ. Let F $\subseteq \chi$ be the set of possible k cluster centers. It is important to note that F and C need not necessarily be disjoint, and F could be equal to C as well. For a set S $\subseteq \chi$ and for a point $v \in \chi$, we use d(v, S) to denote the minimum distance between a set S and a point v, that is $\min_{y \in S} d(v, y)$.

Vanilla (k, p)- Clustering problem:

Given a metric space (χ, d) and an non-negative integer k, the objective of the Vanilla (k,p) Clustering problem is to choose a subset S $\subseteq$ F such that $|S| = k$ and find an assignment ϕ: $C \rightarrow S$ such that the objective function L_p(S,ϕ) = $[\sum$v $\in C$ d(v, ϕ(v))p$]^{\frac{1}{p}}$ is minimized. In this vanilla version of the problem, there are no fairness notions being considered and hence every point v $\in$ C will be assigned to its nearest center, denoted by ϕ(v), as determined by the L$_p$ norm specified in the objective. As p = $\{1, 2, \infty\}$, we obtain the k-medians, k-means and k-centers problems respectively. We shall denote the optimal value of the Vanilla (k,p) Clustering problem for any instance I by $OPT_{vnll}(I)$

Fair (k, p)- Clustering problem:

In the fair version of the clustering problem, we have additionally been provided with l groups of C, given by C$_1$, C$_2$, C$_3$...C$_l$. It is to be noted that the groups need not be disjoint, and we can allow a single point $v \in$ C to be a part of more than one group, and we denote Δ as maximum number of groups a point can be a part of. In this fair version of the problem, in addition to choosing a subset S $\subseteq$ F such that $|S| = k$ and minimizing the cost of an assignment ϕ: $C \rightarrow S$ denoted by L$_p$(S,ϕ), we also have to satisfy the following fairness constraints:

$$|\{v \in C_i : \phi(v) = f\}| \leq \alpha_i |\{v \in C : \phi(v) = f\}|$$

$$|\{v \in C_i : \phi(v) = f\}| \geq \beta_i |\{v \in C : \phi(v) = f\}|$$

The assignment defined by ϕ clusters a set of points v to a particular center given by $\phi(v) = f$. As explained in the problem definition, eq. (MP) denotes the minority protection property which imposes a lower bound of a group's representation (C_i) in any cluster centered around $\phi(v)$. This protects us against under-representation of a particular group. Eq. (RD) denotes the restricted dominance property which imposes an upper bound of a group's representation (C_i) in any cluster, which protects against domination of a particular group in that cluster. Due to the above constraints, we can no longer assume that $\phi(v)$ is the 'nearest' center to any point v in C. We will henceforth use (S, ϕ) to denote a fair clustering scenario.

3 Summary of Results

3.1 Solution Technique

The problem stated in the previous section is solved in several steps.Concisely, the solution procedure consists of two steps: vanilla clustering and fair assignment. Initially, we use any vanilla (k, p) clustering algorithm on the given data set C. The purpose of this step is to establish the set of centers S. Now, with these centers and other parameters (including fairness notions), we solve an LP (LP1).Next, we define a linear programming problem to reassign the points around the center. It is formulated as:

$$min \sum_{v \in C, f \in S} d(v, f)^p x_{v,f}$$

s.t.

$$\beta_i \sum_{v \in C} x_{v,f} \leq \sum_{v \in C_i} x_{v,f} \leq \alpha_i \sum_{v \in C} x_{v,f} \; \forall f \in S, i \in [l]$$

$$\sum_{f\in S} x_{v,f} = 1 \;\forall v \in C$$

$$0 \leq x_{v,f} \leq 1$$

This is in fact a relaxed version of the actual assignment problem. By checking the optimal solution, we finalize the assignment of points v to centers f for which $x^*_{v,f} = 1$. We further remove them from the set of points to be clustered C as well as their groups C_i. Next, we define two variables:

$$T_f = \sum_{v\in C} x^*_{v,f} \;\forall f \in S$$

$$T_{f,i} = \sum_{v\in C_i} x^*_{v,f} \;\forall f \in S, i \in [l]$$

We now construct another LP (LP2) with decision variables $x*_{v,f} > 0$. The formulation is:

$$min \sum_{v\in C, f\in S} d(v,f)^p x_{v,f}$$

s.t.

$$\lfloor T_f \rfloor \leq \sum_{v\in C} x_{v,f} \leq \lceil T_f \rceil \;\forall f \in S$$

$$\lfloor T_{f,i} \rfloor \leq \sum_{v\in C_i} x_{v,f} \leq \lceil T_{f,i} \rceil \;\forall f \in S, i \in [l]$$

$$\sum_{f\in S} x_{v,f} = 1 \;\forall v \in C$$

$$0 \leq x_{v,f} \leq 1$$

This LP is run in a loop until all points have been assigned to a center. During each iteration after obtaining solution, the following checks are made:

- If any optimal solution $x^*_{v,f} = 1$ we assign point v to center f and remove it from the set of unassinged points. We also reduce T_f and $T_{f,i}$ by 1 for each such point.
- If any optimal solution $x^*_{v,f} = 0$, we remove it from the set of decision variables.
- If $|x^*_{v,f} : 0 < x^*_{v,f} < 1, v \in C_i| < 2(\Delta + 1)$, we remove the corresponding constraint for that $f \in S, i \in [l]$.
- If $|x^*_{v,f} : 0 < x^*_{v,f} < 1, v \in C| < 2(\Delta + 1)$, we remove the corresponding constraint for that $f \in S$.

The whole procedure is represented graphically in the flow chart. In case, $p = \infty$, the objective functions in the LPs might not be meaningful.In that case, they suggest starting with an initial guess G of the optimal value.; For all $d(v, f) > G$, we can set $x_{v,f} = 0$ and check if it satisfies the constraints. If not, then the guess is too small, and we increase it and iterate again.

3.2 Theoretical Result

The main theoretical result for the algorithm involves the correspondence between a vanilla and a fair clustering problem. For a ρ-approximate algorithm A for the Vanilla (k, p)-Clustering problem, a $(\rho + 2)$-approximate solution (S, Φ) with $\lambda = (4\Delta + 3)$-additive violation can be achieved for the FAIR (k, p)-clustering problem. By λ-additive violation, we indicate that the constraints in LP1 are satisfied within the upper bound $UB + \lambda$ and lower bound $LB - \lambda$.

$$\beta_i|\{v \in C : \phi(v) = f\}| - \lambda \leq |\{v \in C_i : \phi(v) = f\}| \leq \alpha_i|\{v \in C : \phi(v) = f\}| + \lambda$$

Here, Δ denotes the maximum number of overlapping groups. Essentially, they propose to get $O(1)$ factor approximation to the fair (k, p) clustering problem with $O(\Delta)$ additive violation for any l_p norm. However, in case of$\Delta = 1$, it does not work. Instead, it is considered as a special case, where the maximum violation was 3. To show this, they used the generalized assignment problem (GAP) rounding technique Shmoys and Tardos [1993].

3.3 Algorithm Flowchart

The following figure depicts the procedure we described above. For brevity, we have used the while loop involved in checking $x^*(v, f)$ only once. In other two case, we have just stated the term **componentwise**.

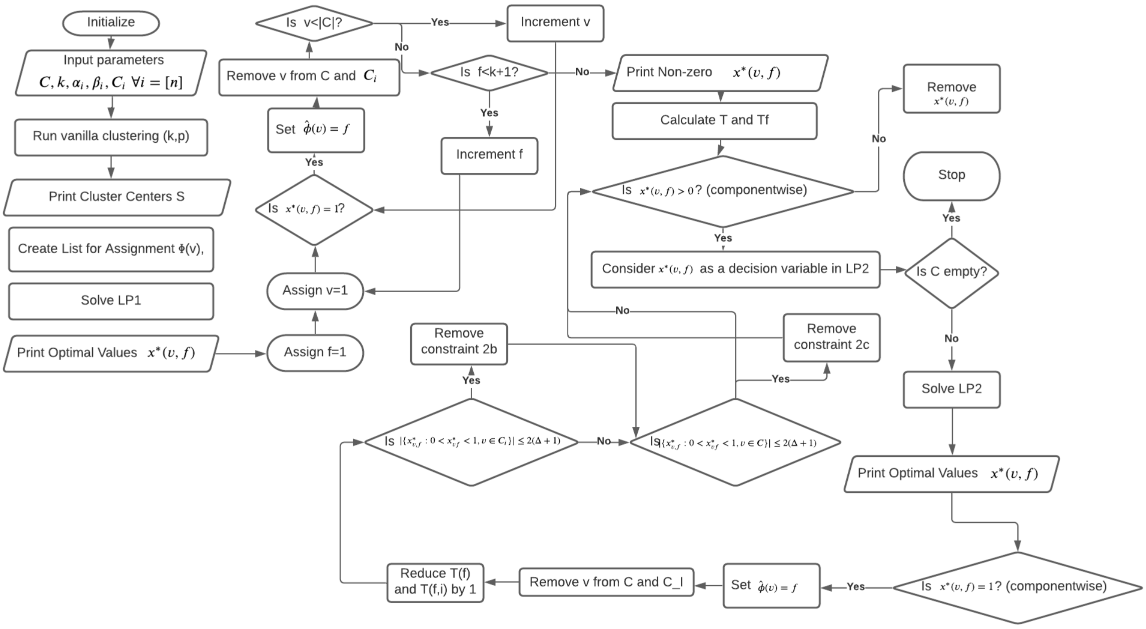
Initialize
Input parameters $C, k, \alpha_i, \beta_i, C_i\ \forall i = [n]$
Run vanilla clustering (k,p)
Print Cluster Centers S
Create List for Assignment Φ(v),
Solve LP1
Print Optimal Values $x^*(v, f)$
Assign f=1
Assign v=1
Is $x^*(v, f) = 1$?
Yes
Set $\hat{\phi}(v) = f$
Remove v from C and C_i
Is v<|C|?
Yes
No
Increment v
Is f<k+1?
Yes
Increment f
No
Print Non-zero $x^*(v, f)$
Calculate T and Tf
Is $x^*(v, f) > 0$? (componentwise)
No
Remove $x^*(v, f)$
Yes
Consider $x^*(v, f)$ as a decision variable in LP2
Is C empty?
Yes
Stop
No
Solve LP2
Print Optimal Values $x^*(v, f)$
Is $x^*(v, f) = 1$? (componentwise)
Yes
Set $\hat{\phi}(v) = f$
Remove v from C and C_I
Reduce T(f) and T(f,i) by 1
Is $|\{x^*_{v,f} : 0 < x^*_{vf} < 1, v \in C_i\}| \leq 2(\Delta + 1)$
Yes
Remove constraint 2b
No
Is$|\{x^*_{v,f} : 0 < x^*_{vf} < 1, v \in C\}| \leq 2(\Delta + 1)$
Yes
Remove constraint 2c
No

Figure 1: Flow Chart of the (k, p)-fair clustering algorithm

4 Implementation

To demonstrate the efficacy of the algorithm, we decided to implement it to a particular dataset (to be described in later subsections). The coding was done in Python, while the optimization was achieved by implementing Gurobipy. As for vanilla clustering, we used the k-means package from scikit-learn. The choice of these programs or functions were motivated by our familiarity with them. The authors also coded in Python, however they used CPLEX for solving the linear programs. The code used for replicating the algorithm is avaialble at Appendix.

4.1 Data and Application

The paper uses four data-sets: records of phone calls from a bank, diabetes - extracted from diabetic patient records, a credit card holder information database from Taiwan, and 1994 U.S. census. We employ the first data set, 'bank' in this work. It has 4,521 points in the data-set, consisting of records of phone calls from a marketing campaign by a Portuguese banking institution. It consists of 17 attributes; however, for clustering, we only use three attributes: age, balance and duration. We choose two more sensitive attributes- marital and default as our protected groups, to ensure fair participation from these groups according to our MP and RD constraints. The attribute marital has three different groups: clients who are single, married or divorced and the attribute default has has two groups: yes or no. We encode all text variables to numeric values, so as to cluster them in the euclidean space. We choose k-means++ as our vanilla clustering approach, although one could go for k-medians or k-centers as well. For different values of k, we compare the performance between (k,p)-vanilla clustering and (k,p)-fair clustering via two metrics: balance and cost-of-fairness. They are defined in the next section.

4.2 Result Statistics

In the paper, the authors' experiments showed that on established data sets, the proposed algorithm performs much better in practice than in theory. However, in this report, we shall be reporting the result of our own experiments. First, let us define the two metrics that were mentioned earlier. The first one is the cost of fairness (COF), is defined as the ratio of the objective value of the fair clustering over that obtained in the vanilla (k-means++) clustering. In our experiments, the COF was found be just above 1 for all used values of k (See Figure 2). In fact, the values of COF typically lie between 1.01 and 1.07, So, it is evident that the switching to fair clustering approach from the traditional one is not very expensive. We also notice that the cost of clustering decreases with increase in number of clusters, and one has to decide the trade off between number of cluster centers to be assigned/opened and the expense of making the assignment.

The second defining metric, balance of each cluster center, is a measure of unfairness in each cluster [Bera et al., 2019]. To determine the balance, we define two intermediate values r_i, the proportion of each group i in the data-set, and $r_i(f)$, the proportion of group i in cluster f. Mathematically, we can express

$$r_i = \frac{|C_i|}{|C|}$$

and

$$r_i(f) = \frac{|C_i(f)|}{|C(f)|}$$

Using these values, we express the balance as

$$balance(f) = min\{r_i/r_i(f), r_i(f)/r_i\}$$

In this scenario, we have the maximum number of groups a client could belong to, $\Delta = 2$. The values for balance for each of the k centers have been plotted in Fig. 2, and is typically high for majority of the clusters. Although we did prove the existence of a λ which is at most $4\Delta + 3$ in the beginning, we eventually observe that these violations have not crossed values of 0.9 for the scenario with $k = 4$, whereas vanilla k-means can result in values which are unfair by a large margin.

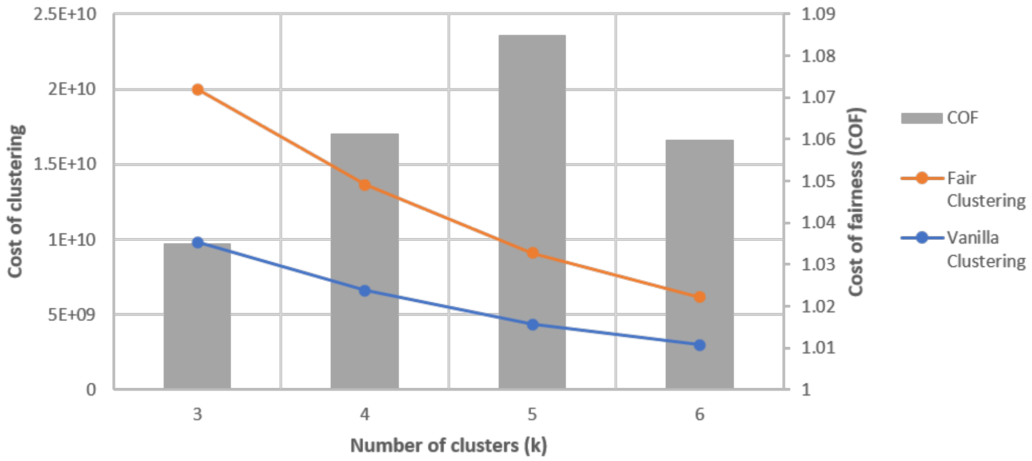

Figure 2: Cost of clustering and Cost of fairness plotted against number of clusters

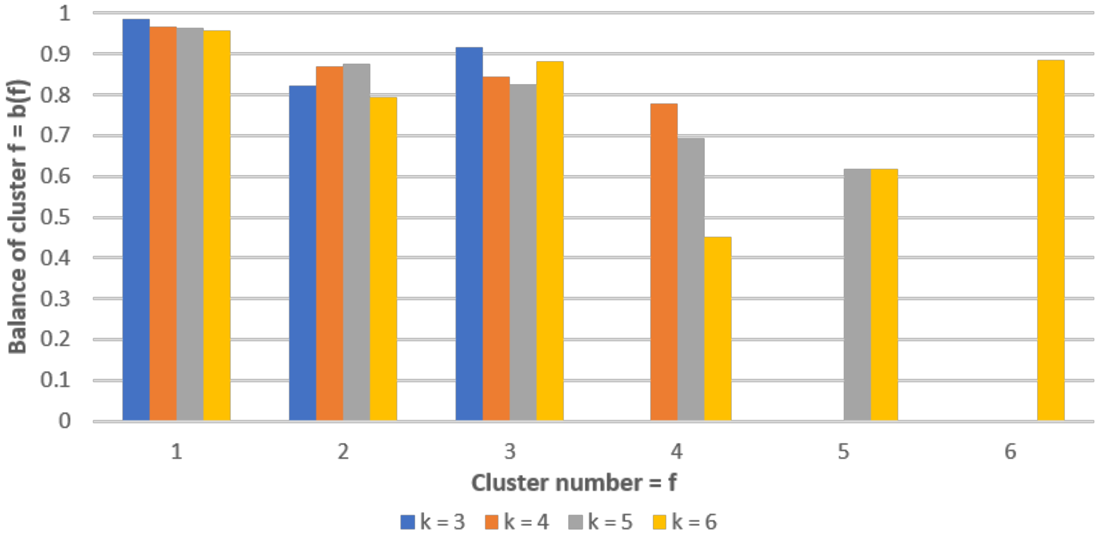

Figure 3: Values of balance for each cluster

4.3 Complexity Analysis

We can divide the entire algorithm employed in this work into three parts: (1) the vanilla (k, p) clustering to assign initial centers, (2) LP1 solved with relaxed integral constraints, and (3) LP2 solved iteratively until all points have been assigned to a center. For the first part, we have used k-means++ algorithm which runs in $O(n^{k+\frac{2}{p}})$ time in the worst case scenario (Pedregosa et al. [2011]). Here, n stands for the number of data points, k is number of cluster centers and p denotes the number of attributes chosen for clustering. Assuming that the linear programming formulation is solved in polynomial time through the ellipsoid method, the worst case complexity would be $O((nk)^4 log(nkU))$, where U is the largest element among the input matrices for the linear program. LP1 would be solved only once, whereas LP2 could be iterated at most n times, assuming only one variable $x^*_{v,f}$ is removed each time. Hence, the complexity of the entire algorithm is given as follows:

$$Complexity = O(n^{k+\frac{2}{p}} + (nk)^4 log(nkU) + n^2k^4 log(nkU))$$

5 Summary

The modern world is growing increasingly reliant on artificial intelligence for a wide range of applications. Currently, people and businesses are placing more emphasis on machine learning approaches to improve decision-making efficiency. However, in the pursuit of gaining global optima of quality, the fairness or bias in the result is often ignored. When the application area is societal problem, this issue is no longer trivial. Yet, it is surprising to see the negligence of addressing this issue for so long. Nevertheless, in recent years, research on algorithmic fairness has received significant traction. The paper under investigation is a proof of that. With their proposed fair clustering algorithm, the credibility of the results will increase significantly, specially in societal problem- solving. Thus, this paper offer significant contribution from both academic and practical perspective.

In terms of technical perspective, the algorithm is easily comprehensible as well as implementable. The ability to use any vanilla clustering technique for center selection gives it great flexibility. Furthermore, consideration of multiple overlapping groups and customized constraint limit makes it resemble the real-life situation. And both practical and theoretical results indicate a relatively small cost of fairness, which advocates for the usefulness of the algorithm. However, one issue that requires further work according to us is the identification of values of α and β for different notions of exact and approximate fairness (e.g., proportionality, envy-freeness). In fact, an interesting research area might be the investigation of whether different types of fair solutions exist in a clustering problem.

The course **IE529: Statistics of Big Data** introduced us to different analysis techniques for significantly large datasets. One of the major foci in the course was clustering techniques. We thoroughly enjoyed learning and practicing these techniques. However, the concept of fairness was not included in their objective. That is why we became curious of this paper and chose to review it. And since the algorithm included running any vanilla clustering problem, our understanding on clustering algorithm were further enhanced. In short, we found the paper very interesting and hope to work on it future at some point. We thank Dr. Beck for introducing us to the paper.

References

Heidi Ledford. Millions of black people affected by racial bias in health-care algorithms, Oct 2019. URL `https://www.nature.com/articles/d41586-019-03228-6`.

Will Douglas Heaven. Bias isn't the only problem with credit scores-and no, ai can't help, Jun 2021. URL shorturl.at/buLNO/.

Michael Kearns. Fair algorithms for machine learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 1–1, 2017.

Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. Building and auditing fair algorithms: A case study in candidate screening. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 666–677, 2021.

Kristian Lum and James Johndrow. A statistical framework for fair predictive algorithms. *arXiv preprint arXiv:1610.08077*, 2016.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

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

Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. *arXiv preprint arXiv:1802.02497*, 2018.

David B Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical programming*, 62(1):461–474, 1993.

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.