Efficient Algorithms for Fair Clustering with a New Notion of Fairness

Shivam Gupta[1*], Ganesh Ghalme[2], Narayanan C. Krishnan[1] and Shweta Jain[1]

[1*]Indian Institute of Technology (IIT) Ropar, India.
[2]Indian Institute of Technology (IIT) Hyderabad, Kandi, India.

*Corresponding author(s). E-mail(s): shivam.20csz0004@iitrpr.ac.in;
Contributing authors: ganeshghalme@ai.iith.ac.in; ckn@iitrpr.ac.in; shwetajain@iitrpr.ac.in;

Abstract

We revisit the problem of fair clustering, first introduced by Chierichetti et al. (2017), that requires each protected attribute to have approximately equal representation in every cluster; i.e., a Balance property. Existing solutions to fair clustering are either not scalable or do not achieve an optimal trade-off between clustering objective and fairness. In this paper, we propose a new notion of fairness, which we call $\boldsymbol{\tau}$-ratio fairness, that strictly generalizes the Balance property and enables a fine-grained efficiency vs. fairness trade-off. Furthermore, we show that simple greedy round-robin based algorithms achieve this trade-off efficiently. Under a more general setting of multi-valued protected attributes, we rigorously analyze the theoretical properties of the our algorithms. Our experimental results suggest that the proposed solution outperforms all the state-of-the-art algorithms and works exceptionally well even for a large number of clusters.

Keywords: Fairness, Clustering, Machine Learning, Unsupervised Learning

1 Introduction

Advances in machine learning research have resulted in the development of increasingly accurate models, leading to the wide adoption of these algorithms in applications ranging from self-driving cars, approving home loan applications, criminal risk prediction, college admissions, and health risk prediction. While improving the accuracy is the primary objective of these algorithms, their use to allocate social goods and opportunities such as access to healthcare and job and educational opportunities warrants a closer look at the societal impacts of their outcomes (Carey and Wu (2022); Ntoutsi et al. (2020)). Recent studies have exposed a discriminatory outlook in the outcomes of these algorithms leading to treatment disparity towards individuals belonging to marginalized groups based on gender and race in real-world applications like automated resume processing (Dastin, 2018), loan application screening, and criminal risk prediction (Julia et al., 2016). Designing fair and accurate machine learning models is thus an essential and immediate requirement for these algorithms to make a meaningful real-world impact.

While fairness in supervised learning is studied (Correa et al., 2021; Chikahara et al., 2021; Lee et al., 2021; Mehrabi et al., 2021; Le Quy et al., 2022; Dwork et al., 2012), the fairness in unsupervised learning is still in its formative stages (Deepak et al. (2020); Chhabra et al. (2021)). To emphasize the importance of fairness in unsupervised learning, we consider the following hypothetical scenario: An employee-friendly company is looking to open branches at multiple locations across the city and distribute its workforce in these branches to improve work efficiency and minimize overall travel time to work. The company has employees with diverse backgrounds based on, for instance, race and gender and does not prefer any group of employees over other groups based on these attributes. The company's diversity policy dictates hiring a minimum fraction of employees from each group in every branch. Thus, the natural question is: where should the branches be set up to maximize work efficiency, minimize travel time, and maintain diversity. In other words, the problem is to devise an unsupervised learning algorithm for identifying branch locations with the fairness (diversity) constraints applied to each branch. This problem can be naturally formulated as a clustering problem with additional fairness constraints on allocating the data points to the cluster centers. Clustering, along with classification, forms the core of powerful machine learning algorithms with significant societal impact through applications such as automated assessment of job suitability (Padmanabhan, 2020) and facial recognition (Li et al., 2020). These constraints arise naturally in applications where data points correspond to individuals, and cluster association signifies the partitioning of individuals based on features.

Typically, fairness in supervised learning is measured by the algorithm's performance over different groups based on protected(sensitive) attributes such as gender, race, and ethnicity. The first fairness notion for clustering was proposed by Chierichetti et al. (2017), wherein each cluster is required to exhibit a Balance; defined as the ratio of protected attribute and non-protected attribute in each cluster to the level of this ratio in the entire dataset. Their methodology— apart from having significant computational complexity—applies only to binary-valued protected attributes and does not allow for trade-offs between the clustering objective and fairness guarantees. The subsequent literature Backurs et al. (2019); Schmidt et al. (2019); Schmidt and Wargalla (2021); Huang et al. (2019) improve efficiency; however, do not facilitate explicit trade-off between the clustering objective cost and the fairness guarantee. In this paper we define a new notion of fairness which we call τ-ratio guarantee. To each cluster, a τ-ratio guarantee ensures a certain fraction of data points for a given protected attribute. We show that this simple notion of fairness has several advantages. First, the definition of τ-ratio naturally extends to multi-valued protected attributes; second τ-ratio fairness strictly generalizes the Balance property; third, it admits an intuitive and computationally efficient round-robin approach to fair allocation; and fourth, it is straightforward for the algorithm designer to input the requirement into the algorithm as constraints and easy to interpret and evaluate it from the output. In our running example, if a company wants to have minimum fraction of employees from each group in every branch (clusters) then one can simply specify this in the form of a vector τ of size equal to number of protected groups. Through rigorous theoretical analysis, we show that the proposed algorithm FRAC$_{OE}$ provides a $2(\alpha + 2)$-approximate guarantee on the objective cost with τ-ratio fairness guarantee up to three clusters. Here, α is the approximation factor achieved by the vanilla clustering algorithm. We further experimentally demonstrate that our approach can achieve better clustering objective costs than any state-of-the-art (SOTA) approach on real-world data sets, even for a large number of clusters. Overall, the following are the contributions of our work.

1.1 Our Contribution

Conceptual Contribution

We introduce a new notion of fairness which we call a τ-ratio guarantee and show that any algorithm satisfying a τ-ratio guarantee also satisfies the Balance property (Theorem 4). Also, we show that every parameter setting of Balance collapses to a degenerate value of τ-ratio fairness showing generalisation of proposed notion. We propose two simple and efficient round-robin-based algorithms for the τ-ratio fair allocation problem (see, Section 4). Our algorithms use the clustering algorithm as a black-box implementation and modify its output appropriately to ensure τ-ratio guarantee. The fairness guarantee is deterministic and verifiable, i.e., holds for every run of the algorithm, and can be verified from the outcome without explicit knowledge of the underlying clustering algorithm. The guarantee on objective cost, however, depends on the approximation guarantee of the clustering algorithm.

Our algorithms can handle multi-valued protected attributes, allow user-specified bounds on Balance, are computationally efficient, and incur only an additional time complexity of $O(kn\log(n))$, best in the current literature. Here, n is the size of the dataset, and k is the number of clusters.

Theoretical Contributions

We show theoretical guarantees for our first algorithm; FRAC$_{OE}$. First, we show that our algorithm achieves $2(\alpha + 2)$-approximate fairness for clustering instances upto three clusters (Theorem 7 and Lemma 11) with respect to optimal fair clustering cost for τ=$1/k$; here α is a clustering algorithm specific constant. That is, given a fair clustering instance with $k \leq 3$ clusters, n datapoints and a fairness vector τ, our proposed algorithm returns an allocation that has objective cost of $2(\alpha + 2)$ times the objective cost of optimal assignment that also satisfies the τ-ratio guarantee. We further show that this guarantee is tight (Proposition 12). For $k > 3$ clusters we show $2^{k-1}(\alpha + 2)$-approximation guarantee on the τ-ratio. We conjecture that the exponential dependence of the approximation guarantee on k can be reduced to a constant. The guarantees are extended to work for any general τ vector (see Section 5.2). We also theoretically analyse the convergence of FRAC$_{OE}$ (Lemma 14).

Experimental Contributions

Through extensive experiments on four datasets (Adult, Bank, Diabetes, and Census II), we show that the proposed algorithm outperforms all the existing algorithms on fairness and objective costs. Perhaps the most important insight from our experiments is that the performance of our proposed algorithms does not deteriorate with increasing k, experimentally validating our conjecture. We compare our algorithms with SOTA algorithms for their fairness guarantee, objective cost, and runtime analysis. We also note that our algorithms do not require hyper-parameter tuning, making our method easy to train and scalable. While our algorithms are applicable to center based clustering approach, we demonstrate its efficacy using k-means and k-median.

2 Related Work

While there is abundant literature on fairness in supervised learning (Chikahara et al. (2021); Gong et al. (2021); Zhang et al. (2021); Ranzato et al. (2021); Lohaus et al. (2020); Cho et al. (2020); Baumann and Rumberger (2018);), research on fair clustering is still in infancy and is rapidly gathering attention (Chierichetti et al. (2017); Kleindessner et al. (2019); Ziko et al. (2021); Liu and Vicente (2021); Davidson and Ravi (2020);Bercea et al. (2018); Chhabra et al. (2021)). These studies include extending the existing fairness notions such as group and individual fairness to clustering (Bera et al. (2019); Kleindessner et al. (2020); Chen et al. (2019a)), proposing new problem-specific fairness notions such as social fairness (Abbasi et al. (2021); Makarychev and Vakilian (2021)), characterizing the fairness and efficiency trade-off (Ziko et al. (2021); Abraham et al. (2020)) and developing and analyzing fair and efficient algorithms (Bandyapadhyay et al. (2020); Schmidt et al. (2019)).

The fairness in clustering is introduced at different stages of implementation namely – pre-processing, in-processing and post-processing.

Pre-processing: Following a disparate impact doctrine (Barocas and Selbst (2016)), Chierichetti et al. (2017), in their pioneering work, defines fairness in clustering through a Balance property. Balance is the ratio of data points with different protected attribute values in a cluster. A balanced clustering ensures Balance in all the clusters equal to the Balance in the original dataset (see Definition 2). Chierichetti et al. (2017) achieve balanced clustering through the partitioning of the data into balanced sets called fairlets, followed by merging of the partitions. Subsequently, Backurs et al. (2019) proposes an efficient algorithm to compute the fairlets. Both the approaches have two major drawbacks: they are limited to the datasets having only binary-valued protected attributes, and can only create clusters exhibiting the exact Balance present in the original dataset, thereby not being flexible in achieving an optimal trade-off between Balance and accuracy. Schmidt et al. (2019) extend the notion of coresets to fair clustering and provide an efficient and scalable algorithm using *composable* fair coresets (see also Huang et al. (2019); Schmidt and Wargalla (2021); Bandyapadhyay et al. (2020); Feng et al. (2021)). A coreset is a set of points approximating the optimal clustering objective value for any k cluster centers. Though the coreset construction can be performed in a single pass over the data as opposed to the fairlets construction, storing coresets takes exponential space in terms of the dimension of the dataset. Bandyapadhyay et al. (2020) though reduces this exponential size requirement to linear in terms of space; the algorithm still has the running complexity that is exponential in the number of clusters. Our proposed approach is efficient because we do not need any additional space. Simultaneously, the running complexity is linear in the number of clusters and near-linear in the number of data points.

In-processing: Böhm et al. (2020) propose an (α+2)-approximate algorithm for fair clustering using minimum cost-perfect matching algorithm. While the approach works with a multi-valued protected attribute, it has $O(n^3)$ time complexity and is not scalable. Ziko et al. (2021) propose a variational framework for fair clustering. Apart from being applicable on datasets with multi-valued protected attributes, the approach works for both prototype-based (k-mean/k-median) and graph-based clustering problems (N-cut or Ratio-cut). However, the sensitivity of the hyper-parameter to various datasets and the number of clusters necessitates extensive tuning rendering the approach computationally expensive. Further, the clustering objective also deteriorates significantly under strict fairness constraints when dealing with many clusters (refer Section 7.1). Along the same lines, Abraham et al. (2020) devise an optimization-based approach for fair clustering with multiple multi-valued protected attributes with a trade-off hyper-parameter similar to Ziko et al. (2021).

Post-processing: Bera et al. (2019) converted fair clustering into a fair assignment problem and formulated a linear programming (LP) based solution. The LP-based formulation leads to a higher execution time (refer to Section 7.4). Also, the approach fails to converge when dealing with a large number of clusters. The proposed approach takes a similar route as Bera et al. (2019) to convert the fair clustering problem into a fair allocation problem. However, we give a simple polynomial-time algorithm which, in $O(nk \log n)$ additional computations, guarantees a more general notion of fairness which we call τ-ratio fairness. Our allocation algorithms have following main advantages over the current state of the art;

1. they are computationally efficient,
2. they work for multi-valued protected attributes,
3. no hyperparameter tuning is required and,
4. they are simple and more interpretable (refer Section 3).

The work by Bera et al. (2019) is extended by Harb and Lam (2020) for k-center problem whereas we in present study consider k-means and k-median based centering techniques. Similarly the works by (Ahmadian et al. (2019); Jones et al. (2020); Bandyapadhyay et al. (2019); Jia et al. (2020); Anegg et al. (2020); Chakrabarti et al. (2022); Brubach et al. (2020)) are applicable only for k-center clustering. While we focus on the fairness notion of Balance based on the protected attribute value, other perspectives on fairness are defined in the literature. Kleindessner et al. (2020) define individual fairness: every data point on average is closer to the points in its cluster than to the points in any other cluster, while Chen et al. (2019a); Mahabadi and Vakilian (2020); Vakilian and Yalciner (2022); Negahbani and Chakrabarty (2021) uses a radii-based approach to characterize fairness. Ghadiri et al. (2021); Abbasi et al. (2021); Deepak and Abraham (2020); Makarychev and Vakilian (2021); Goyal and Jaiswal (2021) study social fairness inspired by equitable representation. This body of work mainly seeks to equalize the objective cost across all groups. The notion of proportionally fair clustering is proposed by (Chen et al. (2019b); Micha and Shah (2020)) wherein subset of points are allowed to form their own clusters if a center exists that is close to all points in subset. While existing works tightly integrate achieving fairness with the clustering algorithms, Chhabra et al. (2021) recently devised the idea to use a pre-processing technique by addition of a small number of extra data points called antidotes. Vanilla clustering techniques applied to this augmented dataset result in fair clusters with respect to the original data. The pre-processing technique to add antidotes requires solving a bi-level optimization problem. While the pre-processing routine makes fair clustering algorithms irrelevant, its high running time limits its usability.

Another line of related works studying fairness in clustering revolves around hierarchical clustering, spectral clustering algorithms for graphs, and hypergraph clustering (Bose and Hamilton (2019);Kleindessner et al. (2019)). Jones et al. (2020) define fairness on the cluster centers, wherein each center comes from a demographic group. Clustering has also been used for solving fair facility location problems (Jung et al. (2020); Micha and Shah (2020); Chen et al. (2019a)). Recently, Li et al. (2021) propose a new fairness notion of core fairness that is motivated by both group and individual fairness (Kar et al. (2021)). Elzayn et al. (2019) use fair clustering for resource allocation problems. Kleindessner et al. (2019) use fair clustering for data summarization. Fair clustering is also being studied in dynamic (Chan et al. (2018)), capacitated (Quy et al. (2021)), bounded cost (Esmaeili et al. (2021)), budgeted (Byrka et al. (2014)), privacy preserving (Rösner and Schmidt (2018)), probabilistic (Esmaeili et al. (2020)), correlated (Ahmadian et al. (2020)), diversity aware (Thejaswi et al. (2021)) and distributed environments (Anderson et al. (2020)). Finally, our fairness notion (τ-ratio), resembles to that of balanced (in terms of number of points in each cluster) clustering studied by Banerjee and Ghosh (2006) without fairness constraint. However, their proposed sampling technique is not designed to guarantee τ-ratio fairness and does not analyze loss incurred due to having these fairness constraint.

3 Preliminaries

Let $X \subseteq \mathbb{R}^d$ be a finite set of points that needs to be partitioned into k clusters. Each data point $x_i \in X$ is a feature vector described using d real valued features. A k-clustering [1] algorithm $\mathcal{C} = (C, \phi)$ produces a partition of X into k subsets ($[k]$) with centers $C = \{c_j\}_{j=1}^{k}$ using an assignment function $\phi : X \rightarrow C$ which maps each point to corresponding cluster center. Throughout this paper we consider that each point $x_i \in X$ is associated with a *single* protected attribute ρ_i (say ethinicity from a pool of other available protected attributes) which takes values from the set m values denoted by $[m]$. The number of distinct protected attribute values is finite and much smaller than size of set X [2]. Furthermore, let $d : X \times X \rightarrow \mathbb{R}_+$ be a distance metric defined on X and measures the dissimilarity between features. Additionally, we are also given a vector $\tau = \{\tau_\ell\}_{\ell=1}^{m}$ where each component τ_ℓ satisfies $0 \leq \tau_\ell \leq \frac{1}{k}$ and denotes the fraction of data points from the protected attribute value $\ell \in [m]$ required to be present in each cluster. An end-user can simply specify a ℓ dimensional vector with values between 0 to $1/k$ as fairness target. Also, let us denote X_ℓ, n_ℓ as set of datapoints and number of points having value ℓ in X. Let $\mathbb{I}(.)$ denote the indicator function. A vanilla (an unconstrained) clustering algorithm determines the cluster centers as to minimize the clustering objective cost which is defined as follows:

Definition 1 (Objective Cost). *Given p, the cluster objective cost with respect to the metric space (X, d) is defined as:*

$$L_p(X, C, \phi) = \left(\sum_{x_i \in X} \sum_{j \in [k]} \mathbb{I}(\phi(x_i) = j) d(x_i, c_j)^p \right)^{\frac{1}{p}} \tag{1}$$

Different values of p, will result in different objective cost: $p = 1$ for k-medians, $p = 2$ for k-means, and $p = \infty$ for k-centers. Our aim is to develop an algorithm that minimizes the objective cost irrespective of p while ensuring the fairness.

[1] Throughout the paper, for simplicity, we call a k-clustering algorithm as a clustering algorithm.
[2] Otherwise, the problem is uninteresting as the balanced clustering may not be feasible.

Group Fairness Notions: We begin with first defining the most popular notion of group fairness which is called Balance. The notion is first put forward for binary protected groups by Chierichetti et al. (2017) and extended to multi-valued group by Bera et al. (2019); Ziko et al. (2021). The balanced fairness notion is defined as follows.

Definition 2 (Balance). *[Chierichetti et al. (2017)] The Balance of an assignment function ϕ is defined as*

$$\mathsf{Balance}(\phi) = \min_{j \in [k]} \left(\min \left(\frac{\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = a)}{\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = b)} \right) \right) \ \forall a, b \in [m] \quad (2)$$

Balance is computed by finding the minimum possible ratio of protected (say. male) and non-protected group (say. female) over all clusters. Any fair clustering algorithm using Balance as a measure of fairness would produce clusters that maximize the Balance. Note that the maximum Balance achieved by an algorithm is equal to the ratio of points available in the dataset having a and b as the protected attribute values and is known as dataset balance. Further, the clusters maximizing the Balance are not unique.

A generalization of Balance to multi-valued protected attributes is proposed by Bera et al. (2019) in terms of cluster sizes. The fairness notion constraints the upper and lower bound on the number of points from each protected group in every cluster.

Definition 3 (Minority Protection). *A clustering $\mathcal{C}$ is τ-MP if*

$$\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = \ell) \geq \tau_\ell \sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j) \ \forall \ell \in [m], \forall j \in [k] \quad (3)$$

Definition 4 (Restricted Dominance). *A clustering $\mathcal{C}$ is τ-RD if*

$$\sum_{x_i \in X} \mathbb{I}(\rho_i = \ell)\mathbb{I}(\phi(x_i) = j) \leq \tau_\ell \sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j) \ \forall \ell \in [m], \forall j \in [k] \quad (4)$$

The generalization by Bera et al. (2019) needs cluster sizes that are not known beforehand. Thus, Bera et al. (2019) proposes a linear programming-based solution.

We now define our proposed τ-ratio fairness notion which ensures that each cluster has a predefined fraction of points for each protected attribute value. τ-ratio requires only priorly known dataset composition, which helps achieve polynomial-time algorithms.

Definition 5 (τ-ratio Fairness). *An assignment function ϕ satisfies τ-ratio fairness if*

$$\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = \ell) \geq \tau_\ell \sum_{x_i \in X} \mathbb{I}(\rho_i = \ell) \ \forall j \in [k] \ \mathit{and} \ \forall \ell \ \in [m] \qquad (5)$$

The τ-ratio fairness is different from the balanced fairness of Chierichetti et al. (2017) that tries to Balance the ratio of points for any pair of values corresponding to the protected attribute in each cluster.

Our first theorem (Theorem 4) in Section 5 shows that an algorithm satisfying τ-ratio fairness notion produces one set of clusters that maximizes the Balance. In particular, when $\tau_\ell = \frac{1}{k}$, then τ-ratio fairness achieve the Balance equal to the dataset ratio. We also show that a perfectly balanced cluster need not imply τ-ratio fairness for arbitrary τ (Lemma 6 in Section 5). Hence τ-ratio is a more generalized fairness notion.

We now define the fair clustering problem with respect to the proposed fairness notion:

Definition 6 (τ-ratio Fair Clustering Problem). *The objective of a τ-ratio fair clustering problem $\mathcal{I}$ is to estimate $\mathcal{C} = (C, \phi)$ that minimizes the objective cost $L_p(X, C, \phi)$ subject to the τ-ratio fairness guarantee. The optimal objective cost of a τ-ratio fair clustering problem is denoted by $\mathcal{OPT}_{clust}(\mathcal{I})$.*

A solution to this problem is to rearrange the points (learn a new ϕ) with respect to the cluster centers obtained after a traditional clustering algorithm to guarantee τ-ratio fairness. The problem of rearrangement of points with respect to the fixed centers is known as the fair assignment problem, which we define below:

Definition 7 (τ-ratio Fair Assignment Problem). *Given X and $C = \{c_j\}_{j=1}^k$, the solution to the fair assignment problem $\mathcal{T}$ produces an assignment $\phi : X \to C$ that ensures τ-ratio fairness and minimizes $L_p(X, C, \phi)$. The optimal objective function value to a τ-ratio fair assignment problem is denoted by $\mathcal{OPT}_{assign}(\mathcal{T})$.*

However, this transformation of the fair clustering problem $\mathcal{I}$ into a fair assignment problem $\mathcal{T}$ should ensure that $\mathcal{OPT}_{assign}(\mathcal{T})$ is not too far from $\mathcal{OPT}_{clust}(\mathcal{I})$. The connection between fair clustering and fair assignment problem is established through the following lemma.

Lemma 1. *Let $\mathcal{I}$ be an instance to fair clustering problem and $\mathcal{T}$ is an instance to τ-ratio fair assignment problem after applying α-approximate algorithm to the vanilla clustering problem, then $\mathcal{OPT}_{assign}(\mathcal{T}) \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.*

Proof Let C the cluster centers obtained by running a vanilla clustering algorithm on instance $\mathcal{I}$. The proof of the Lemma depends on the existence of an assignment ϕ' satisfying τ-ratio fairness such that $L_p(X, C, \phi') \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$. As $\mathcal{OPT}_{assign}(\mathcal{T}) \leq L_p(X, C, \phi') \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$. Let (C^*, ϕ^*) denote the optimal solution to $\mathcal{I}$. Define ϕ' as follows: for every $c^* \in C^*$, let $nrst(c^*) = \operatorname{argmin}_{c \in C} d(c, c^*)$ be the nearest center to c^*. Then, for every $x \in X$, define $\phi'(x) = nrst(\phi^*(x))$. Then we have the following two claims:

Claim 2. *ϕ' satisfies τ-ratio fairness.*

Proof Let set of points having protected attribute value ℓ in cluster $c^* \in C^*$ be $n_\ell(c^*)$. Since (C^*, ϕ^*) satisfy τ-ratio fairness we have $\mid n_\ell(c^*) \mid \geq \tau_\ell n_\ell \; \forall c^* \in C^*$. For any center $c \in C$, let $N(c) = \{c^* \in C^* : nrst(c^*) = c\}$ be all the centers in C^* for which c is the nearest center. Then: $|\{x \in X_\ell : \phi'(x) = c\}| = | \cup_{c^* \in N(c)} n_\ell(c^*)| \geq n_\ell \tau_\ell$ that is union over combined assignments for each center in $N(c)$ and since each set of assignments satisfy τ-ratio so union will also satisfy τ-ratio fairness. □

Claim 3. *$L_p(X, C, \phi') \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.*

The proof of this claim uses triangle inequality and is exactly same as claim 5 of Bera et al. (2019). □

A similar technique of converting fair clustering to a fair assignment problem was proposed by Bera et al. (2019). However, Bera et al. (2019) proposed a linear programming based solution to obtain the Balance fair assignment. Although, the solution is theoretically strong, there are two issues with the algorithm. Firstly, the time complexity is high (as can be seen from the experiments in Section 7.4) and secondly, the solution obtained is not easy to interpret due to the use of the complicated linear program. By interpretability we try to find the answer to the following question – Why is a point assigned to a specific cluster to maintain fairness? What criteria did the algorithm decide for a data-point to go to a particular cluster? To answer these, our paper proposes a simple round-robin algorithm for fair assignment problem with a time complexity of $O(kn\log(n))$.

4 Fair Round-robin Algorithm for Clustering Over End (FRAC$_{OE}$)

Fair Round-robin Algorithm for Clustering Over End (FRAC$_{OE}$) algorithm first runs a vanilla clustering algorithm to produce the initial clusters $\mathcal{C} = (C, \phi)$ and then *make corrections* as follows. The algorithm first checks if τ-ratio fairness is met with the current allocation ϕ, in which case it returns $\hat{\phi} = \phi$ and $\hat{C} = C$. If the assignment ϕ violates the τ-ratio fairness constraint then the new assignment function $\hat{\phi}$ is computed according to FairAssignment procedure in Algorithm 2.

Algorithm 1: τ-FRAC$_{OE}$

```
Input: set of datapoints $X$, number of clusters $k$, fairness requirement
       vector $\tau$, range of protected attribute values $m$, clustering
       objective norm $p$
Output: cluster centers $\hat{C}$ and assignment function $\hat{\phi}$
Solve the vanilla $(k, p)$-clustering problem and let $(C, \phi)$ be the solution
  obtained.
  if $\tau$-ratio fairness is met then
      return $(C, \phi)$
      else
          $(\hat{C}, \hat{\phi}) = \textsc{FairAssignment}(C, X, k, \tau, m, p, \phi)$
          return $(\hat{C}, \hat{\phi})$
  end
end
```

Algorithm 2 iteratively allocates the data points with respect to each protected attribute value. Let X_ℓ and n_ℓ denote the set of data points and the number of data points having ℓ as the protected attribute value. The algorithm allocates $\lfloor \tau_\ell \cdot n_\ell \rfloor$ number of points [3] to each cluster in a round-robin fashion as follows. Let $\{c_1, c_2, \ldots, c_k\}$ be a random ordering of the cluster centers. At each round t, each center c_j picks the point x of its preferred choice from X_ℓ i.e. $\hat{\phi}(x) = j$. Once the τ_ℓ fraction of points are assigned to the centers, i.e., after $\tau_\ell \cdot n_\ell$ number of rounds, the allocation of remaining data points is set to its original assignment ϕ. Note that this algorithm will certainly satisfy τ-ratio fairness as, in the end, the algorithm assures that at least τ_ℓ fraction of points are allotted to each cluster for a protected attribute value ℓ. We defer to theoretical results to assert the quality of the clusters.

FRAC$_{OE}$ ensures fairness at the last step. The run time complexity of Algorithm 2 is $O(kn\log(n))$ as step 4 requires the data points to be sorted in the increasing order of their distances with the cluster centers.

5 Theoretical Results

Our first result provides the relationship between the two notions of fairness, namely τ-ratio fairness and the Balance fairness.

Theorem 4. *Let a and b be two values of a given binary protected attribute with n_a and n_b being the total number of datapoints respectively. Suppose an allocation returned by a clustering algorithm satisfies τ-ratio guarantee, then the Balance of the given allocation is atleast $\frac{\tau_a n_a}{n_b(1-k\tau_b+\tau_b)}$.*

[3] For the sake of simplicity we assume $\tau_\ell \cdot n_\ell \in \mathbb{N}$ and ignore the floor notation.

Algorithm 2: FairAssignment

```
Input: Cluster centers $C$, Set of datapoints $X$, Number of clusters $k$,
       Fairness requirement vector $\tau$, Range of protected attribute $m$,
       clustering objective norm $p$, Assignment function $\phi$
Output: Cluster centers $\hat{C}$ and assignment function $\hat{\phi}$
Fix a random ordering on centers and let the centers are numbered
  from 1 to $k$ with respect to this random ordering.
  Initialize $\hat{\phi}(x) \leftarrow 0 \ \forall x \in X$.
  for $\ell \leftarrow 1$ to $m$ do
    $n_\ell \leftarrow$ number of datapoints having value of protected attribute $\ell$.
     $X_\ell \leftarrow$ set of datapoints having value of protected attribute $\ell$.
     for $t \leftarrow 1$ to $\tau_\ell n_\ell$ do
        for $j \leftarrow 1$ to $k$ do
            $x_{min} \leftarrow \text{argmin}_{x \in X_\ell : \hat{\phi}(x)=0} d(x, c_j)$
             $\hat{\phi}(x_{min}) = j$
        end
    end
    For all $x \in X_\ell$ such that $\hat{\phi}(x) = 0$, set $\hat{\phi}(x) = \phi(x)$
end
Recompute the centers $\hat{C}$ with respect to the new allocation function $\hat{\phi}$.
  Return $(\hat{C}, \hat{\phi})$.
```

Proof Suppose an algorithm satisfies τ-ratio fairness then for any cluster C_j and protected attribute value a, we have:

$$\tau_a n_a \leq \sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = a) \leq n_a(1 - k\tau_a + \tau_a)$$

Here, the lower bound comes directly from the fairness definition and upper bound is derived from the fact that all the clusters together will be allocated at least $k\tau_a n_a$ number of points. The extra points that a particular cluster can take is upper bounded by $n_a - kn_a\tau_a$. Thus, the Balance of the cluster with respect to the two values a and b should follow

$$\frac{\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = a)}{\sum_{x_i \in X} \mathbb{I}(\phi(x_i) = j)\mathbb{I}(\rho_i = b)} \geq \frac{\tau_a n_a}{n_b(1 - k\tau_b + \tau_b)}$$

□

We remark here that the notion of Balance which is concerned with allocation of the points to clusters such that each cluster satisfies the dataset balance. We now show that the τ-ratio guarantee strictly generalizes Balance as follows. We first show that setting $\tau_i = 1/k$ for all attributes values i implies dataset balance.

Corollary 5. *For $\tau_a = \tau_b = \frac{1}{k}$, τ-ratio fairness guarantee ensures the dataset Balance for all the clusters.*

This result follows from trivially by replacing the attribute constraints in Theorem 4. We now show that the converse is not true. That is, a clustering satisfying Balance equal to dataset balance can result in arbitrary bad τ-ratio fairness.

Lemma 6. *There exists a fair clustering instance and an allocation of points such that the allocation satisfies the* Balance *property and has arbitrarily low* τ-ratio *fairness.*

Proof Consider a fair clustering instance with $k = 2$ and let the protected attribute be binary; call them a and b. Further, let $n_a = n_b = n/2$. It is easy to see that the dataset balance is 1. Consider the following allocation that satisfies the dataset balance for each cluster. Cluster 1 is assigned two points, one belonging to each attribute value and rest of the points are allocated to cluster 2. Note that for this allocation, $\tau_a = \tau_b = 1/n_a = 1/n_b = 2/n$. For large value of n this value can be made arbitrarily small. □

Along with Theorem 4, Lemma 6 shows that τ-ratio is a more general fairness notion than Balance. Apart from above technical difference, these fairness notions differ conceptually in the way they induce fair clustering. The Balance property requires a certain minimum representation ratio guarantee to hold in each cluster without any additional constraint on relative size of each of the cluster. This may lead to (potentially) skewed cluster sizes. Whereas under τ-ratio the algorithm can appropriately control the minimum number of points to be assigned to each cluster.

We now provide the theoretical guarantees of FRAC_{OE} with respect to τ-ratio fairness. We begin by providing guarantees for a perfectly balanced clusters i.e. $\tau_\ell = 1/k \; \forall \ell \in [m]$.

5.1 Guarantees for FRAC_{OE} for $\tau = \{1/k\}_{l=1}^{m}$

Theorem 7. *Let $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$. An allocation returned by* FRAC_{OE} *guarantees* τ-ratio *fairness and satisfies* 2*-approximation guarantee with respect to an optimal fair assignment upto an instance-dependent additive constant.*

Proof **Correctness and Fairness:** Clear from the construction of the algorithm.
Proof of (approximate) Optimality: We will prove 2-approximation with respect to each value ℓ of protected attribute separately. Let n_ℓ be the number of data points corresponding the value ℓ. Let c_1 and c_2 be the cluster centers and $\mathcal{C}_1$, $\mathcal{C}_2$ be the optimal fair assignment of data points with respect to these centers.[4]

We now show that $\text{FRAC}_{OE}(\mathcal{T}) \leq 2\; \mathcal{OPT}_{assign}(\mathcal{T}) + \beta$, where $\text{FRAC}_{OE}(\mathcal{T})$ and $\mathcal{OPT}_{assign}(\mathcal{T})$ denote the objective value of the solution returned by FRAC_{OE} and optimal assignment algorithm respectively on given instance $\mathcal{T} = (C, X)$. Let,

[4] Note that an optimal fair allocation need not be unique. Our result holds for any optimal fair allocation.

$\beta := 2\sup_{x,y \in X} d(x,y)$ be the diameter of the feature space. We begin with the following useful definition.

Definition 8. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent the set of points assigned to c_1 and c_2 by optimal assignment algorithm. The i^{th} round (i.e. assignments g_i to c_1 and h_i to c_2) of* FRAC$_{OE}$ *is called*

- 1*-bad if exactly one of 1) $g_i \notin \mathcal{C}_1$ and 2) $h_i \notin \mathcal{C}_2$ is true, and*
- 2*-bad if both 1) and 2) above are true.*

Furthermore, a round is called bad if it is either 1*-bad or* 2*-bad and called good otherwise.*

Let all incorrectly assigned points in a bad round be called bad assignments. We use following convention to distinguish between different bad assignments. If $g_i \notin \mathcal{C}_1$ holds we refer to it as type 1 bad assignment i.e. if point g_i is currently assigned to $\mathcal{C}_1$ but should belong to optimal clustering $\mathcal{C}_2$. Similarly if $h_i \notin \mathcal{C}_2$ holds it is a type 2 bad assignment i.e. h_i should belong to optimal clustering $\mathcal{C}_1$ but is currently assigned to c_2. Hence a 2-bad round results in 2 bad assignments one of each i.e. $g_i \in \mathcal{C}_2$ and $h_i \in \mathcal{C}_1$. Finally let B be the set of all bad rounds.

Definition 9. *(Complementary Bad Pair) A pair of points $w, z \in B$ such that w is a bad point of type t and z is a bad point of type $|3-t|$ is called a complimentary bad pair if,*

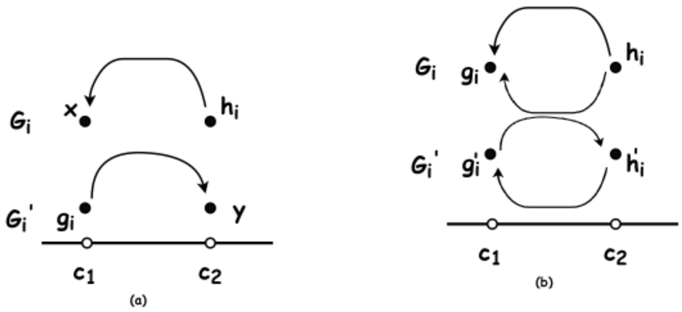
1) w and z are allocated in same round (i.e. a 2*-bad round) or*

2) if they are allocated in i^{th} and j^{th} 1*-bad rounds respectively with $i < j$, then z is the first bad point of type $|3-t|$ which has not been assigned a complementary point.*

Lemma 8. *If n_ℓ is even, every bad point in the allocation returned by* FRAC$_{OE}$ *has a complementary point. If n_ℓ is odd, at most one bad point will be left without a complementary point.*

Proof Let $B = B_1 \cup B_2$, where B_t is a set of t-bad rounds. Note that the claim is trivially true if $B_1 = \emptyset$. Hence, let $|B_1| > 0$ and write $B_1 = B_{1,1} \cup B_{1,2}$. Here $B_{1,t}$ is a 1-bad round that resulted in type t bad point. Let $H_{1,t}$ be the set of good points of type t (i.e. correctly assigned to the center c_t) allocated in 1-bad rounds. When n_ℓ is even, $|\mathcal{C}_1| = |\mathcal{C}_2|$ we have $|B_{1,2}| + |H_{1,1}| = |B_{1,1}| + |H_{1,2}|$. This is true because one can ignore good rounds and 2-bad rounds as every 2-bad round can be converted into a good round by switching the assignments. Further observe that, as FRAC$_{OE}$ distributes two points per round and each round assigns exactly one bad point, each round must assign exactly one good point i.e. $|H_{1,t}| = |B_{1,(3-t)}|$. Together, we have $|B_{1,1}| = \frac{|B_{1,2}|+|H_{1,1}|}{2} = |B_{1,2}|$. When n_ℓ is odd, we might have one additional point left in the last 1-bad round that is not being assigned any complementary point. This completes the proof of the lemma. □

We will bound the optimality of 1-bad rounds and 2-bad rounds separately.



(a) Two 1-bad round pairs (b) Two 2-bad round pairs

Fig. 1: Different cases for $k = 2$. (a) Shows two 1-bad rounds with four points such that x, y are good points and allocated to the optimal center by algorithm, whereas g_i and h_i are bad points with an arrow showing the direction to the optimal center from the assigned center. (b) Shows four bad points such that g_i, g_i' are assigned to c_1 but should belong to c_2 in optimal clustering (the arrow depicts the direction to optimal center). Similarly h_i, h_i' should belong to c_1 in optimal clustering.

Bounding* 1*-bad rounds:

When n_ℓ is even, from Lemma 8, there are even number of 1-bad rounds; two for each complimentary bad pair. Let the 4 points of corresponding two 1-bad rounds be $G_i : (x, h_i)$ and $G_i^{'} : (g_i, y)$ as shown in Fig. 1a. Note that $x \in \mathcal{C}_1$ and $y \in \mathcal{C}_2$ ie. both are good points and $g_i \notin \mathcal{C}_1$, $h_i \notin \mathcal{C}_2$ ie. are bad points. Now, consider an instance $\mathcal{T}_i = \{C, \{x, h_i, g_i, y\}\}$, then $\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x, c_1) + d(h_i, c_1) + d(g_i, c_2) + d(y, c_2)$. We consider, without loss of generality, that the round G_i takes place before $G_i^{'}$ in the execution of FRAC_{OE}. The proof is similar for the other case. First note that since FRAC_{OE} allocated the point h_i to cluster 2 while both the points g_i and y were available, we have

$$d(h_i, c_2) \le d(g_i, c_2) \text{ and } d(h_i, c_2) \le d(y, c_2) \tag{6}$$

So,

$$\begin{aligned}
&\text{FRAC}_{OE}(\mathcal{T}_i)\\
&= d(x, c_1) + d(h_i, c_2) + d(g_i, c_1) + d(y, c_2)\\
&\le d(x, c_1) + d(h_i, c_2) + d(g_i, c_2) + d(c_1, c_2) + d(y, c_2) \qquad \text{(triangle inequality)}\\
&\le d(x, c_1) + d(h_i, c_2) + d(g_i, c_2) + d(h_i, c_2) + d(h_i, c_1) + d(y, c_2)\\
&\le d(x, c_1) + d(y, c_2) + d(g_i, c_2) + d(g_i, c_2) + d(h_i, c_1) + d(y, c_2) \qquad \text{(Eqn. 6)}\\
&\le 2\ \mathcal{OPT}_{assign}(\mathcal{T}_i)
\end{aligned}$$

If n_ℓ is odd, then all the other rounds can be bounded using the above cases except one extra 1-bad round. Let the two points corresponding to this round G_i be (g_i, y). Thus, $\text{FRAC}_{OE}(\mathcal{T}_i) \le 2\mathcal{OPT}_{assign}(\mathcal{T}_i) + \beta$.Here β=$2\sup_{x,y\in\mathcal{X}} d(x, y)$ is the diameter of the feature space.

Bounding* 2*-bad rounds:

First assume that there are even number of 2-bad rounds. In this case consider the pairs of consecutive 2-bad rounds as $G_i : (g_i, h_i)$ and $G_i^{'} = (g_i^{'}, h_i^{'})$ with $G_i^{'}$ bad round followed by G_i (Fig. 1b). Note that $g_i, g_i^{'} \in \mathcal{C}_2$ and $h_i, h_i^{'} \in \mathcal{C}_1$. Now consider instance $\mathcal{T}_i = \{C, \{g_i, g_i^{'}, h_i, h_i^{'}\}\}$, then , $\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(h_i, c_1) + d(h_i^{'}, c_1) + d(g_i, c_2) + d(g_i^{'}, c_2)$. As a consequence of allocation rule used by FRAC_{OE} we have

$$d(g_i, c_1) \leq d(h_i, c_1),\ d(g_i^{'}, c_1) \leq d(h_i^{'}, c_1) \text{ and } d(h_i, c_2) \leq d(h_i^{'}, c_2). \tag{7}$$

Furthermore,

$$\begin{aligned}
\text{FRAC}_{OE}(\mathcal{T}_i) &= d(g_i, c_1) + d(g_i^{'}, c_1) + d(h_i, c_2) + d(h_i^{'}, c_2) \\
&\leq d(h_i, c_1) + d(h_i^{'}, c_1) + d(g_i^{'}, c_2) + d(h_i^{'}, c_2) && \text{(using Eqn. 7)} \\
&\leq d(h_i, c_1) + d(h_i^{'}, c_1) + d(g_i^{'}, c_2) + d(h_i^{'}, c_1) + d(c_1, c_2) \\
& && \text{(triangle inequality)} \\
&\leq d(h_i, c_1) + d(h_i^{'}, c_1) + d(g_i^{'}, c_2) + d(h_i^{'}, c_1) + d(g_i, c_1) \\
&+ d(g_i, c_2) && \text{(triangle inequality)} \\
&\leq d(h_i, c_1) + d(h_i^{'}, c_1) + d(g_i^{'}, c_2) + d(h_i^{'}, c_1) + d(h_i, c_1) \\
&+ d(g_i, c_2) && \text{(Using Eqn. 7)} \\
&\leq 2d(h_i, c_1) + 2d(h_i^{'}, c_1) + d(g_i, c_2) + d(g_i^{'}, c_2) \\
&\leq 2\mathcal{OPT}_{assign}(\mathcal{T}_i)
\end{aligned}$$

If there are odd number of 2-bad rounds then, let $G = (g_i, h_i)$ be the last 2-bad round. It is easy to see that $\text{FRAC}_{OE}(\mathcal{T}_i) - \mathcal{OPT}_{assign}(\mathcal{T}_i) = d(g_i, c_1) + d(h_i, c_2) - d(g_i, c_2) - d(h_i, c_1) \leq d(g_i, c_1) + d(h_i, c_2) \leq \beta$. Thus,

$$\begin{aligned}
\text{FRAC}_{OE}(\mathcal{T}) &= \begin{cases} \sum_{i=1}^{r/2} \text{FRAC}_{OE}(\mathcal{T}_i) & \text{if even no. of 2-bad rounds} \\ \sum_{i=1}^{\lfloor r/2 \rfloor} \text{FRAC}_{OE}(\mathcal{T}_i) + \beta & \text{Otherwise} \end{cases} \\
&\leq 2 \sum_{i=1}^{\lfloor r/2 \rfloor} \mathcal{OPT}_{assign}(\mathcal{T}_i) + \beta = 2\mathcal{OPT}_{assign}(\mathcal{T}) + \beta
\end{aligned}$$

Here, r is the number of 2-bad rounds. and $\beta{=}2\sup_{x,y \in \mathcal{X}} d(x,y)$ is the diameter of the feature space. □

Corollary 9. *For $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, we have $\text{FRAC}_{OE}(\mathcal{I})$ $\leq (2(\alpha+2)\mathcal{OPT}_{clust}(\mathcal{I}) + \beta)$-approximate where α is approximation factor for vanilla clustering problem for any given instance $\mathcal{I}$.*

The above corollary is a direct consequence of Lemma 1 and the fact that $\text{FRAC}_{OE}(\hat{C}, X) \leq \text{FRAC}_{OE}(C, X)$. The result can easily be extended for k clusters to directly obtain 2^{k-1}-approximate solution with respect to τ-ratio fair assignment problem.

Theorem 10. *When $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, an allocation returned by FRAC_{OE} for given centers and data points is τ-ratio fair and satisfies 2^{k-1}-approximation guarantee with respect to an optimal τ-ratio fair assignment problem up to an instance-dependent additive constant.*

Proof In the previous proof we basically considered two length cycles. Two 1-bad allocations resulted in 1 cycles and one 2-bad allocations resulted in another type of cycles. When the number of clusters are greater than two, then any $2 \leq q \leq k$ length cycles can be formed. Without loss of generality, let us denote $\{c_1, c_2, \dots, c_q\}$ as the centers that are involved in forming such cycles. Further denote by set X_i^j to be the set of points that are allotted to cluster i by FRAC$_{OE}$ but should have been allotted to cluster j in an optimal fair clustering. The q length cycle can then be visualized in the Fig. 2. Since the cycle is formed with respect to these points, we have $|X_1^q| = |X_2^1| = \ldots = |X_q^{q-1}|$ The cost by FRAC$_{OE}$ algorithm is then given as:

$$\sum_{i=2}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i) + \sum_{x \in X_1^q} d(x, c_1)$$

$$\leq 2 \left(\sum_{x \in X_2^1} d(x, c_1) + \sum_{x \in X_1^q} d(x, c_2) + \beta \right) + \sum_{i=3}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i)$$

$$\leq 2(\sum_{x \in X_2^1} d(x, c_1) + \beta) + 2^2 \left(\sum_{x \in X_3^2} d(x, c_2) + \sum_{x \in X_1^q} d(x, c_3) + \beta \right) + \sum_{i=4}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i)$$

$$\leq 2^{q-1} \left(\sum_{i=2}^{q} \sum_{x \in X_i^{i-1}} d(x, c_{i-1}) + \sum_{x \in X_1^q} d(x, c_q) \right) + 2^q \beta$$

Here, the first inequality follows by exchanging the points in X_2^1 and X_1^q using Theorem 7. Since the maximum length cycle possible is k, we straight away get the proof of 2^{k-1}- approximation. □

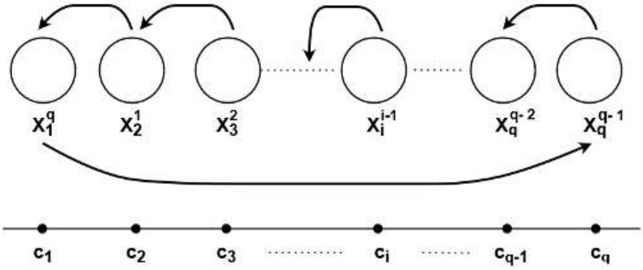


Fig. 2: Visual representation of set X_i^j and cycle of length q for Theorem 10. The arrow represents the direction from the assigned center to the center in optimal clustering. Thus, for each set X_i^j we have c_i as the currently assigned center and c_j as the center in optimal assignment.

Next, in contrast with Theorem 10 which guarantees a 4-approximation for $k = 3$, we show that one can achieve a 2-approximation guarantee. The proof of this result relies on explicit case analysis and, as the number of cases to be solved increase exponentially with k, one needs a better proof technique for larger values of k. We leave this analysis as an interesting future work.

Theorem 11. *For k=3 and $\tau_\ell = \frac{1}{k}$ allocation returned by* FRAC_{OE} *with arbitrary centers and data points is* 2*-approximate with respect to optimal* τ*-ratio fair assignment.*

Proof We will here find the approximation for $k = 3$ using number of possible cases where one can have cycle of three length. Let the centers involved in three cycles be denoted by c_i, c_j, and c_k. Note that if there is only one cycle involving these three centers, then it will lead to only constant factor approximation. The challenge is when multiple such cycles are involved. Unlike $k = 2$ proof, here we bound the cost corresponding to each cycle with respect to the cost of another cycle. The three cases shown in Fig. 3 depicts multiple rounds when the two 3-length cycles can be formed. In the figure, if c_i is taking a point from c_j it is denoted using an arrow from c_i to c_j. It can further be shown that it is enough to consider these three cases. Further, let $\mathcal{T}_i = \{C, \{x_i, x_j, x_k, g_i, g_j, g_k\}\}$ and $\mathcal{T}_i' = \{C, \{y_i, y_j, y_k, g_i', g_j', g_k'\}\}$ denote the two instances.

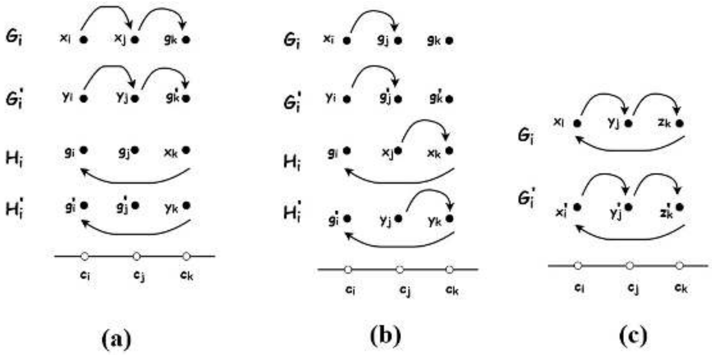


Fig. 3: Different use cases for 3-length cycle involving k=3 clusters (a) Case 1: Two-three length cycle pair (G_i, H_i) and (G_i', H_i') (b) Case 2: Second possibility of two-three length cycle pair (G_i, H_i) and (G_i', H_i') (c) Case 3:Three length cycle pair (G_i, G_i').

Case 1: In this case we bound the rounds shown in Fig. 3(a). Let, one cycle completes in rounds G_i, H_i and another cycle completes in rounds G_i', H_i'. Then,

$$\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x_i, c_j) + d(x_j, c_k) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_i)$$
$$\mathcal{OPT}_{assign}(\mathcal{T}_i') = d(y_i, c_j) + d(y_j, c_k) + d(g_k', c_k) + d(g_i', c_i) + d(g_j', c_j) + d(y_k, c_i)$$

Further,

$$\begin{aligned}\mathrm{FRAC}_{OE}(\mathcal{T}_i) &= d(x_i, c_i) + d(x_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k) \\ &\le d(g_i', c_i) + d(g_j', c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k)\end{aligned}$$

Now,

$$\begin{aligned}d(x_k, c_k) &\le d(x_k, c_i) + d(c_i, c_k) \le d(x_k, c_i) + d(c_i, c_j) + d(c_j, c_k) \\ &\le d(x_k, c_i) + d(x_i, c_i) + d(x_i, c_j) + d(x_j, c_j) + d(x_j, c_k)\end{aligned}$$

$$\leq d(x_k, c_i) + d(y_k, c_i) + d(x_i, c_j) + d(y_i, c_j) + d(x_j, c_k)$$

Combining the above two, we get:

$$\text{FRAC}_{OE}(\mathcal{T}_i) \leq \mathcal{OPT}_{assign}(\mathcal{T}_i) + \mathcal{OPT}_{assign}(\mathcal{T}_i')$$

Thus, the cost of each cycle can be bounded by the sum of optimal cost of its own and the optimal cost of the next cycle. If we take sum over all such cycles, we will get 2-approximation result plus a constant due to the last remaining cycle.

Case 2: In this case we bound the rounds shown in Fig. 3(b). The optimal assignments in this case will be

$$\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x_i, c_j) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(x_j, c_k) + d(x_k, c_i)$$

$$\mathcal{OPT}_{assign}(\mathcal{T}_i^{'}) = d(y_i, c_j) + d(g_j^{'}, c_j) + d(g_k^{'}, c_k) + d(g_i^{'}, c_i) + d(y_j, c_k) + d(y_k, c_i)$$

Also, we know that

$$\begin{aligned}\text{FRAC}_{OE}(\mathcal{T}_i) &= d(x_i, c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(x_j, c_j) + d(x_k, c_k)\\ &\leq d(g_i^{'}, c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(c_i, c_j) +\\ &d(y_j, c_k)\\ &\leq d(g_i^{'}, c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(x_k, c_i) +\\ &d(x_i, c_j) + d(y_j, c_k)\end{aligned}$$

Combining the above two, we get:

$$\text{FRAC}_{OE}(\mathcal{T}_i) \leq \mathcal{OPT}_{assign}(G_i, H_i) + \mathcal{OPT}_{assign}(G_i', H_i')$$

Case 3: Here again we will have two allocation rounds namely $G_i, G_i^{'}$ as shown in Fig. 3 (c). It is easy to see that for this case,

$$\text{FRAC}_{OE}(G_i) \leq \mathcal{OPT}_{assign}(G_i')$$

This completes the proof for $k = 3$. □

The following proposition proves that 2-approximation guarantee is tight with respect to FRAC_{OE} algorithm.

Proposition 12. *There is an instance with arbitrary centers and data points on which* FRAC_{OE} *achieves* 2*-approximation with respect to optimal assignment.*

Proof The worst case for any fair clustering instance can be the situation wherein rather than choosing the points from the center's own set of optimal points, it prefers points from other centers. One such example is depicted in Fig. 4. In this example we consider k centers, and for each of these centers we have set of n optimal points that are at a negligible distance (say zero) and these set are denoted by X_i for center c_i except the last center c_k. The set of optimal points for center c_k is located at a distance Δ such that $\Delta = (k-1)\delta$ where δ is the distance between all the centers. Now we will try to approximate the tightest bound on cost that one can achieve. In optimal assignment each cluster center will take points from its optimal set of points. Thus optimal cost can be summed up as

$$\mathcal{OPT}_{assign} = \sum_{x_i \in X_1} d(x_i, c_1) + \sum_{x_i \in X_2} d(x_i, c_2) + \ldots + \sum_{x_i \in X_k} d(x_i, c_k)$$

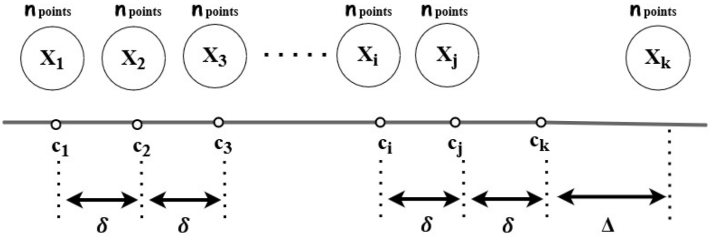


Fig. 4: The worst case example for fair clustering instance.

$$= 0 + 0 + 0 + n\Delta$$

If one uses round-robin based FRAC$_{OE}$ to solve assignment problem then at the start of $t = 0^{th}$ round, each of the set X_i has n points. Now since Δ is quite large as compared to δ so c_k will prefer to chose points from the set of previous center c_{k-1}. Rest all centers will take points from their respective set of optimal points as those points will be at the least cost of zero. This type of assignment will continue until all the points in set X_{k-1} gets exhausted. Thus the cost after $n/2$ rounds will be

$$Cost_1 = \sum_{x_i \in X_1} d(x_i, c_1) + \ldots + \sum_{x_i \in X_{k-1}} d(x_i, c_{k-1}) + \sum_{x_i \in X_{k-1}} d(x_i, c_{k-1})$$
$$= 0 + 0 + 0 + \frac{n\delta}{2}$$

Now since all the points in set X_{k-1} are exhausted, both c_{k-1} and c_k will prefer to choose the points from set X_{k-2}. Other centers will still continue to choose the points from their respective optimal sets. It should be noted that now $\frac{n}{2}$ points are left with the center X_{k-2} that are being distributed amongst 3 clusters. Such assignments will be take place for next $\frac{n}{6}$ rounds and after that the set X_{k-2} will get exhausted. The cost incurred to different centers in such assignment will be

$$Cost_2 = \sum_{x_i \in X_1} d(x_i, c_1) + \ldots + \sum_{x_i \in X_{k-2}} d(x_i, c_{k-2}) + \sum_{x_i \in X_{k-2}} d(x_i, c_{k-1})$$
$$+ \sum_{x_i \in X_{k-2}} d(x_i, c_k)$$
$$= \frac{n\delta}{6} + \frac{2n\delta}{6}$$
$$= \frac{3n\delta}{6} = \frac{n\delta}{2}$$

It is easy to see that the additional cost that is incurred at each phase will be $\frac{n\delta}{2}$ until the only left out points are from X_k. The total number of such phases will be $k-1$. Thus, exhibiting a cost of $\frac{n(k-1)\delta}{2}$. Further, at the last round all the points from X_k need to be equally distributed amongst $X_1, X_2, \ldots, X_k$, thus incurring the total cost of $((k-1)\delta + \Delta + (k-2)\delta + \Delta + \ldots + \delta + \Delta + \Delta)\frac{n}{k}$. Thus, the total cost by FRAC$_{OE}$ is given as:

$$Cost_{\text{FRAC}_{OE}} = \frac{n(k-1)\delta}{2} + ((k-1)\delta + \Delta + (k-2)\delta + \Delta + \ldots + \delta + \Delta + \Delta)\frac{n}{k}$$

$$
\begin{aligned}
&= \frac{n(k-1)\delta}{2} + \frac{nk(k-1)\delta}{2k} + \frac{nk\Delta}{k} \\
&= n(k-1)\delta + n\Delta \\
&= 2n\Delta
\end{aligned}
$$

□

Research gap: Theorem 10 suggests that the approximation ratio with respect to the number of clusters k can be exponentially bad. However, our experiments show—agreeing with our finding on small values of $k(\leq 3)$—that the performance of FRAC$_{OE}$ does not degrade with k. To assert a 2-approximation bound for general k a novel proof technique is needed and we leave this analysis as an interesting future work. Here, we provide the following conjecture.

Conjecture 13. FRAC$_{OE}$ *is* 2*-approximate with respect to optimal* τ-ratio *fair assignment problem for any value of* k*.*

We note that FRAC$_{OE}$ uses vanilla k-means/k-median algorithm followed by one round of fair assignment procedure. It remains to be shown that given a convergence guarantee of a clustering algorithm, the output of the returned by the FRAC$_{OE}$ algorithm indeed converges to approximately optimal τ-ratio allocation in finite time. Convergence guarantees of vanilla clustering algorithms are well known in the literature (Bottou and Bengio (1994); Kalyanakrishnan (2016); Krause (2016)). Since, fair assignment procedure performs corrections for all available data points only once, FRAC$_{OE}$ is bound to converge. This gives us the following lemma.

Lemma 14. FRAC$_{OE}$ *algorithm converges.*

5.2 Guarantees for FRAC$_{OE}$ for general τ

We first begin with a simple observation that the problem of solving τ-ratio fair assignment problem on instance $\mathcal{T}$ for given centers C and set of points X. The problem can be divided into two subproblems:

1. Solving $1/k$-ratio fair assignment problem on subset of points $X_1 \in X$ such that $|X_1| = \sum_{\ell \in [m]} k.\tau_\ell.n_\ell$.
2. Solving optimal fair assignment problem on $X_2 \in X \setminus X_1$ without any fairness constraint.

Let us denote the first instance by $\mathcal{T}^{1/k}$ and second instance with $\mathcal{T}^0$.

Lemma 15. *There exists two separate instances* $\mathcal{T}^{1/k}$ *with* $\tau{=}\{1/k\}_{\ell=1}^m$ *and* $\mathcal{T}^0$ *with* $\tau{=}\{0\}_{\ell=1}^m$ *such that fair assignment problem on instance* $\mathcal{T}$ *can be divided into solving two problem on these two instances, i.e.,* $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}^0)$*.*

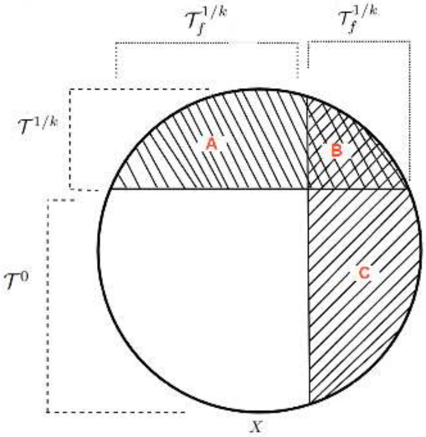


Fig. 5: Set of points X divided into instance $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Further the instances $\mathcal{T}_f^{1/k}$ and $\mathcal{T}_f^0$ are depicted in the same set of points X leading to formation of regions A, B, C.

Proof The $\mathcal{T}$ basically ensures that each cluster should have atleast $\tau_\ell.n_\ell$ number of points. Rest all the points can be allocated in the optimal manner without any fairness constraint. Therefore in optimal assignment, there exists a set X_1^{OPT} such that $|X_1^{OPT}| = \sum_{\ell=1}^{m} \tau_\ell.n_\ell.k$ that satisfy the $\tau-$ratio fairness with $\tau_\ell = 1/k\, \forall \ell \in [m]$. □

Let X_1^f be the set of points that are allocated in line number 4 by Algorithm 2. Further, let $\mathcal{T}_f^{1/k}$ be an instance to τ-ratio fair assignment problem with $\tau = \{1/k\}_{\ell=1}^m$ and $\mathcal{T}_f^0$ be instance when τ=$\{0\}_{\ell=1}^m$ by FRAC$_{OE}$ (depicted in Fig. 5). Then, our next lemma shows that the partition returned by FRAC$_{OE}$ is the optimal one.

Lemma 16. *$\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}^0)$ for any partition $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Thus, $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.*

Proof We divide the complete set of points X into three regions A, B, and C as shown in Fig. 5. The region B contains the points in the overlap of $\mathcal{T}^{1/k}$ and $\mathcal{T}_f^{1/k}$. Since, we are talking about the optimal assignment problem, these points will be assigned to same centers and hence we can ignore these points. Let the points allocated to any center c_j in $\mathcal{T}_f^{1/k}$ by FRAC$_{OE}$ be $P = \{x_1, x_2, x_3, \ldots, x_{m_j}\}$ and points allocated to c_j in partition $\mathcal{T}^{1/k}$ be $Q = \{y_1, y_2, y_3, \ldots, y_{m_j}\}$. Let g be a mapping function from $P \rightarrow Q$. It maps any point x_j assigned to center i to some point y_j assigned to same center when partition under consideration is $\mathcal{T}^{1/k}$. Then, we have $\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) \leq$ FRAC$_{OE}(\mathcal{T}_f^{1/k}) = \sum_{j=1}^{k}\sum_{i=1}^{m_j} d(x_i, c_j) \leq$

$\sum_{j=1}^{k}\sum_{i=1}^{m_j} d(y_i, c_j) = \mathcal{OPT}_{assign}(\mathcal{T}^{1/k})$. This is because despite point y_i being available to center c_j, it chose the point x_i. Since other points have no such constraint, we have, $\mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq \mathcal{OPT}_{assign}(\mathcal{T}^0)$.

□

Theorem 17. *For k=2, 3 and any general τ vector, an allocation returned by* FRAC$_{OE}$ *guarantees τ-ratio fairness and satisfies* $(2(\alpha + 2)\mathcal{OPT}_{clust})$-*approximate guarantee with respect to an fair clustering problem where α is approximation factor for vanilla clustering problem.*

Proof With the help of Lemma 15 the cost of FRAC$_{OE}$ on instance $\mathcal{T}_f$ can be computed as,

$$\text{FRAC}_{OE}(\mathcal{T}) = \text{FRAC}_{OE}(\mathcal{T}_f^{1/k}) + \text{FRAC}_{OE}(\mathcal{T}_f^{0}) \tag{8}$$

Now, from Section 5.1, FRAC$_{OE}(\mathcal{T}_f^{1/k}) \leq 2.\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k})$.

Also, since $\mathcal{T}_f^0$ is solved for $\tau=\{0\}_{\ell=1}^{m}$ i.e. assignment is carried solely on the basis of $k-$means clustering, so we have FRAC$_{OE}(\mathcal{T}_f^0) = \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq 2.\mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.

So Equation 8 becomes,

$$\begin{aligned} \text{FRAC}_{OE}(\mathcal{T}) &= 2.\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + 2.\mathcal{OPT}_{assign}(\mathcal{T}_f^{0}) \\ &= 2.\mathcal{OPT}_{assign}(\mathcal{T}) && \text{(using Lemma 15)} \\ &= 2.(\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I}) && \text{(Using Lemma 1)} \end{aligned}$$

□

6 Fair Round Robin Algorithm for Clustering (FRAC) –A Heuristic Approach

We now propose another algorithm, a general version of FRAC$_{OE}$ where the fairness constraints are satisfied at each allocation round: Fair Round-Robin Algorithm for Clustering FRAC (described in Algorithm 3). FRAC runs a fair assignment problem at each iteration of a vanilla clustering algorithm.

It is theoretically hard to analyze FRAC as it is an in-processing algorithm and each round's allocation depends upon previous rounds, i.e., the rounds are not independent of each other. Thus, we experimentally show the convergence of both FRAC and FRAC$_{OE}$ on real-world datasets. We also show that FRAC achieves the best objective cost amongst all the available algorithms in the literature. Since both FRAC$_{OE}$ and FRAC solve the fair assignment problem on the top of the vanilla clustering problem. Thus, one can use them to find fair clustering for center-based approaches, i.e., k-means and k-median.

7 Experimental Result and Discussion

We validate the performance of proposed algorithms across many benchmark datasets and compare it against the SOTA approaches. We observe in Section

Algorithm 3: τ-FRAC

```
Input: Set of datapoints $X$, Number of clusters $k$, Fairness
       requirement vector $\tau$, Range of protected attribute $m$,
       clustering objective norm $p$
Output: Cluster centers $C$ and assignment function $\phi$
Choose the random centers as $C$
while $UntilConvergence$ do
    for each $x_i \in X$ do
        $\phi(x_i) = \text{argmin}_m d(x_i, c_m)$
    end
    $(C, \phi) = \textsc{FairAssignment}(C, X, k, \tau, m, p, \phi)$
end
```

7.3.1 that the performance of FRAC is better than FRAC_{OE} in terms of objective cost. It is also evident that FRAC applies the fairness constraints after each round.

The bench marking datasets used in the study are

- **Adult**[5] **(Census)**- The data set contains information of 32562 individuals from the 1994 census, of which 21790 are males and 10771 are females. We choose five attributes as feature set: age, fnlwgt, education_num, capital_gain, hours_per_week; the binary-valued protected attribute is sex, which is consistent with prior literature. The Balance in the dataset is 0.49.
- **Bank**[6]- The dataset consists of marketing campaign data of portuguese bank. It has data of 41108 individuals, of which 24928 are married, 11568 are single, and 4612 are divorced. We choose six attributes as the feature set: age, duration, campaign, cons.price.idx, euribor3m, nr.employed; the ternary-valued feature martial status is chosen as the protected attribute to be consistent with prior literature, resulting in a Balance of 0.18.
- **Diabetes**[7]- The dataset contains clinical records of 130 US hospitals over ten years. There are 54708 and 47055 hospital records of males and females, respectively. Consistent with the prior literature, only two features: age, time_in_hospital are used for the study. Gender is treated as the binary-valued protected attribute yielding a Balance of 0.86.
- **Census II**[8]- It is the largest dataset used in this study containing 2458285 records from of US 1990 census, out of which 1191601 are males, and 1266684 are females. We choose 24 attributes commonly used in prior literature for this study. Sex is the binary-valued protected attribute. The Balance in the dataset is 0.94.

The dataset characteristics are summarized in Table 1. We compare the application of FRAC to k-means and k-median against the following baseline and SOTA approaches

[5] https://archive.ics.uci.edu/ml/datasets/Adult

[6] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

[7] https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

[8] https://archive.ics.uci.edu/ml/datasets/US+Census+Data+%281990%29

Dataset Name	**#Cardinality**	**#Feature Attributes**	**Protected Attribute**	**Protected Attribute Cardinality**	**Protected Attribute Composition**			**Dataset Balance**
Adult (Census)	32562	5	gender	binary	21790 males	10771 females	–	0.49
Bank	41108	6	marital status	ternary	24928 married	11568 unmarried	4612 divorced	0.18
Diabetes	101763	2	gender	binary	54708 males	47055 females	–	0.86
Census II	2458285	24	gender	binary	1191601 males	1266684 females	–	0.94

Table 1: Characteristics for real-world datasets commonly used in evaluation of fair clustering algorithms. Number of feature attributes exclude protected attribute and for complete list of feature attributes see Section 7.

- **Vanilla k-means**: A Euclidean distance-based k-means algorithm that does not incorporate fairness constraints
- **Vanilla k-median**: A Euclidean distance-based k-median algorithm that does not incorporate fairness constraints.
- **Bera et al. (2019)**: The approach solves the fair clustering problem through an LP formulation. The fairness is added as an additional constraint in LP by bounding the minimum (minority protection see Definition 3) and maximum (restricted dominance see Definition 4) fraction of points belonging to the particular protected group in each cluster. Due to the high computational complexity of the k-median version of the approach, we restrict the comparison to the k-means version. Furthermore, the algorithm fails to converge in a reasonable time when the number of clusters is greater than 10 for larger datasets.
- **Ziko et al. (2021)**: This approach formulates a regularized optimization function incorporating clustering objective and fairness error. It does not allow the user to give an arbitrary fairness guarantee but computes the optimal trade-off by tuning a hyper-parameter λ. We compare against both the k-means and k-median version of the algorithm. We observed that the hyper-parameter λ is extremely sensitive to the datasets and the number of clusters. Tuning this hyper-parameter is computationally expensive. We were able to tune value of λ in a reasonable amount of time only for adult and bank datasets for k-means clustering for varying number of clusters. Due to the added complexity of k-medians, we were able to fine tune λ only for the adult dataset. For the other cases, we have used the hyper-parameter value reported by Ziko et al. We have used the same value across varying number of cluster centers. The paper does not report any results for diabetes dataset; we have chosen the best λ value over a single run of fine-tuning. This value is used across all experiments related to diabetes dataset.
- **Backurs et al. (2019)**: This approach computes the fair clusters using fairlets in an efficient manner and is the extension to that of Chierichetti et al. (2017). This approach could only be integrated with k-median

clustering. Further, we could not compare against k-median version of Chierichetti et al. (2017) due to high computational ($O(n^2)$) and space complexities. We offset this comparison using Backurs et al. (2019) that has shown to result in better performance than Chierichetti et al. (2017).

We use the following popular metrics in the literature for measuring the performance of the different approaches.

- **Objective Cost**: We use the squared euclidean distance ($p = 2$) as the objective cost to estimate the cluster's compactness (see Definition 1).
- **Balance**: The Balance is calculated using Definition 2
- **Fairness Error** This notion of fairness constraints is introduced by Ziko et al. (2021). It is the Kullback-Leibler (KL) divergence between the required protected group proportion τ and achieved proportion within the clusters:

$$FE(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{\ell \in [m]} \left(-\tau_\ell \log \left(\frac{q_\ell}{\tau_\ell} \right) \right) \; where \; q_\ell = \left(\frac{\sum_{x_i \in C} \mathbb{I}(\rho_i = \ell)}{\sum_{x_i \in X} \mathbb{I}(\rho_i = \ell)} \right) \tag{9}$$

The τ vector in fairness error captures the target proportion in each cluster for different protected groups $\ell \in [m]$. It can be any arbitrary ℓ dimensional vector. In the experimental setting with $\tau = 1/k$, target reduces to dataset proportion for different groups to evaluate all baselines. In a generalized setting, when $\tau < 1/k$, it is the same as the input vector τ for FRAC and FRAC$_{OE}$ algorithms that achieve τ-ratio fairness constraints. Similarly, in Bera et al. (2019), the target vector is δ (refer Section 7.3.3 for details on the parameter δ). We report the average and standard deviation of the performance measures across 10 independent trials for every approach. The code for all the experiments is publicly available[9]. We begin the empirical analysis of various approaches under both k-means and k-median settings for a fixed value of k (=10) in line with the previous literature. The top and bottom row in Fig. 6 summarize the results obtained for the k-means and k-median settings respectively. The plots for k-means clustering clearly reveal the ability of FRAC and FRAC$_{OE}$ to maintain the perfect Balance and zero fairness error. While Bera et al. (2019) is also able to achieve similar fairness performance, FRAC, FRAC$_{OE}$ has significantly lower objective cost. Though Ziko et al. (2021) returns tighter clusters ie., the objective cost is lower than FRAC, FRAC$_{OE}$ and Bera et al. (2019), the lower objective comes at the cost of poor performance on both the fairness measures. It is also observed that the cost of fairness is relatively high in the Census-II dataset, which has the largest number of points and features among all datasets. It may be due to the shifting of an increased number of points compared to vanilla clustering for satisfying the hard constraint.

In the k-median setting, it can been observed from the plots that Backurs et al. (2019) results in fair clusters with high objective cost. On the other hand

[9] https://github.com/shivi98g/Fair-k-means-Clustering-via-Algorithmic-Fairness

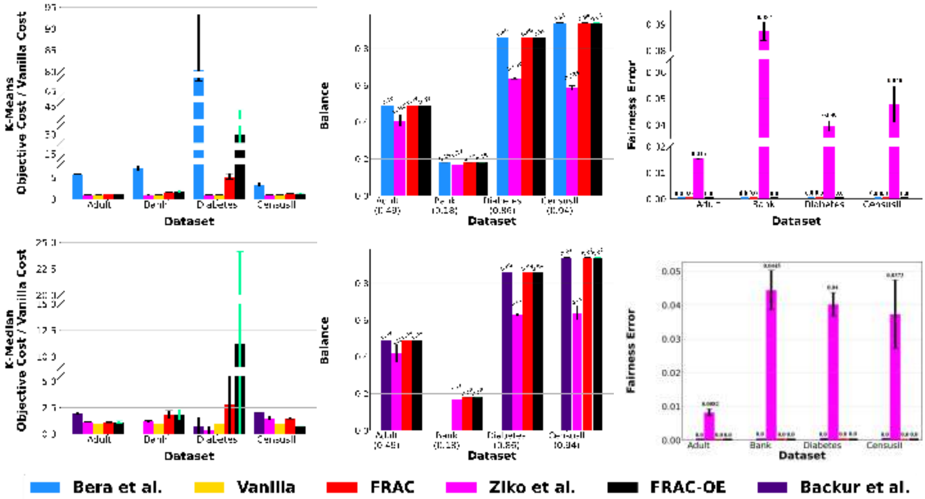


Fig. 6: The plot in the first row shows the variation in evaluation metrics for k=10 clusters.The objective cost is scaled against vanilla objective cost. For Ziko et al. the λ values for k-means and k-median are taken to be same as in their paper. The second row comprises of plots for k-median setting on same k value. It should be noted that Backur et al. does not work for bank dataset which has ternary valued protected group. The target Balance of each dataset is evident from the axes of the plot. (Best viewed in color)

Ziko et al. achieves better objective costs trading off for fairness. The k-median version of FRAC, FRAC$_{OE}$ obtains the least fairness error and a Balance that is equal to the required dataset ratio ($\tau_\ell = \frac{1}{k}$) while having comparable objective cost.

7.1 Comparison across varying number of clusters (k)

In this experiment, we measure the performance of the k-means version of the different approaches across all the datasets as the number of clusters increased from 2 to 40. Fig. 7 summarises the results obtained for 2, 5, 10, 15, 20, 30, and 40 number of clusters on all datasets. It can be observed for all datasets that Bera et al. (2019) maintain fairness constraints but with a much higher objective cost and standard deviation. For the largest dataset, Census-II, results are obtained for only $k = 5$ and $k = 10$ due to the large time complexity of solving the LP problem. Another interesting observation is that the LP-solver fails to return any solution for $k = 2$. When we allow fine tuning of hyperparameter λ, it can be observed that the trend in the objective cost value for Ziko et al. with increasing the number of clusters follows closely to that of the vanilla k-means objective cost on the Adult and Bank datasets. However, there is a significant deterioration in the Balance and fairness error measures. The results of Ziko et al. when using the λ value reported in the paper for a particular k

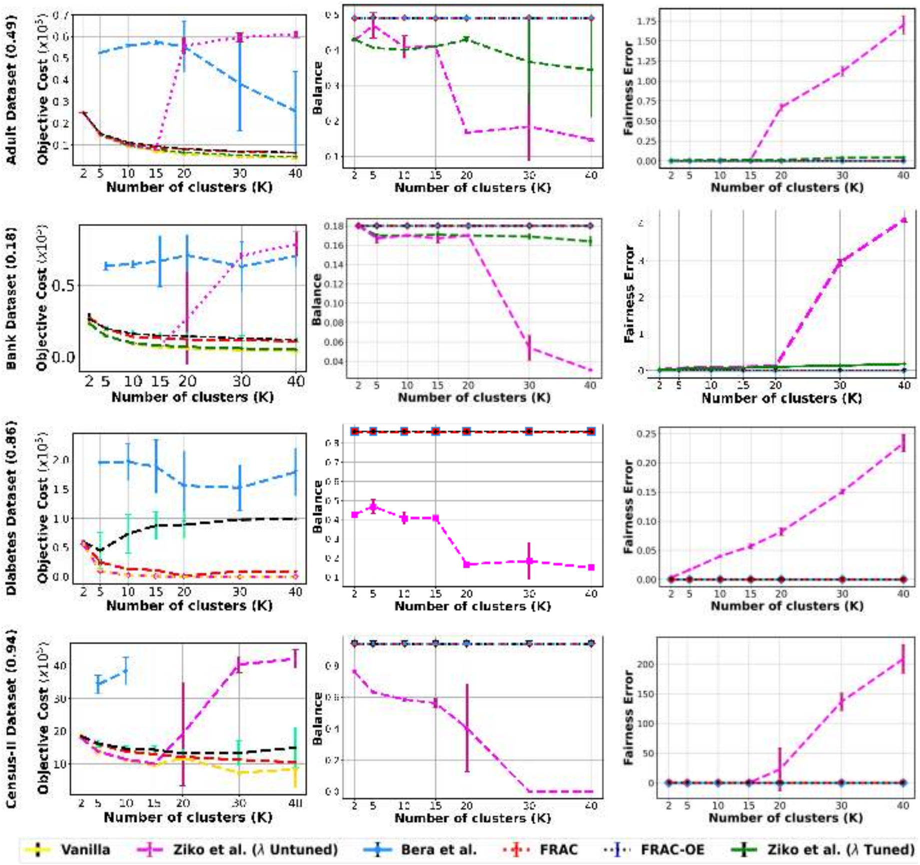


Fig. 7: The line plot shows variation of evaluation metrics over varying number of cluster center for k-means setting. The hyper-tuned variation of Ziko et al. is available only for adult and bank dataset due to expensive computational requirements. For other datasets the hyper-parameter λ is taken same as that is reported in Ziko et al. paper ie. λ=9000, 6000, 6000, 500000 for Adult, Bank, Diabetes and Census II dataset respectively. On the similar reasons Bera et al. results for Census-II are evaluated for k=5 and k=10. (Best viewed in color)

for all the number of clusters show higher objective costs as well as fairness error. This indicates the sensitivity of the approach to the hyper-parameter λ. The proposed approach FRAC gives the best result maintaining a relatively low objective cost without compromising fairness. Similarly, FRAC$_{OE}$ has marginal cost difference from FRAC with same fairness guarantees over most of the datasets showing efficacy of the approach.

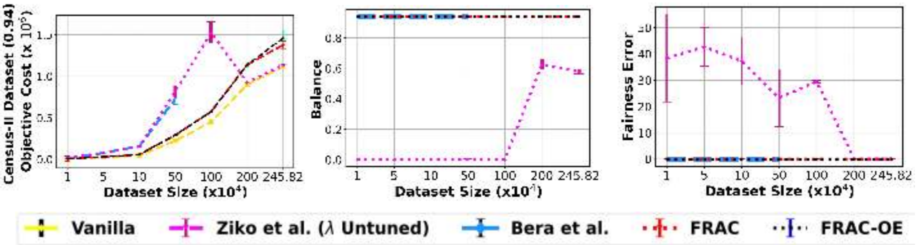


Fig. 8: The line plot shows variation of evaluation metrics over varying data set size for k(=10)-means setting. The hyper-parameter λ=500000 is taken same as that is reported in Ziko et al. paper for Census-II data set due to expensive computational requirements. On the similar reasons Bera et al. results for Census-II are evaluated up to 50×10^4. The target balance for Census-II is evident from plot axes and complete data set size is 245.82×10^4. (Best viewed in color)

7.2 Comparison across varying data set sizes

In this experiment, we measure the performance of k(=10)-means version of different approaches as number of points in data set increases in largest data set – Census-II. Fig. 8. plots the results for evaluation metrics on data set size increasing from 10000 to complete size of 2458285 points. The plot clearly reveals that FRAC, FRAC$_{OE}$, and Bera et al. (2019) are able to maintain strict fairness constraints. But Bera et al. (2019) is able to achieve fairness guarantees at higher objective cost. Due to high computation requirements for Bera et al. (2019) (refer Section 7.4), we limit the results up to $500,000$ number of points. For Ziko et al. (2021), owning to high tuning time (refer run time analysis section 7.4) we use the hyper-parameter value for Census-II same as that reported in Ziko et al. (2021) ie. λ=500000 for complete data set. Though initially Ziko et al. (2021) is having performance close to other approaches but objective cost increases as data set size increases. One reason for this can be the hyper-parameter value used for approach. It may also be noted that, as the data set size reaches to completion, the objective cost improves to that of vanilla clustering but this comes at significant deterioration in fairness metrics. Both Balance and fairness error is quite far from the required target of 0.94 and 0.0 respectively. On the other hand our proposed algorithms FRAC and FRAC$_{OE}$ achieves strict fairness guarantees with slight increase in objective cost from vanilla clustering. Among FRAC and FRAC$_{OE}$, both have marginal difference in objective cost.

7.3 Additional Analysis on Proposed Algorithms

In this section we perform additional study on FRAC and FRAC$_{OE}$ to illustrate their effectiveness.

7.3.1 FRAC vs FRAC$_{OE}$

While FRAC uses round-robin allocation after every clustering iteration, FRAC$_{OE}$ applies the round-robin allocation only at the end of clustering. Both the approaches will result in a fair allocation, but might exhibit different objective costs. We conduct an experiment under the k-means setting with $k = 10$ to study the difference in the objective costs for the two approaches. Like other experiments, we conduct this experiment over ten independent runs and plot the mean objective cost (line) and standard deviation (shaded region) at each iteration over different runs. The plots in Fig. 9 indicates that FRAC has a lower objective cost at convergence than FRAC$_{OE}$. The plot for FRAC$_{OE}$ follows the same cost variation as that of vanilla k-means in the initial phase, but at the end there is a sudden jump that overshoots the cost of FRAC (to accommodate fairness constraints). Thus, applying fairness constraints after every iteration is better than applying it only once at the end. The plot also helps us experimentally visualize the convergence of both FRAC and FRAC$_{OE}$ algorithms. It may be observed that the change in objective cost becomes negligible after a certain number of iterations.

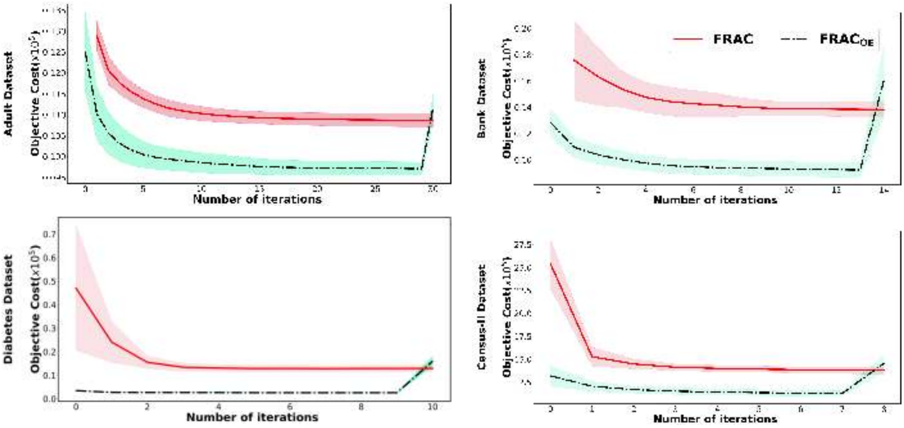
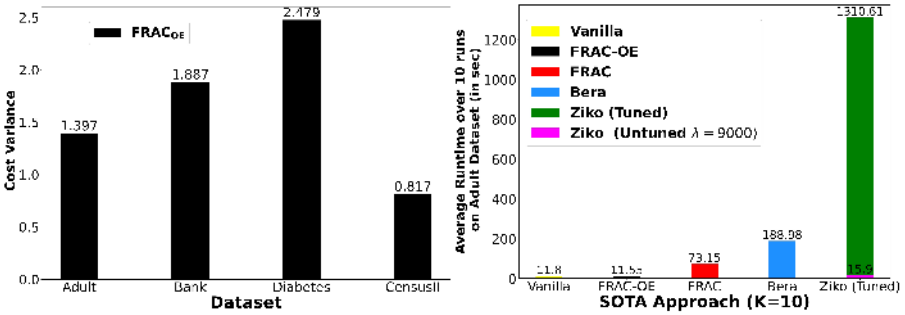


Fig. 9: The cost variation over the iterations for different approaches in k-mean setting is plotted for k=10.

7.3.2 Impact of order in which the centers pick the data points

FRAC assumes an arbitrary order of the centers for allocating data points at every iteration. We verify if the order in which the centers pick the data points impacts the clustering objective cost. We vary the order of the centers picking the data points for the k-mean clustering version with $k = 10$. We report the objective cost variance computed across 100 permutations of the ten



(a)

(b)

Fig. 10: (a) Bar plot shows the variance in objective cost over different 100 random permutations of converged centers returned by standard unfair k-means clustering in FRAC$_{OE}$. (b) k-means runtime analysis of different SOTA approaches on Adult dataset for k=10.

centers. Applying the permutations at every iteration in FRAC is an expensive proposition. Hence we restrict the experiment to the FRAC$_{OE}$ version. The variance of the 100 final converged clustering objective costs (averaged over ten trials) is presented in Fig. 10 (a). It is evident from the plot that the variance is consistently extremely small for all datasets. Thus, we conclude that FRAC$_{OE}$ (and FRAC by extension) is invariant to the order in which the centers pick the data points.

7.3.3 Comparison for τ-ratio on fixed number of clusters(k)

All the experiments till now considered the Balance to be same as the dataset ratio ($\tau_\ell = \frac{1}{k}$). But FRAC and FRAC$_{OE}$ can be used to obtain any desired τ-ratio fairness constraints other than dataset proportion. The results for other τ vector values on k=10 number of clusters are reported in Table 2. We compare the performance of the proposed approach against Bera et al. that also allows for a desired τ-ratio fairness in a restrictive manner. Bera et al. reduces the degree of freedom using δ parameter that controls the lower and upper bound on number of points needed in each cluster belonging to a protected group. Experimentally δ can take values only in terms of dataset proportion r_ℓ for protected group $\ell \in [m]$, i.e. with lower bound as $r_\ell(1-\delta)$ and upper bound as $\frac{r_\ell}{(1-\delta)}$. Further δ needs to be same across all the protected groups making it infeasible to achieve different lower bound for each protected group. Thus Bera et al. cannot be used to have any general fairness constraints for each protected group and can act as baseline only for certain τ_ℓ values. In Table 2 we present results for the τ corresponding to δ=$0.2, 0.8$. Additionally, our algorithms can achieve any generalized τ vectors like $[0.25, 0.12]$, which makes more sense in real-world applications like requiring at least 25% and 12% points in each cluster for males and females. The objective cost obtained by FRAC and FRAC$_{OE}$ is close to Bera et al. (2019) but, the work by Bera et al. (2019) is extendible to multi-valued problem.

Dataset	τ**- vector**	**FRAC**	**FRAC$_{OE}$**	**Bera et al.**	
		Objective Cost	**Objective Cost**	δ **Value**	**Objective Cost**
Adult	<0.133, 0.066 >	9804.65 ± 221.05	9616.51 ± 111.49	0.8	9515.30 ± 19.94
	<0.535, 0.264 >	10010.39 ± 211.27	10011.78 ± 239.73	0.2	9788.73 ± 23.32
	<0.25, 0.12 >	9870.93 ± 261.24	9714.06 ± 157.45	*Cannot be computed*	
Bank	<0.121, 0.056, 0.022 >	9210.38 ± 640.76	9043.51 ± 461.23	0.2	9588.30 ± 48.82
	<0.485, 0.225, 0.089 >	10982.63 ± 1228.28	11317.61 ± 1310.32	0.8	8472.65 ± 37.30
	<0.25, 0.10, 0.04 >	9548.68 ± 540.86	9465.35 ± 476.88	*Cannot be computed*	

Table 2: k-means objective cost for τ-ratio for adult and bank dataset for k=10 clusters.

7.4 Run-time Analysis

Finally, we compare the run-time of the different approaches for the $k(=10)$-means clustering versions on the Adult dataset. The average run-time over 10 different runs is reported in Fig. 10 (b). It can be clearly seen that the run-time of FRAC is significantly better than the fair SOTA approaches. The run-time of Ziko et al. is quite high due to hyper-parameter tuning required to find the best suited λ value. The run-time of Ziko et al. without hyper-parameter tuning is comparable to vanilla clustering. However, without hyper-tuning it has been observed from previous sections that Ziko et al.'s performance can deteriorate significantly on the fairness constraints. FRAC$_{OE}$ runtime has marginal difference from vanilla clustering runtime since FRAC$_{OE}$ applies a single round of fair assignment following vanilla clustering. Bera et al. requires double the time of FRAC. In general, LP formulations to fair clustering are observed to have higher complexities. In contrast, FRAC is able to achieve better objective costs and comparable fairness measures with significantly less complexity.

Motivated by Kriegel et al. (2017), we further study the runtime behaviour across varying number of datapoints and varying number of clusters. For the scalablity study, we perform the analysis using Census-II as it is largest dataset.

7.4.1 Runtime comparison with number of cluster(k)

In this study we conduct experiment to find the variation in runtime as number of clusters k varies from 2 to 40. We observe the results for $2, 5, 10, 15, 20, 30$ and 40. From the results summarized in Fig. 11, we can observe that Bera et al. (2019) is having significantly high execution time. Thus, we limit the results upto $k(=5, 10)$-clustering. As pointed out in previous section Bera et al. (2019), LP fails to converge for k=2.

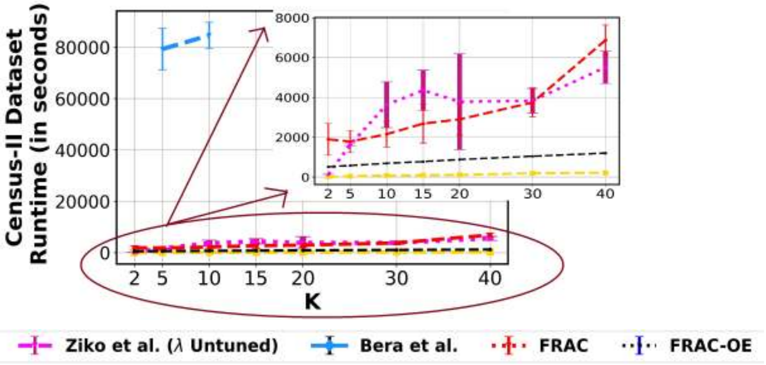


Fig. 11: The line plot shows variation of runtime over varying number of clusters(k) for k-means setting on complete dataset size. The hyper-parameter λ=500000 is taken same as that is reported in Ziko et al. paper for Census-II dataset due to expensive computational requirements. On the similar reasons Bera et al. results for Census-II are evaluated for k=5 and k=10. For better visualization the results are zoomed out for approaches other than Bera et al. (Best viewed in color)

We can clearly see from the plots that FRAC$_{OE}$ has runtime close to vanilla clustering. For Ziko et al. (2021), even in untuned version (using same hyper-parameter as reported in Ziko et al. paper) we still have runtime close to proposed FRAC. Tuning the hyper-parameter will result in significant increase in overall runtime for the approach as observed in Section 7.4.

7.4.2 Runtime comparison across varying data set size

We study the scalability of different approaches to increase in the data set size. We conduct experiments using the largest data set, Census-II at k=10. For Bera et al. (2019), plots in Fig. 12 reveal that the run time significantly increases with 50×10^4 points in the data set. So we limit the study up to this size. The run time for untuned Ziko et al. (2021) is close to vanilla clustering. However, the gap starts to widen after a certain number of data points. On the contrary, our proposed FRAC$_{OE}$ follows a similar trend close to vanilla and does not deteriorate with the varying number of clusters showing the efficiency of FRAC$_{OE}$. The FRAC being an in-processing heuristic has a run time larger than vanilla clustering but is comparable to untuned Ziko et al. (2021). Tuning the Ziko et al. (2021) will result in additional overhead.

8 Discussion

We proposed a novel τ-ratio fairness notion. The new notion generalizes the existing Balance notion and admits an efficient round-robin algorithm to the corresponding fair assignment problem. We also showed that our proposed

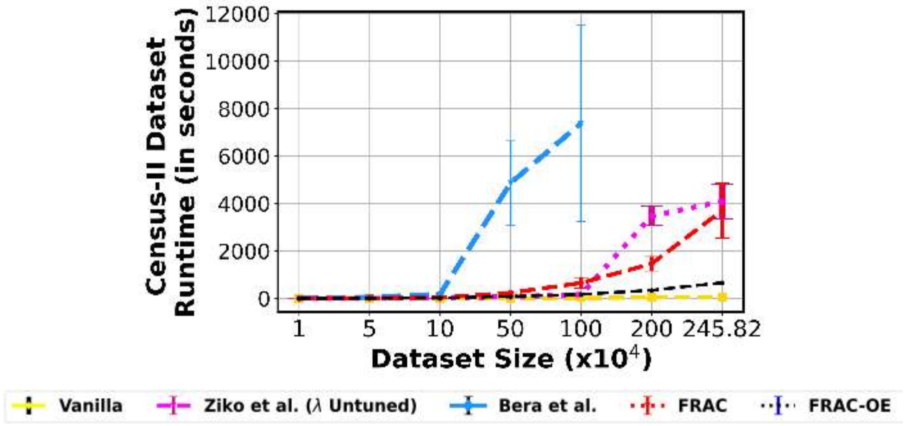


Fig. 12: The line plot shows variation of runtime over varying dataset size (upto complete dataset size of 245.82×10^4) for k=10-means setting. The hyper-parameter λ=500000 is taken same as that is reported in Ziko et al. paper for Census-II dataset due to expensive computational requirements. On the similar reasons Bera et al. results for Census-II are evaluated for dataset size of $10,000$, $50,000$ and 100000. (Best viewed in color)

algorithm, FRAC$_{OE}$, (i) achieves $2(\alpha + 2)$-approximate solution up to three clusters, and (ii) achieves $2^{k-1}(\alpha+2)$-approximate guarantees to general k with τ=$1/k$. Current proof techniques for $k \leq 3$ requires intricate case analysis which becomes intractable for larger k. However, our experiments show that FRAC outperforms SOTA approaches in terms of objective cost and fairness measures even for k >3. We also proof the cost approximation for general τ vector and show convergence analysis for FRAC$_{OE}$. An immediate future direction is to analytically prove $2(\alpha + 2)$-approximation guarantee for general k.

It is worth noting here that the τ-ratio fairness ensures the Balance property. However, if one is to use Balance as a constraint, one could get a better approximation guarantee. Surprisingly, we observe from our experiments that this is not the case. We leave the theoretical and experimental analysis of these two notions of fairness in the presence of large data as an interesting future work. Apart from above mentioned immediate future directions, extending the current work to multi-valued multiple protected attributes similar to the one proposed by Bera et al. (2019), or achieving the notion of individual fairness along while maintaining group fairness are also interesting research problems.

Declaration

Funding: The research is funded by Department of Science & Technology, India under grant number SRG/2020/001138 (Recipient name- Dr. Shweta Jain).

Conflicts of interest/Competing interests: No potential competing interest was reported by the authors.

Availability of data and material: All datasets used in the experiments are publicly available on UCI repository.

Code availability: The code has been made publicly available at https://github.com/shivi98g/Fair-k-means-Clustering-via-Algorithmic-Fairness

Ethics approval: Not applicable

Consent for publication : The paper is the authors' own original work, which has not been previously published elsewhere. The paper is not currently being considered for publication elsewhere. The paper reflects the authors' own research and analysis in a truthful and complete manner. The paper properly credits the meaningful contributions of co-authors and co-researchers.

References

Abbasi, M., A. Bhaskara, and S. Venkatasubramanian 2021. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 504–514.

Abraham, S.S., D. Padmanabhan, and S.S. Sundaram 2020. Fairness in clustering with multiple sensitive attributes. In *EDBT/ICDT 2020 Joint Conference*, pp. 287–298.

Ahmadian, S., A. Epasto, R. Kumar, and M. Mahdian 2019. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 267–275.

Ahmadian, S., A. Epasto, R. Kumar, and M. Mahdian 2020, 26–28 Aug. Fair correlation clustering. In S. Chiappa and R. Calandra (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Volume 108 of *Proceedings of Machine Learning Research*, pp. 4195–4205. PMLR.

Anderson, N., S.K. Bera, S. Das, and Y. Liu. 2020. Distributional individual fairness in clustering. *arXiv:2006.12589* .

Anegg, G., H. Angelidakis, A. Kurpisz, and R. Zenklusen 2020. A technique for obtaining true approximations for k-center with covering constraints. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 52–65. Springer.

Backurs, A., P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner 2019. Scalable fair clustering. In *International Conference on Machine Learning*, pp. 405–413. PMLR.

Bandyapadhyay, S., F.V. Fomin, and K. Simonov. 2020. On coresets for fair clustering in metric and euclidean spaces and their applications. *arXiv:2007.10137* .

Bandyapadhyay, S., T. Inamdar, S. Pai, and K. Varadarajan. 2019. A constant approximation for colorful k-center. *arXiv:1907.08906* .

Banerjee, A. and J. Ghosh. 2006. Scalable clustering algorithms with balancing constraints. *Data Mining and Knowledge Discovery 13*(3): 365–395 .

Barocas, S. and A.D. Selbst. 2016. Big data's disparate impact. *CALIFORNIA LAW REVIEW*: 671–732 .

Baumann, E. and J.L. Rumberger. 2018. State of the art in fair ML: from moral philosophy and legislation to fair classifiers. *CoRR* abs/1811.09539. https://arxiv.org/abs/1811.09539 .

Bera, S., D. Chakrabarty, N. Flores, and M. Negahbani. 2019. Fair algorithms for clustering. *Advances in Neural Information Processing Systems* 32: 4954–4965 .

Bercea, I.O., M. Groß, S. Khuller, A. Kumar, C. Rösner, D.R. Schmidt, and M. Schmidt. 2018. On the cost of essentially fair clusterings. *arXiv:1811.10319* .

Böhm, M., A. Fazzone, S. Leonardi, and C. Schwiegelshohn. 2020. Fair clustering with multiple colors. *arXiv:2002.07892* .

Bose, A. and W. Hamilton 2019. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pp. 715–724. PMLR.

Bottou, L. and Y. Bengio. 1994. Convergence properties of the k-means algorithms. *Advances in neural information processing systems* 7 .

Brubach, B., D. Chakrabarti, J. Dickerson, S. Khuller, A. Srinivasan, and L. Tsepenekas 2020. A pairwise fair and community-preserving approach to k-center clustering. In *International Conference on Machine Learning*, pp. 1178–1189. PMLR.

Byrka, J., T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh 2014. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pp. 737–756. SIAM.

Carey, A.N. and X. Wu. 2022. The fairness field guide: Perspectives from social and formal sciences. *arXiv:2201.05216* .

Chakrabarti, D., J.P. Dickerson, S.A. Esmaeili, A. Srinivasan, and L. Tsepenekas 2022. A new notion of individually fair clustering: α-equitable k-center. In *International Conference on Artificial Intelligence and Statistics*, pp. 6387–6408. PMLR.

Chan, T.H., A. Guerqin, and M. Sozio 2018. Fully dynamic k-center clustering. In *Proceedings of the 2018 World Wide Web Conference*, pp. 579–587.

Chen, X., B. Fain, L. Lyu, and K. Munagala 2019a. Proportionally fair clustering. In *International Conference on Machine Learning*, pp. 1032–1041. PMLR.

Chen, X., B. Fain, L. Lyu, and K. Munagala 2019b. Proportionally fair clustering. In *International Conference on Machine Learning*, pp. 1032–1041. PMLR.

Chhabra, A., K. Masalkovaitė, and P. Mohapatra. 2021. An overview of fairness in clustering. *IEEE Access* 9: 130698–130720. https://doi.org/10.1109/ACCESS.2021.3114099 .

Chhabra, A., A. Singla, and P. Mohapatra. 2021. Fair clustering using antidote data. *arXiv:2106.00600* .

Chierichetti, F., R. Kumar, S. Lattanzi, and S. Vassilvitskii 2017. Fair clustering through fairlets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5036–5044.

Chikahara, Y., S. Sakaue, A. Fujino, and H. Kashima 2021. Learning individually fair classifier with path-specific causal-effect constraint. In *International Conference on Artificial Intelligence and Statistics*, pp. 145–153. PMLR.

Cho, J., G. Hwang, and C. Suh 2020. A fair classifier using mutual information. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2521–2526.

Correa, J., A. Cristi, P. Duetting, and A. Norouzi-Fard 2021. Fairness and bias in online selection. In *International Conference on Machine Learning*, pp. 2112–2121. PMLR.

Dastin, J. 2018. Amazon scraps secret ai recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G. [Online; accessed 15-August-2021].

Davidson, I. and S. Ravi 2020. Making existing clusterings fairer: Algorithms, complexity results and insights. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 34, pp. 3733–3740.

Deepak and S.S. Abraham. 2020, Jun. Representativity fairness in clustering. *12th ACM Conference on Web Science* .

Deepak, J.M. Jose, and S. V 2020. Fairness in unsupervised learning. In *Proceedings of the 29th ACM International Conference on Information &amp; Knowledge Management*, CIKM '20, New York, NY, USA, pp. 3511–3512. Association for Computing Machinery.

Dwork, C., M. Hardt, T. Pitassi, O. Reingold, and R. Zemel 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226.

Elzayn, H., S. Jabbari, C. Jung, M. Kearns, S. Neel, A. Roth, and Z. Schutzman 2019. Fair algorithms for learning in allocation problems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 170–179.

Esmaeili, S., B. Brubach, A. Srinivasan, and J. Dickerson. 2021. Fair clustering under a bounded cost. *Advances in Neural Information Processing Systems* 34: 14345–14357 .

Esmaeili, S., B. Brubach, L. Tsepenekas, and J. Dickerson. 2020. Probabilistic fair clustering. *Advances in Neural Information Processing Systems* 33: 12743–12755 .

Feng, Z., P. Kacham, and D. Woodruff 2021. Dimensionality reduction for the sum-of-distances metric. In *International Conference on Machine Learning*, pp. 3220–3229. PMLR.

Ghadiri, M., S. Samadi, and S. Vempala 2021. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 438–448.

Gong, S., X. Liu, and A.K. Jain 2021. Mitigating face recognition bias via group adaptive classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3414–3424.

Goyal, D. and R. Jaiswal. 2021. Tight fpt approximation for socially fair clustering. *arXiv:2106.06755* .

Harb, E. and H.S. Lam. 2020. Kfc: A scalable approximation algorithm for k-center fair clustering. *Advances in Neural Information Processing Systems* 33: 14509–14519 .

Huang, L., S. Jiang, and N. Vishnoi. 2019. Coresets for clustering with fairness constraints. *Advances in Neural Information Processing Systems* 32: 7589–7600 .

Jia, X., K. Sheth, and O. Svensson 2020. Fair colorful k-center clustering. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 209–222. Springer.

Jones, M., H. Nguyen, and T. Nguyen 2020. Fair k-centers via maximum matching. In *International Conference on Machine Learning*, pp. 4940–4949. PMLR.

Julia, A., J. Larson, S. Mattu, and L. Kirchner. 2016. Propublica–machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. [Online; accessed 13-August-2021].

Jung, C., S. Kannan, and N. Lutz. 2020. Service in your neighborhood: Fairness in center location. *Foundations of Responsible Computing (FORC)* .

Kalyanakrishnan, S. 2016. *k*-means clustering. https://www.cse.iitb.ac.in/~shivaram/teaching/old/cs344+386-s2017/resources/classnote-2.pdf. [Online; accessed 29-May-2022].

Kar, D., S. Medya, D. Mandal, A. Silva, P. Dey, and S. Sanyal. 2021. Feature-based individual fairness in k-clustering. *arXiv:2109.04554* .

Kleindessner, M., P. Awasthi, and J. Morgenstern 2019. Fair k-center clustering for data summarization. In *International Conference on Machine Learning*, pp. 3448–3457. PMLR.

Kleindessner, M., P. Awasthi, and J. Morgenstern. 2020. A notion of individual fairness for clustering. *arXiv:2006.04960* .

Kleindessner, M., S. Samadi, P. Awasthi, and J. Morgenstern 2019. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, pp. 3458–3467. PMLR.

Krause, A. 2016. Clustering and *k*-means. https://las.inf.ethz.ch/courses/lis-s16/hw/hw4_sol.pdf. [Online; accessed 29-May-2022].

Kriegel, H.P., E. Schubert, and A. Zimek. 2017, aug. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowl. Inf. Syst. 52*(2): 341–378. https://doi.org/10.1007/s10115-016-1004-2 .

Le Quy, T., A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi. 2022. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary*

Reviews: Data Mining and Knowledge Discovery: e1452 .

Lee, J.K., Y. Bu, D. Rajan, P. Sattigeri, R. Panda, S. Das, and G.W. Wornell 2021. Fair selective classification via sufficiency. In *International Conference on Machine Learning*, pp. 6076–6086. PMLR.

Li, B., L. Li, A. Sun, C. Wang, and Y. Wang 2021. Approximate group fairness for clustering. In *International Conference on Machine Learning*, pp. 6381–6391. PMLR.

Li, P., H. Zhao, and H. Liu 2020, June. Deep fair clustering for visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Liu, S. and L.N. Vicente. 2021. A stochastic alternating balance k-means algorithm for fair clustering. *arXiv:2105.14172* .

Lohaus, M., M. Perrot, and U.V. Luxburg 2020, 13–18 Jul. Too relaxed to be fair. In H. D. III and A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning*, Volume 119 of *Proceedings of Machine Learning Research*, pp. 6360–6369. PMLR.

Mahabadi, S. and A. Vakilian 2020. Individual fairness for k-clustering. In *International Conference on Machine Learning*, pp. 6586–6596. PMLR.

Makarychev, Y. and A. Vakilian. 2021. Approximation algorithms for socially fair clustering. *arXiv:2103.02512* .

Mehrabi, N., F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. 2021, jul. A survey on bias and fairness in machine learning. *ACM Comput. Surv. 54*(6). https://doi.org/10.1145/3457607 .

Micha, E. and N. Shah 2020. Proportionally fair clustering revisited. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Negahbani, M. and D. Chakrabarty. 2021. Better algorithms for individually fair k-clustering. *Advances in Neural Information Processing Systems* 34: 13340–13351 .

Ntoutsi, E., P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, et al. 2020. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10*(3): e1356 .

Padmanabhan, D. 2020. Whither fair clustering? In *AI for Social Good: Harvard CRCS Workshop.*

Quy, T.L., A. Roy, G. Friege, and E. Ntoutsi. 2021. Fair-capacitated clustering. *arXiv:2104.12116* .

Ranzato, F., C. Urban, and M. Zanella. 2021. Fair training of decision tree classifiers. *arXiv:2101.00909* .

Rösner, C. and M. Schmidt. 2018. Privacy preserving clustering with constraints. *arXiv:1802.02497* .

Schmidt, M., C. Schwiegelshohn, and C. Sohler 2019. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pp. 232–251. Springer.

Schmidt, M. and J. Wargalla. 2021. Coresets for constrained k-median and k-means clustering in low dimensional euclidean space. *arXiv:2106.07319* .

Thejaswi, S., B. Ordozgoiti, and A. Gionis 2021. Diversity-aware k-median: Clustering with fair center representation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 765–780. Springer.

Vakilian, A. and M. Yalciner 2022. Improved approximation algorithms for individually fair clustering. In *International Conference on Artificial Intelligence and Statistics*, pp. 8758–8779. PMLR.

Zhang, W., A. Bifet, X. Zhang, J.C. Weiss, and W. Nejdl. 2021. Farf: A fair and adaptive random forests classifier, *Advances in Knowledge Discovery and Data Mining*, 245–256. Springer International Publishing.

Ziko, I.M., J. Yuan, E. Granger, and I.B. Ayed 2021. Variational fair clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 35, pp. 11202–11209.