



Efficient algorithms for fair clustering with a new notion of fairness

Shivam Gupta¹ · Ganesh Ghalme² · Narayanan C. Krishnan³ · Shweta Jain¹

Received: 31 August 2021 / Accepted: 6 February 2023 / Published online: 20 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2023

Abstract

We revisit the problem of fair clustering, first introduced by Chierichetti et al. (Fair clustering through fairlets, 2017), which 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 objectives and fairness. In this paper, we propose a new notion of fairness which we call τ -ratio fairness, that strictly generalizes the Balance property and enables a fine-grained efficiency vs. fairness trade-off. Furthermore, we show that a simple greedy round-robin-based algorithm achieves this trade-off efficiently. Under a more general setting of multi-valued protected attributes, we rigorously analyze the theoretical properties of the proposed algorithm, the Fair Round-Robin Algorithm for Clustering Over-End (FRAC_{OE}). We also propose a heuristic algorithm, Fair Round-Robin Algorithm for Clustering (FRAC), that applies round-robin allocation at each iteration of a vanilla clustering algorithm. Our experimental results suggest that both FRAC and FRAC_{OE} outperform all the state-of-the-art algorithms and work exceptionally well even for a large number of clusters.

Keywords Fairness · Clustering · Machine learning · Unsupervised learning

Responsible editor: T. Calders, E. Ntoutsi, M. Pechenizkiy, B. Rosenhahn, S. Ruggieri

✉ Shivam Gupta
shivam.20csz0004@iitrpr.ac.in

Ganesh Ghalme
ganeshghalme@ai.iith.ac.in

Narayanan C. Krishnan
ckn@iitpkd.ac.in

Shweta Jain
shwetajain@iitrpr.ac.in

¹ Indian Institute of Technology, Ropar, India

² Indian Institute of Technology Hyderabad, Kandi, India

³ Indian Institute of Technology, Palakkad, India

1 Introduction

Machine learning (ML) research advances have resulted in the development of increasingly accurate models. These advancements lead to the widespread adoption of ML algorithms in applications ranging from self-driving cars, loan approvals, criminal risk prediction, college admissions, and health risk prediction. The primary objective of these algorithms is to obtain improved accuracy and predictive performance. However, their use to allocate social goods and opportunities such as access to healthcare, job, and education warrants a closer look at the societal impacts of their outcomes (Carey and Wu 2022; Ntoutsis et al. 2020). Recent studies have exposed a discriminatory outlook on the outcomes of these algorithms. The adverse societal effects include 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 impact in the real world.

While fairness in supervised learning is well-studied (Dwork et al. 2012; Correa et al. 2021; Chikahara et al. 2021; Lee et al. 2021; Mehrabi et al. 2021; Le Quy et al. 2022), fairness in unsupervised learning is still in its formative stages (Deepak et al. 2020; Chhabra et al. 2021; Harris et al. 2019). 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.

To emphasize the importance of fairness in unsupervised learning, we consider the following example: An employee-friendly company is looking to open multiple branches across the city and distribute its workforce in these branches. The goal is to improve work efficiency and minimize overall travel time to work. The company has employees with diverse backgrounds (race and gender). 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.

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. Chierichetti et al. (2017) proposed the first fairness notion in unsupervised clustering, wherein each cluster must exhibit a Balance. The Balance represents the ratio of data points with different values of the protected attribute in each cluster. 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) improves efficiency but does not facilitate an 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 fairness. The τ -ratio fairness ensures a certain fraction of data points for a given protected attribute in each cluster. 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. Fourth, it is straightforward for the algorithm designer to input the requirement into the algorithm as constraints. And fifth, it is easy to interpret and evaluate it from the output. In our running example, if a company wants to have a minimum fraction of employees from each group in every branch (clusters), then one can specify it in the form of a vector τ of size equal to the number of data points needed from each group. The contributions of our work are summarized in the following section:

1.1 Our contribution

Conceptual contribution We introduce a new notion of fairness which we call τ -ratio fairness and show that any algorithm satisfying a τ -ratio fairness also satisfies the Balance property (Theorem 4). We also show that sometimes one can obtain a degenerate value of τ -ratio fairness using Balance. We propose two simple and efficient round-robin-based algorithms for the τ -ratio fair allocation problem, namely, FRAC_{OE} (see Sect. 4) and a heuristic algorithm called FRAC (Sect. 6). Our algorithm FRAC_{OE} uses the unconstrained clustering algorithm (referred to as vanilla clustering) as a black-box implementation and modifies its output appropriately to ensure τ -ratio fairness. The fairness guarantee is deterministic and verifiable, i.e., it holds for every algorithm run 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 algorithm, FRAC_{OE} . We first show that FRAC_{OE} achieves a $2(\alpha + 2)$ -approximate for clustering instances up to three clusters (Theorem 7 and Lemma 11) to optimal fair clustering cost for $\tau = 1/k$ which corresponds to maximally balanced clusters. Here, α is a clustering algorithm-specific constant. That is, given a fair clustering instance with $k \leq 3$ clusters and n data points, our proposed algorithm returns an allocation with an objective cost of at most $2(\alpha + 2)$ times the objective cost of optimal assignment to maximally balanced clusters. 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 Sect. 5.2). We also theoretically analyze the convergence of FRAC_{OE} (Lemma 14). The time complexity of FRAC_{OE} is $O(kn \log(n))$ where n is the size of the dataset, and k is the number of clusters.

Experimental contributions Through extensive experiments on four datasets (Adult, Bank, Diabetes, and Census II), we show that the proposed algorithms, FRAC and FRAC_{OE} outperform all the existing algorithms on fairness and objective costs. Perhaps the most critical insight from our experiments is that the performance of our proposed algorithms does not deteriorate with increasing k , validating our conjecture. Experiments also show that while we do not have convergence guarantees for the heuristic algorithm FRAC, it does converge on all the datasets and performs slightly better than FRAC_{OE} . Thus, making it suitable for practical applications. We compare our algorithms with SOTA algorithms for their fairness guarantee, objective cost, and runtime analysis. We also remark that our algorithms do not require hyper-parameter tuning, making our method easy to train and scalable. While our algorithms apply to the center-based clustering approach, we demonstrate its efficacy using k -means and k -median.

2 Related work

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). However, the research on fair clustering is still in its infancy and rapidly gathering attention (Chierichetti et al. 2017; Kleindessner et al. 2019; Brubach et al. 2021; Liu and Vicente 2021; Davidson and Ravi 2020; Bercea et al. 2018; Le Quy and Ntoutsu 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. 2019), proposing new problem-specific fairness notions such as social fairness (Abbasi et al. 2021; Makarychev and Vakilian 2021), characterizing the fairness v/s efficiency trade-off (Ziko et al. 2021; Abraham et al. 2020), and developing, analyzing efficient fair algorithms (Bandyapadhyay et al. 2020; Schmidt et al. 2019). We now categorize the literature on fairness in clustering based on 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, define fairness in clustering through a Balance property. Chierichetti et al. (2017) achieve balanced clustering by partitioning the data into balanced sets called fairlets. These fairlets are then merged while maintaining the Balance property in each merge operation. Backurs et al. (2019) propose an efficient algorithm to compute the fairlets. Both the above approaches have two major drawbacks: they only work for binary-valued protected attributes and can only create clusters exhibiting the exact dataset ratio. Schmidt et al. (2019) 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) reduce this exponential size requirement to linear in terms of space, but it still has the running complexity that is exponential in the number of clusters. Chhabra et al. (2021) propose a pre-processing technique by adding a small number of extra points. Our algorithms are efficient in terms of space and time complexity.

In-processing: Böhm et al. (2020) propose an $(\alpha+2)$ -approximate algorithm for fair clustering using a minimum cost-perfect matching algorithm. The approach works with a multi-valued protected attribute but 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 to 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 leading to a high computational cost. Further, the clustering objective also deteriorates significantly under strict fairness constraints when dealing with many clusters (refer Sect. 7.1). Along the same lines, Abraham et al. (2020) devise an optimization-based approach for fair clustering with multiple multi-valued protected attributes. It has a trade-off hyper-parameter similar to Ziko et al. (2021).

Post-processing: Our proposed algorithm FRAC_{OE} follows the post-processing approach. Bera et al. (2019) solved the fair clustering problem via a fair assignment problem and formulated a linear programming (LP) based solution. The LP-based formulation leads to a higher execution time (refer to Sect. 7.4). Also, the approach fails to converge in a reasonable time for larger datasets. Our proposed approach takes a similar route as Bera et al. (2019) and transforms the fair clustering problem into a fair assignment problem. 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. Harb and Lam (2020) extended the fair clustering problem to the k -center problem, whereas we consider k -means and k -median based centering techniques. There are other works that are applicable only for k -center clustering (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).

Other related work: While we focus on the fairness notion of Balance, other perspectives on fairness are also defined in the literature. These include individual fairness (Kleindessner et al. 2020), proportionality fairness (Chen et al. 2019; Mahabadi and Vakilian 2020; Vakilian and Yalciner 2022; Negahbani and Chakrabarty 2021; Han et al. 2022), and social fairness (Ghadiri et al. 2021; Abbasi et al. 2021; Ghadiri et al. 2022; Deepak and Abraham 2020; Makarychev and Vakilian 2021; Goyal and Jaiswal 2021; Chlamtáč et al. 2022).

Another line of related work in fair clustering revolves around hierarchical clustering, spectral clustering algorithms for graphs, deep clustering (Zhang and Davidson

2021; Wang and Davidson 2019; Song et al. 2021), and hypergraph clustering (Bose and Hamilton 2019; Kleindessner et al. 2019). Jones et al. (2020) define fairness in 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. 2019). Recently, Li et al. (2021) proposed a new notion of core fairness which 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 that of balanced (in terms of the number of points in each cluster) clustering (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 these fairness constraints.

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 -clustering¹ 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 $\phi : X \rightarrow [k]$ that maps each point to the corresponding cluster. Throughout this paper, we consider that each point, $x_i \in X$, is associated with a *single* protected attribute $\rho(x_i)$ (say ethnicity from a pool of other available protected attribute). Let the protected attribute takes values from the set of m values denoted by $[m]$. The number of distinct protected attribute values is finite and much smaller than the size of X .² Furthermore, let $d : X \times X \rightarrow \mathbb{R}_+$ be a distance metric defined on X that measures the dissimilarity between features. Additionally, we are also given a vector $\tau = \{\tau_\ell\}_{\ell=1}^m$, where each component τ_ℓ satisfies $0 \leq \tau_\ell \leq \frac{1}{k}$. The τ vector 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 specify an m -dimensional vector with values between 0 and $1/k$ as the fairness target. Also, let us denote X_ℓ and n_ℓ as set and number of points respectively corresponding to the points having protected attribute value ℓ in X . Let $\mathbb{I}(\cdot)$ denote the indicator function. Vanilla (an unconstrained) clustering algorithm determines the cluster centers to minimize the following clustering objective cost:

Definition 1 (*Objective cost*). Given $p > 0$, the cluster objective cost with respect to the metric space (X, d) is defined as:

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.

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

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

Group Fairness Notions: We begin with first defining the most popular notion of group fairness Balance. The notion was 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

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

Balance is computed by finding the minimum possible ratio of one value of the protected attribute (say, male) to the other value of the protected attribute (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 ratio. Further, the clusters maximizing the Balance are not unique. Bera et al. (2019) proposed a generalization of the Balance notion to multi-valued protected attributes in terms of cluster sizes by providing the lower and upper bounds on the number of points from each group in every cluster.

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

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

Definition 4 (Restricted dominance) A clustering \mathcal{C} is τ -RD if

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

We remark here that minority protection provides the lower bound on the number of points from each protected group in every cluster, whereas restricted dominance provides the upper bound. For binary protected attribute with $\tau_a = \tau_b = \min_{a,b} \frac{n_a}{n_b}$, satisfying τ -RD and τ -MP together is same as obtaining a Balance property. We now

define our proposed τ -ratio fairness notion, which ensures that each cluster is assigned at least 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

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

Our first theorem (Theorem 4) in Sect. 5 shows that an algorithm satisfying τ -ratio fairness notion produces a set of clusters that maximizes the Balance. In particular, when $\tau_\ell = \frac{1}{k}$, then τ -ratio fairness achieves the Balance equal to the dataset ratio. We also show that a maximally balanced cluster need not imply τ -ratio fairness for arbitrary τ (Lemma 6 in Sect. 5). Hence τ -ratio is a more generalized fairness notion. We now define the fair clustering problem for 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 from a vanilla 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 $\phi : X \rightarrow [k]$ 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 of a fair clustering problem and \mathcal{T} an instance of τ -ratio fair assignment problem after applying an α -approximate solution to the vanilla clustering problem, then $\mathcal{OPT}_{assign}(\mathcal{T}) \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.*

Proof Let C be the cluster centers obtained by running a vanilla clustering algorithm on instance \mathcal{I} . We prove the lemma by constructing a τ -ratio assignment ϕ' that satisfies $L_p(X, C, \phi') \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I}) \implies \mathcal{OPT}_{assign}(\mathcal{T}) \leq L_p(X, C, \phi') \leq (\alpha + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.

Construction of ϕ' : 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 $x_i \in X$, define $\phi'(x_i) = nrst(\phi^*(x_i))$. Then we have the following two claims:

Claim 2 ϕ' satisfies τ -ratio fairness.

Proof Let the set of points having protected attribute value ℓ in cluster $c^* \in C^*$ be $n_\ell(c^*)$. Since (C^*, ϕ^*) satisfy τ -ratio fairness then using Definition 5 we have

$$|n_\ell(c^*)| \geq \tau_\ell n_\ell \quad \forall c^* \in C^*.$$

Now, for any center $c \in C$ belonging to vanilla clustering, we will find the set of all centers in optimal solution (C^*) that are nearest to c . Let us denote this set by $N(c) = \{c^* \in C^* : nrst(c^*) = c\}$ Then the way ϕ' is defined, we have, $\forall c$:

$$|\{x_i \in X_\ell : \phi'(x) = c\}| = |\cup_{c^* \in N(c)} n_\ell(c^*)|$$

Now as each center c^* satisfies τ -ratio fairness, so the union over combined assignments will also satisfy τ -ratio fairness i.e. $|\cup_{c^* \in N(c)} n_\ell(c^*)| \geq n_\ell \tau_\ell$. □

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

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, the algorithm has two issues. Firstly, the time complexity is high (as seen from the experiments in Sect. 7.4). Secondly, the solution obtained is not easy to interpret.³ We propose a simple round-robin (easily interpretable) algorithm for a 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}) first runs a vanilla clustering algorithm to produce the initial clusters $\mathcal{C} = (C, \phi)$. It then makes 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, then the new assignment function $\hat{\phi}$ is computed according to FAIRASSIGNMENT procedure in Algorithm 2.

Algorithm 2 iteratively allocates the data points concerning each protected attribute value. To recollect X_ℓ and n_ℓ denote the set and the number of data points having ℓ as the protected attribute value, respectively. The algorithm allocates $\lfloor \tau_\ell n_\ell \rfloor$ number of points⁴ to each cluster in a round-robin fashion as follows. Let $\{c_1, c_2, \dots, c_k\}$ be a random ordering of the cluster centers. At each round t , each center c_j picks

³ It is unclear why a certain assignment to a specific cluster helps maintain a fairness guarantee?

⁴ For the sake of simplicity, we assume $\tau_\ell n_\ell \in \mathbb{N}$ and ignore the floor notation.

Algorithm 1: τ -FRAC_{OE}

Input: set of datapoints X , number of clusters k , fairness requirement vector τ , range of protected attribute values m , clustering objective norm p

Output: cluster centers \hat{C} and assignment function $\hat{\phi}$

- 1 Solve the vanilla (k, p) -clustering problem and let (C, ϕ) be the solution obtained.
- 2 **if** τ -ratio fairness is met **then**
- 3 return (C, ϕ)
- 4 **else**
- 5 $(\hat{C}, \hat{\phi}) = \text{FAIRASSIGNMENT}(C, X, k, \tau, m, p, \phi)$
- 6 return $(\hat{C}, \hat{\phi})$
- 7 **end**
- 8 **end**

Algorithm 2: FAIRASSIGNMENT

Input: cluster centers C , set of datapoints X , number of clusters k , fairness requirement vector τ , range of protected attribute values m , clustering objective norm p , assignment function ϕ

Output: cluster centers \hat{C} and assignment function $\hat{\phi}$

- 1 Fix a random ordering on centers and let the centers are numbered from 1 to k with respect to this random ordering.
- 2 Initialize $\hat{\phi}(x_i) \leftarrow 0 \forall x_i \in X$.
- 3 **for** $\ell \leftarrow 1$ **to** m **do**
- 4 $n_\ell \leftarrow$ number of datapoints having value of protected attribute ℓ .
- 5 $X_\ell \leftarrow$ set of datapoints having value of protected attribute ℓ .
- 6 **for** $t \leftarrow 1$ **to** $\tau_\ell n_\ell$ **do**
- 7 **for** $j \leftarrow 1$ **to** k **do**
- 8 $x_{min} \leftarrow \operatorname{argmin}_{x_i \in X_\ell: \hat{\phi}(x_i)=0} d(x_i, c_j)$
- 9 $\hat{\phi}(x_{min}) = j$
- 10 **end**
- 11 **end**
- 12 For all $x_i \in X_\ell$ such that $\hat{\phi}(x_i) = 0$, set $\hat{\phi}(x_i) = \phi(x_i)$
- 13 **end**
- 14 Recompute the centers \hat{C} with respect to the new allocation function $\hat{\phi}$.
- 15 Return $(\hat{C}, \hat{\phi})$.

the point x_i of its preferred choice from X_ℓ i.e. $\hat{\phi}(x_i) = j$. Once the τ_ℓ fraction of points are assigned to the centers, i.e., after $\tau_\ell n_\ell$ 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. The runtime 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 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 data points, respectively. Suppose an allocation returned by a clustering algorithm satisfies τ -ratio guarantee, then the Balance of the given allocation is at least $\frac{\tau_a n_a}{n_b(1-k\tau_b+\tau_b)}$.*

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

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

Here, the lower bound comes directly from the fairness definition, and the upper bound from the fact that all the clusters together will be allocated at least $k\tau_a n_a$ number of points. The extra points a particular cluster can take are 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

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

□

We remark here that the notion of Balance is concerned with allocating the points to clusters so that each cluster satisfies the dataset ratio. It is easy to see from the below corollary that $\tau_\ell = 1/k$ for all protected attribute values $\ell \in [m]$ implies dataset ratio.

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

This result follows trivially by replacing the attribute constraints in Theorem 4. However, the converse is not true. That is, a clustering satisfying Balance (equal to dataset ratio) can result in arbitrary bad τ -ratio fairness. Thus, τ -ratio fairness strictly generalizes Balance as follows.

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 ratio is 1. Consider the following allocation that satisfies the dataset ratio for each cluster. Cluster 1 is assigned two points, one belonging to each protected attribute value, and the remaining points are allocated to cluster 2. Note that for this allocation, $\tau_a = \tau_b = 1/n_a = 1/n_b = 2/n$. This value can be arbitrarily small for a large value of n . □

Along with Theorem 4, Lemma 6 shows that τ -ratio is a more general fairness notion than Balance. Apart from the above technical difference, these fairness notions differ conceptually in how they induce fair clustering. The Balance property requires a certain minimum representation ratio guarantee to hold in each cluster. It does not put

any additional constraint on the relative size of each cluster. This may lead to (potentially) skewed cluster sizes. Under τ -ratio the algorithm can appropriately control the minimum number of points 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 maximally balanced clusters i.e., $\tau_\ell = 1/k \forall \ell \in [m]$.

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

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 a 2-approximation guarantee to an optimal fair assignment up to 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. In particular, we show that $\text{FRAC}_{OE}(\mathcal{T}) \leq 2 \text{OPT}_{\text{assign}}(\mathcal{T}) + \beta$, where $\text{FRAC}_{OE}(\mathcal{T})$ and $\text{OPT}_{\text{assign}}(\mathcal{T})$ denote the objective value of the solution returned by FRAC_{OE} and optimal assignment algorithm, respectively, on given instance $\mathcal{T} = (C, X)$. And $\beta := 2 \sup_{x, y \in X} d(x, y)$ is 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.⁵ 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) $g_i \notin \mathcal{C}_1$ or 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 the 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 type. In summary, each 1-bad round can have either type 1 or type 2 bad assignment and each 2-bad round will have two bad assignments each of type 1 and type 2. Finally, let B be the set of all bad rounds and A be the set of all bad assignments.

Definition 9 (Complementary Assignment) An assignment $z \in A$ of type $(3 - t)$ is called the complementary assignment of $w \in A$ of type t if,

- 1) w and z are allocated in same round (i.e. in a 2-bad round) or
- 2) if w and z are allocated in i^{th} and j^{th} 1-bad rounds respectively with $i < j$, then z is the first bad assignment which has not been yet paired with a complementary assignment.

⁵ Note that an optimal fair allocation need not be unique. Our result holds for any optimal fair allocation.

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

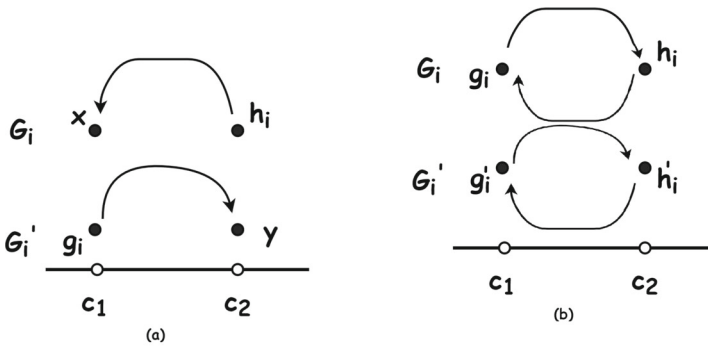
Proof Let $B = B_1 \cup B_2$, where B_t is a set of t -bad rounds. Note that the claim is trivially true if $B_1 = \emptyset$. Hence, let $|B_1| > 0$ and write $B_1 = B_{1,1} \cup B_{1,2}$. Here $B_{1,t}$ is a 1-bad round that resulted in type t bad assignment. Let $H_{1,t}$ be the set of good assignments of type t (i.e., correctly assigned to the center c_t) allocated in 1-bad rounds.

When n_ℓ is even, $|C_1| = |C_2|$ 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. Since in each 1-bad round, FRAC_{OE} results in exactly one bad assignment and exactly one good assignment, we have $|H_{1,t}| = |B_{1,(3-t)}|$. 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 without any complementary assignment. This completes the proof of the lemma. \square

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

Bounding 1-badrounds:

When n_ℓ is even, from Lemma 8, there is an even number of 1-bad rounds; two for each complimentary bad assignment. 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 $x \in C_1$ and $y \in C_2$ i.e. both are good assignments and $g_i \notin C_1, h_i \notin C_2$ are bad assignments. Now, consider an instance $\mathcal{T}_i = \{C, \{x, h_i, g_i, y\}\}$, then $\text{OPT}_{assign}(\mathcal{T}_i) = d(x, c_1) + d(h_i, c_1) + d(g_i, c_2) + d(y, c_2)$. We consider, without loss of generality, that the round G_i takes place before G'_i in the execution of FRAC_{OE} . For the other case, the proof is similar.



(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 assignments such that x, y are good assignments and allocated to the optimal center by algorithm, whereas g_i and h_i are bad assignments 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

First note that since FRAC_{OE} assigns h_i to cluster 2 while both g_i and y were available, we have

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

So,

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

If n_ℓ is odd, 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, $\text{FRAC}_{OE}(\mathcal{T}_i) \leq 2\text{OPT}_{\text{assign}}(\mathcal{T}_i) + \beta$. Here $\beta = 2 \sup_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the feature space.

Bounding 2-badrounds: First, assume that there is an 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, $\text{OPT}_{\text{assign}}(\mathcal{T}_i) = d(h_i, c_1) + d(h'_i, c_1) + d(g_i, c_2) + d(g'_i, c_2)$. As a consequence of the allocation rule used by FRAC_{OE} we have

$$\begin{aligned} d(g_i, c_1) \leq d(h_i, c_1), \quad d(g'_i, c_1) \leq d(h'_i, c_1), \quad d(h_i, c_2) \leq d(g'_i, c_2) \text{ and} \\ d(h_i, c_2) \leq d(h'_i, c_2). \end{aligned} \tag{7}$$

Furthermore,

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

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 $\text{FRAC}_{OE}(\mathcal{T}_i) - \text{OPT}_{assign}(\mathcal{T}_i) = d(g_i, c_1) + d(h_i, c_2) - d(g_i, c_2) - d(h_i, c_1) \leq d(g_i, c_1) + d(h_i, c_2) \leq \beta$. Thus,

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

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 $\text{FRAC}_{OE}(\mathcal{I}) \leq (2(\alpha + 2)\text{OPT}_{clust}(\mathcal{I}) + \beta)$ -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 $\text{FRAC}_{OE}(\hat{C}, X) \leq \text{FRAC}_{OE}(C, X)$. Here, C, \hat{C} are centers of vanilla clustering and fair clustering obtained by FRAC_{OE} respectively. 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 up to an instance-dependent additive constant.*

Proof In the previous proof, we considered two-length cycles. Two 1-bad assignments resulted in one type of cycle, and one 2-bad assignment resulted in another type of cycle. When the number of clusters is 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 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 Fig. 2 with an arrow pointing towards the optimal cluster. As the cycle is formed with respect to these points, we have $|X_1^q| = |X_2^1| = \dots = |X_q^{q-1}|$. The cost by FRAC_{OE} algorithm is then given as:

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

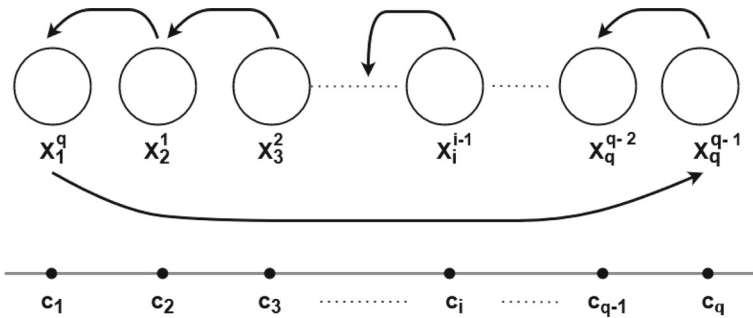


Fig. 2 Visual representation of set X_i^j and cycle of length q for Theorem 10. The arrow represents the direction from the assigned center to the center in optimal clustering. Thus, for each set X_i^j we have c_i as the currently assigned center and c_j as the center in optimal assignment

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

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

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. As the number of cases increases 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 a 2-approximate with respect to optimal τ -ratio fair assignment.

Proof We will find the approximation for $k = 3$ using a number of possible cases where one can have a cycle of length three. Let the centers involved in this 3-length cycle be denoted by c_i, c_j , and c_k . Note that if only one cycle involves these three centers, 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 depict multiple rounds when the two 3-length cycles can be formed. In the figure, if c_i is taking

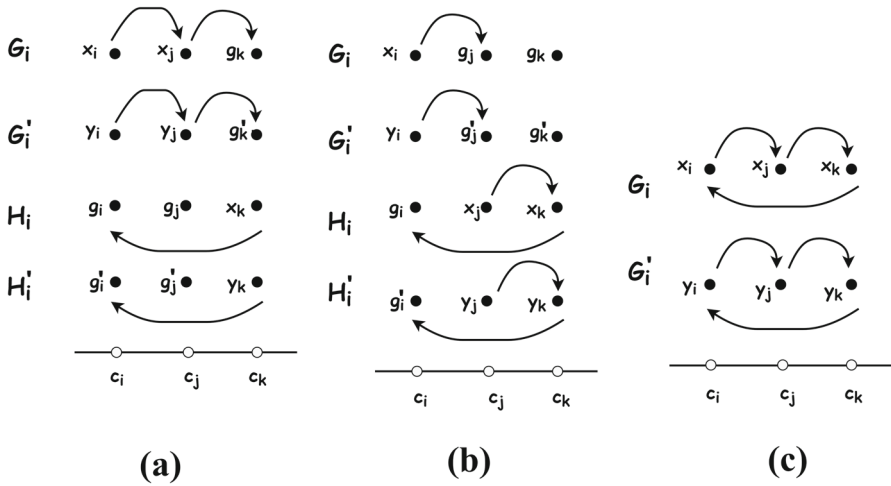


Fig. 3 Different use cases for 3-length cycle involving $k=3$ clusters **a** Case 1: Two-three length cycle pair (G_i, H_i) and (G'_i, H'_i) **b** Case 2: Second possibility of two-three length cycle pair (G_i, H_i) and (G'_i, H'_i) **c** Case 3: Three length cycle pair (G_i, G'_i)

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 cycles.

Case 1: In this case we bound the rounds shown in Fig. 3a. Let, one cycle completes in rounds G_i, H_i (i.e. using points from \mathcal{T}_i) and another cycle completes in rounds G'_i, H'_i (using points from \mathcal{T}'_i). Then,

$$\begin{aligned} OPT_{assign}(\mathcal{T}_i) &= d(x_i, c_j) + d(x_j, c_k) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_i) \\ OPT_{assign}(\mathcal{T}'_i) &= d(y_i, c_j) + d(y_j, c_k) + d(g'_k, c_k) + d(g'_i, c_i) + d(g'_j, c_j) + d(y_k, c_i) \end{aligned}$$

Further,

$$\begin{aligned} FRAC_{OE}(\mathcal{T}_i) &= d(x_i, c_i) + d(x_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k) \\ &\leq d(g'_i, c_i) + d(g'_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k) \end{aligned}$$

Now,

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

Combining the above two, we get:

$$\begin{aligned} FRAC_{OE}(\mathcal{T}_i) &\leq d(g'_i, c_i) + d(g'_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_i) \\ &\quad + d(y_k, c_i) + d(x_i, c_j) + d(y_i, c_j) + d(x_j, c_k) \end{aligned}$$

$$\begin{aligned} &\leq d(g'_i, c_i) + d(g'_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_i) + d(y_k, c_i) \\ &\quad + d(x_i, c_j) + d(y_i, c_j) + d(x_j, c_k) + d(y_j, c_k) + d(g'_k, c_k) \\ \text{FRAC}_{OE}(\mathcal{T}_i) &\leq \text{OPT}_{\text{assign}}(\mathcal{T}_i) + \text{OPT}_{\text{assign}}(\mathcal{T}'_i) \end{aligned}$$

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 a 2-approximation result plus a constant due to the last remaining cycle.

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

$$\begin{aligned} \text{OPT}_{\text{assign}}(\mathcal{T}_i) &= d(x_i, c_j) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(x_j, c_k) + d(x_k, c_i) \\ \text{OPT}_{\text{assign}}(\mathcal{T}'_i) &= d(y_i, c_j) + d(g'_j, c_j) + d(g'_k, c_k) + d(g'_i, c_i) + d(y_j, c_k) + d(y_k, c_i) \end{aligned}$$

Also, we know that

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

Combining the above two, we get:

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

Case 3: Here again we will have two allocation rounds namely G_i, G'_i as shown in Fig. 3c. It is easy to see that for this case,

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

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 a 2-approximation with respect to optimal assignment.*

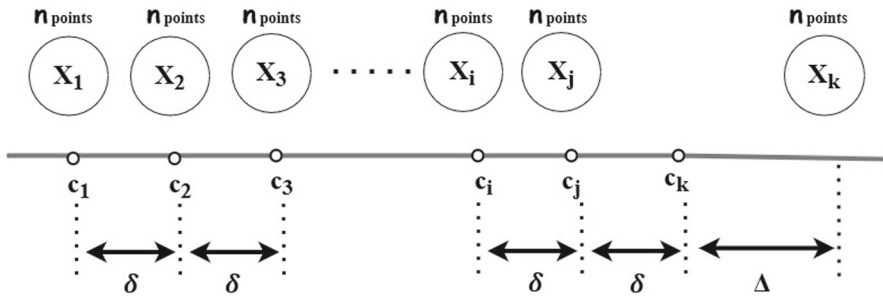


Fig. 4 The worst case example for fair clustering instance

Proof The worst case for any fair clustering instance is when the center, rather than choosing the points from its own optimal set, prefers points from sets of other centers. One example is depicted in Fig. 4. In this example, we consider k centers. For each of these centers, we have a set of n optimal points denoted by X_i for center c_i at a negligible distance (say zero) except the last center c_k . The set of optimal points for center c_k is located at a distance $\Delta = (k - 1)\delta$ where δ is the distance between all the centers. Now we will approximate the tightest bound on the cost. The optimal cost will sum up as

$$\begin{aligned} OPT_{assign} &= \sum_{x_i \in X_1} d(x_i, c_1) + \sum_{x_i \in X_2} d(x_i, c_2) + \dots + \sum_{x_i \in X_k} d(x_i, c_k) \\ &= 0 + 0 + 0 + n\Delta \end{aligned}$$

Suppose one uses round-robin-based $FRAC_{OE}$ to solve the assignment problem. Then at the start of the $t = 0^{th}$ round, each set X_i has n points. Since Δ is quite large compared to δ so c_k will prefer to choose points from the set of the previous center c_{k-1} . The remaining centers will take points from their respective set of optimal points as those points will have the least cost. Such assignments will continue until all the points in set X_{k-1} get exhausted. Thus the cost after $n/2$ rounds will be

$$\begin{aligned} Cost_1 &= \sum_{x_i \in X_1} d(x_i, c_1) + \dots + \sum_{x_i \in X_{k-1}} d(x_i, c_{k-1}) + \sum_{x_i \in X_{k-1}} d(x_i, c_k) \\ &= 0 + 0 + 0 + \frac{n\delta}{2} \end{aligned}$$

Now, as 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} . The other centers will 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 take place for the next $\frac{n}{6}$ rounds, and after that, the set X_{k-2} will get exhausted. The cost incurred to different centers in such an assignment will be

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

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, \dots, X_k , incurring the total cost of $((k-1)\delta + \Delta + (k-2)\delta + \Delta + \dots + \delta + \Delta + \Delta)\frac{n}{k}$. Thus, the total cost by $FRAC_{OE}$ is given as:

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

□

Research gap: Theorem 10 suggests that the approximation ratio to the number of clusters k can be exponentially bad. However, our experiments show—agreeing with our finding on small values of k (≤ 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. We conclude with the following conjecture.

Conjecture 13 $FRAC_{OE}$ is an 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 is left to show that 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). As a fair assignment procedure performs corrections for all available data points only once. Thus, $FRAC_{OE}$ is bound to converge leading us to the following lemma.

Lemma 14 $FRAC_{OE}$ algorithm converges.

5.2 Guarantees for FRAC_{OE} for general τ

Given an instance \mathcal{T} , centers C , and set of points X , we start with a simple observation that the problem of solving τ -ratio fair assignment can be divided into two subproblems:

1. Solving optimal $1/k$ -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 , i.e. $\mathcal{T}^{1/k} = \{X_1, C, \tau\}$ with $\tau_\ell = 1/k$ and $\mathcal{T}^0 = \{X_2, C, \tau\}$ with $\tau_\ell = 0$ for all $\ell \in [m]$.

Lemma 15 *For any given fair assignment instance \mathcal{T} on dataset X with centers C , there exists a partition of X into X_1 and X_2 and corresponding fair assignment instances $\mathcal{T}^{1/k}$ and \mathcal{T}^0 with $\tau_\ell = 1/k$ and $\tau_\ell = 0$ for all $\ell \in [m]$ respectively such that*

$$\text{OPT}_{\text{assign}}(\mathcal{T}) = \text{OPT}_{\text{assign}}(\mathcal{T}^{1/k}) + \text{OPT}_{\text{assign}}(\mathcal{T}^0).$$

Proof Let OPT be an optimal assignment of data points in X for fair clustering instance \mathcal{T} . To construct X_1 and X_2 , assign $\tau_\ell n_\ell$ points from each cluster j in OPT to X_1 and the remaining points to X_2 . If there is any cluster that is assigned exactly $\sum_{\ell=1}^m \tau_\ell n_\ell$, then all of its point will be moved to X_1 . Therefore, we will precisely have, $|X_1| = \sum_{j=1}^k \sum_{\ell=1}^m \tau_\ell n_\ell$ and $|X_2| = |X| - |X_1|$. The existence of X_1, X_2 follows from the feasibility condition of OPT. Further define the $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_1}$ and $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_2}$ to be the optimal cost by OPT on sets X_1 and X_2 respectively.

Consider the following fair assignment instances, $\mathcal{T}^{1/k} = \{X_1, C, \tau\}$ with $\tau_\ell = 1/k$ for all $\ell \in [m]$ and $\mathcal{T}^0 = \{X_2, C, \tau\}$ with $\tau_\ell = 0$ for all $\ell \in [m]$. Let us further denote the points in X_1 with a protected attribute value ℓ as $\{x_i^1, x_i^2, \dots, x_i^k\}_{i=1}^{\tau_\ell n_\ell}$. Thus, x_i^j denote the x_i point allocated to cluster center c_j by optimal assignment on instance \mathcal{T} . We will now prove the following: $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_1} = \text{OPT}_{\text{assign}}(\mathcal{T}^{1/k})$. If the optimal assignment on $\mathcal{T}^{1/k}$ results in the same assignment as \mathcal{T} , we have $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_1} = \text{OPT}_{\text{assign}}(\mathcal{T}^{1/k})$ as stated. Otherwise, there is a sequence of assignments (a cycle) denoted by $\{x_{i_1}^{\alpha_1}, x_{i_2}^{\alpha_2}, \dots, x_{i_{q-1}}^{\alpha_{q-1}}, x_{i_q}^{\alpha_1}\}$ such that the optimal assignment on $\mathcal{T}^{1/k}$ is in a way that the point $x_{i_1}^{\alpha_j}$ is assigned to center $c_{\alpha_{j+1}}$ for $j \in \{1, 2, \dots, q - 1\}$ and $x_{i_1}^{\alpha_q}$ to center c_{α_1} such that $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_1} > \text{OPT}_{\text{assign}}(\mathcal{T}^{1/k})$. Such a cycle however will contradict the optimality of OPT.⁶ Following the similar arguments, it can be shown that $\text{OPT}_{\text{assign}}(\mathcal{T})|_{X_2} = \text{OPT}_{\text{assign}}(\mathcal{T}^0)$. □

Let X_1^f be the set of points 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

⁶ This cycle would have resulted in further reduction of cost with respect to the points in the cycle.

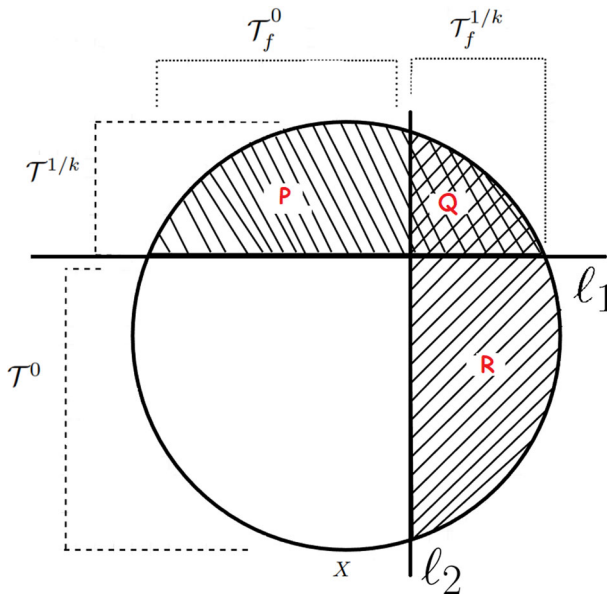


Fig. 5 Set of points X divided into instance $T^{1/k}$ and T^0 . Further the instances $T_f^{1/k}$ and T_f^0 are depicted in the same set of points X leading to formation of regions P, Q, R

consisting of points X_1^f and T_f^0 be instance when $\tau = \{0\}_{\ell=1}^m$ 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 $\text{OPT}_{assign}(T) = \text{OPT}_{assign}(T_f^{1/k}) + \text{OPT}_{assign}(T_f^0)$.

Proof Let optimal fair assignment on the set of points X create a partition along the axis given by line ℓ_1 in Fig. 5. This partition gives us two set of instances $T^{1/k}, T^0$ (as described earlier). Further, FRAC_{OE} achieves a partition along axis given by line ℓ_2 denoted by $T_f^{1/k}, T_f^0$. Now region Q contains the points in the overlap of $T^{1/k}$ and $T_f^{1/k}$. As we are talking about the optimal assignment problem, these points will be assigned to the same centers; hence, we can ignore these points for further analysis. Let the points allocated to any center c_j in $T_f^{1/k}$ by FRAC_{OE} in set R be $R_j = \{x_1, x_2, x_3, \dots, x_{m_j}\}$ and points allocated to c_j in partition P be $P_j = \{y_1, y_2, y_3, \dots, y_{m_j}\}$. Let g_j be a mapping function from $R_j \rightarrow P_j$ which maps any point $x_i \in R_j$ assigned to center j with $T_f^{1/k}$ to some point $y_i \in P_j$ assigned to same center when partition under consideration is $T^{1/k}$. Then, we have $\text{OPT}_{assign}(T_f^{1/k}) \leq \text{FRAC}_{OE}(T_f^{1/k}) = \sum_{j=1}^k \sum_{i=1}^{m_j} d(x_i, c_j) \leq \sum_{j=1}^k \sum_{i=1}^{m_j} d(g_j(x_i), c_j) = \sum_{j=1}^k \sum_{i=1}^{m_j} d(y_i, c_j) = \text{OPT}_{assign}(T^{1/k})$. This is because, for each $x_i \in R_j, \exists y_i \in P_j$ such that despite point $g(x_i)$ being available to center c_j, FRAC_{OE} chose the point x_i . As other points have no such constraint, we have, $\text{OPT}_{assign}(T_f^0) \leq \text{OPT}_{assign}(T^0)$. Together we get,

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

Theorem 17 *For $k=2, 3$ and for any given τ vector with $\tau_\ell \in [0, 1]$ for all ℓ and $\sum_\ell \tau_\ell \leq 1$, an allocation returned by FRAC_{OE} guarantees τ -ratio fairness and satisfies $(2(\alpha + 2)\mathcal{OPT}_{clust})$ approximation guarantee on clustering objective with respect to an fair clustering problem where α is the 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 Sect. 5.1, $\text{FRAC}_{OE}(\mathcal{T}_f^{1/k}) \leq 2\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k})$.

Also, as \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, we have $\text{FRAC}_{OE}(\mathcal{T}_f^0) = \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq 2\mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.

Equation 8 becomes,

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

\square

6 Fair round robin algorithm for clustering (FRAC): a heuristic approach

We now propose another in-processing 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. This may lead to the shuffling of points, affecting the position of next-step cluster centers. Also, modifying allocation does not preserve the convergence guarantee of the vanilla clustering algorithm. Thus, making the theoretical analysis of FRAC to be really hard. However, in experiments, we see that FRAC performs better than FRAC_{OE} on a wide range of real-world datasets.

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 among all the available algorithms in the literature. These empirical results suggest that either the worst-case instances for FRAC are unrealistic or a significantly different proof technique is needed to show the convergence guarantee. We leave this as an interesting future direction. As both FRAC_{OE} and FRAC solve the fair assignment problem through vanilla clustering problem, one can use them to find fair clustering for center-based approaches, i.e., k -means and k -median.

Algorithm 3: τ -FRAC

Input: set of datapoints X , number of clusters k , fairness requirement vector τ , range of protected attribute m , clustering objective norm p

Output: cluster centers C and assignment function ϕ

```

1 Choose the random centers as  $C$ 
2 while UntilConvergence do
3   for each  $x_i \in X$  do
4      $\phi(x_i) = \operatorname{argmin}_{j \in [k]} d(x_i, c_j)$ 
5   end
6    $(C, \phi) = \text{FAIRASSIGNMENT}(C, X, k, \tau, m, p, \phi)$ 
7 end

```

7 Experimental result and discussion

We validate the performance of the proposed algorithms across many benchmark datasets and compare them against the SOTA approaches. We observe in Sect. 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 fairness constraints after each round.

The benchmarking datasets used in the study are

- **Adult⁷ (Census)**- The data set contains information of 32,562 individuals from the 1994 census, of which 21,790 are males, and 10,771 are females. We choose five attributes as the feature set: age, fnlwgt, education_num, capital_gain, hours_per_week. The binary-valued protected attribute is sex, consistent with prior literature (Chierichetti et al. 2017; Bera et al. 2019; Backurs et al. 2019; Ziko et al. 2021). The dataset ratio is 0.49.
- **Bank⁸** - The dataset consists of marketing campaign data of a Portuguese bank. It has data of 41,108 individuals, of which 24,928 are married, 11,568 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 'marital status' is chosen as the protected attribute to be consistent with prior literature, resulting in a Balance of 0.18 (Chierichetti et al. 2017; Bera et al. 2019; Backurs et al. 2019; Ziko et al. 2021).
- **Diabetes⁹** - The dataset contains clinical records of 130 US hospitals over ten years. There are 54,708 and 47,055 hospital records of males and females, respectively. Consistent with the prior literature, only two features: age, time_in_hospital are used for the study (Chierichetti et al. 2017). Gender is treated as the binary-valued protected attribute yielding a Balance of 0.86.
- **Census II¹⁰** - It is the largest dataset used in this study containing 2,458,285 records from of US 1990 census, out of which 1,191,601 are males, and 1,266,684 are females. We chose 24 attributes commonly used in prior literature for this study

⁷ <https://archive.ics.uci.edu/ml/datasets/Adult>

⁸ <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

⁹ <https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008>

¹⁰ <https://archive.ics.uci.edu/ml/datasets/US+Census+Data+%281990%29>

(Bera et al. 2019; Ziko et al. 2021). Sex is the binary-valued protected attribute. The dataset ratio is 0.94.

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

- **Vanilla k -means:** An Euclidean distance-based k -means algorithm that does not incorporate fairness constraints
- **Vanilla k -median:** An Euclidean distance-based k -median algorithm that does not incorporate fairness constraints.
- **Bera et al. (2019):** The approach solves fair clustering through an LP formulation. Fairness is added as an additional constraint in the 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 within a reasonable amount of 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 objective cost 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 versions of the algorithm. We observe that the hyper-parameter λ is extremely sensitive to the datasets and the number of clusters. Further, tuning this hyper-parameter is computationally expensive. We were able to tune the value of λ in a reasonable amount of time only for Adult and Bank datasets for k -means clustering on varying numbers of clusters (called tuned version). Due to the added complexity of k -medians, we could 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 refer to this as Ziko et al. (2021) (untuned) version). In the untuned version, we use the same value across a varying number of cluster centers. The paper does not report any results for the 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 the Diabetes dataset.
- **Backurs et al. (2019):** This approach computes the fair clusters using fairlets in an efficient manner and is the extension of Chierichetti et al. (2017). This approach can 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 outperforms Chierichetti et al. (2017).

We use the following popular metrics in the literature to measure the different approaches' performance.

- **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

Table 1 Characteristics for real-world datasets commonly used in evaluation of fair clustering algorithms

Dataset name	#Cardinality	#Feature attributes	Protected attribute	Protected attribute cardinality	Protected attribute composition	Dataset ratio
Adult (Census)	32,562	5	Gender	Binary	21,790 males 10,771 females	– 0.49
Bank	41,108	6	Marital status	Ternary	24,928 married 11,568 unmarried	4612 divorced 0.18
Diabetes	101,763	2	Gender	Binary	54,708 males 47,055 females	– 0.86
Census-II	2,458,285	24	Gender	Binary	1,191,601 males 1,266,684 females	– 0.94

Number of feature attributes exclude protected attribute and for complete list of feature attributes see Sect. 7

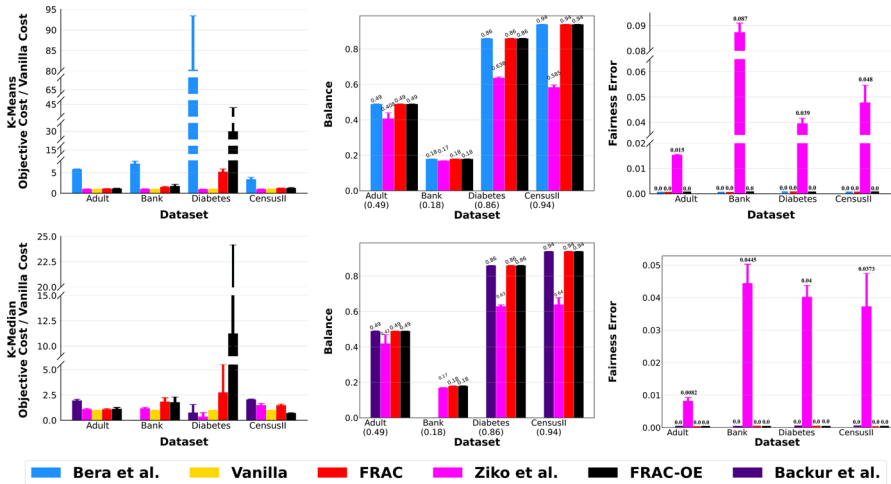


Fig. 6 The plot in the first row shows the variation in evaluation metrics for $k=10$ clusters. The objective cost is scaled against the vanilla objective cost. For Ziko et al. (2021), the λ values for k -means and k -median are taken to be the same as in their paper. The second row comprises plots for the k -median setting on the same k value. It should be noted that Backurs et al. (2019) do not work for the Bank dataset, which has a ternary valued protected group. The target Balance of each dataset is evident from the plot’s axes. (Best viewed in color) (Color figure online)

- **Fairness Error** (Ziko et al. 2021) It is the Kullback–Leibler (KL) divergence between the required protected group proportion τ and achieved proportion within the clusters:

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

The τ vector in fairness error captures the target proportion in each cluster for different protected groups $\ell \in [m]$. It is similar to the input vector τ for FRAC and $FRAC_{OE}$. In Bera et al. (2019), the target vector is denoted by δ (refer Sect. 7.3.3 for details on the parameter δ). We report the average and standard deviation of the performance measures across ten independent trials for every approach. The code for all the experiments is publicly available.¹¹ 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 rows in Fig. 6 summarize the results obtained for the k -means and k -median settings, respectively.

Observation for k -means:

- Ziko et al. (2021) achieve the lowest objective cost but with poor performance on both fairness measures.
- FRAC and $FRAC_{OE}$ achieve maximum Balance and zero fairness error with significantly lower objective costs compared to Bera et al. (2019).

¹¹ <https://github.com/shivi98g/Fair-k-means-Clustering-via-Algorithmic-Fairness>

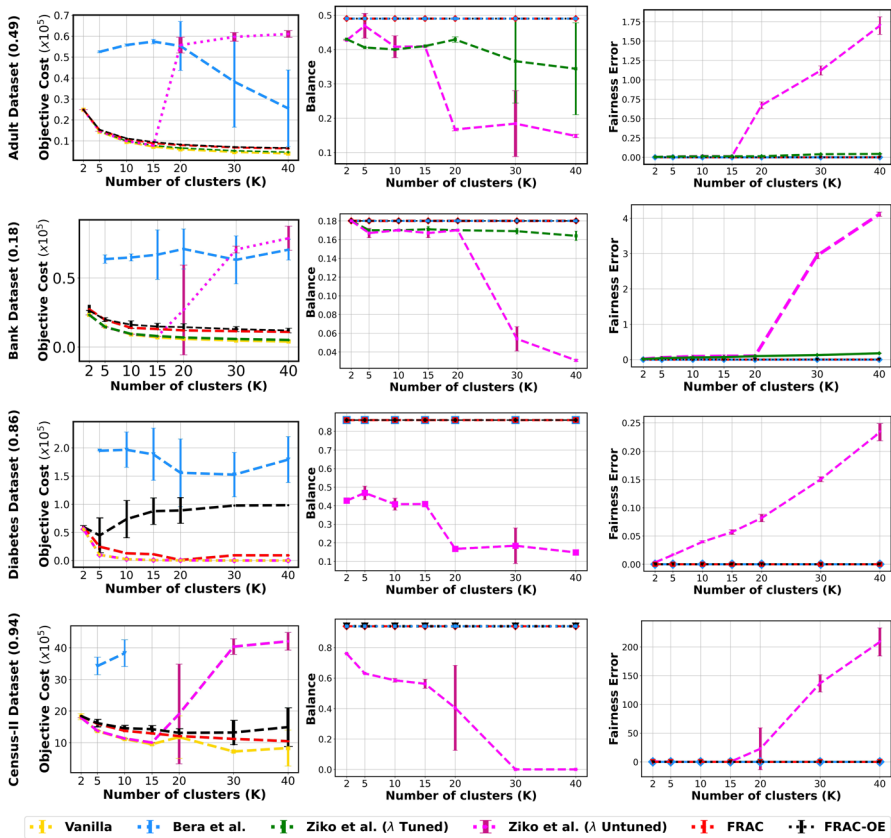


Fig. 7 The line plot shows variation of evaluation metrics over varying number of cluster center for k -means setting. The hyper-tuned variation of Ziko et al. is available only for Adult and Bank dataset due to expensive computational requirements. For other datasets the hyper-parameter λ is taken same as that is reported in Ziko et al. paper i.e. $\lambda=9000, 6000, 6000, 500,000$ 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) (Color figure online)

Observations for k -median setting:

- Backurs et al. (2019) result in fair clusters with high objective costs.
- Ziko et al. (2021) achieve better objective costs by trading off for fairness.
- FRAC and FRAC_{OE} obtain the least fairness error and a Balance that is equal to the required dataset ratio ($\tau_\ell = \frac{1}{k}$) while having a comparable objective cost.

7.1 Comparison across varying number of clusters (k)

This experiment evaluates the k -means version of different approaches with varying clusters from 2 to 40. Figure 7 summarises the results obtained for 2, 5, 10, 15, 20, 30, and 40 clusters on all datasets. For the most extensive dataset, Census-II, results are

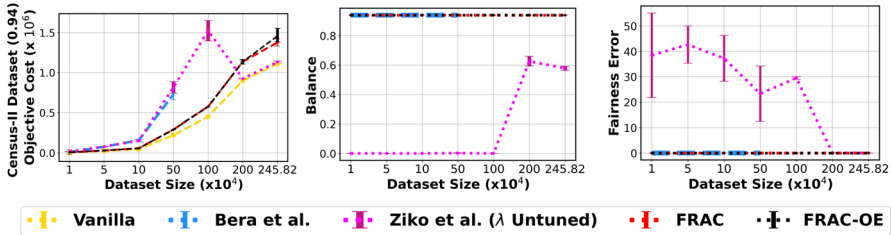


Fig. 8 The line plot shows variation of evaluation metrics over varying data set size for $k(=10)$ -means setting. The hyper-parameter $\lambda = 500,000$ is taken same as that is reported in Ziko et al. paper for Census-II dataset due to expensive computational requirements. On the similar reasons Bera et al. results for Census-II are evaluated up to $500k$. The target Balance for Census-II is evident from plot axes and complete dataset size is 245.82×10^4 . (Best viewed in color) (Color figure online)

obtained for only $k = 5$ and $k = 10$ due to the significant time complexity of solving the LP problem (Bera et al. 2019).

Observations:

- Bera et al. (2019) maintain fairness but with a much higher objective cost and fails to return any solution for $k = 2$.
- Ziko et al. (2021) (tuned) objective cost is close to vanilla k -means on the Adult and Bank datasets but at a significant deterioration in fairness metrics.
- Ziko et al. (2021) (untuned) has high objective cost and fairness error indicating the sensitivity to the hyper-parameter λ .
- FRAC gives the best result by maintaining a relatively low objective cost without compromising fairness.
- $FRAC_{OE}$ has a marginal cost difference from FRAC with the same fairness guarantees showing its efficacy.
- Theoretically, $FRAC_{OE}$ shows an approximation factor of 2^{k-1} , but the experimental performance does not degrade with an increase in k . This validates our conjecture.

7.2 Comparison across varying dataset sizes

In this experiment, we measure the performance of $k(=10)$ -means version of different approaches as the number of points in the data set increases. We use the largest data set – Census-II, for this experiment. Figure 8 shows the plots for evaluation metrics on varying data set sizes increasing from 10,000 to the total size of 2,458,285 data points. Due to the high computation requirements for Bera et al. (2019) (refer Sect. 7.4), we limit the results up to $500k$ data points. For Ziko et al. (2021), owing to high tuning time (refer run time analysis Sect. 7.4), we use the hyper-parameter value for Census-II as reported in Ziko et al. (2021) for complete data set, i.e., $\lambda=500,000$.

Observations:

- Bera et al. (2019) maintain strict fairness at higher objective costs.

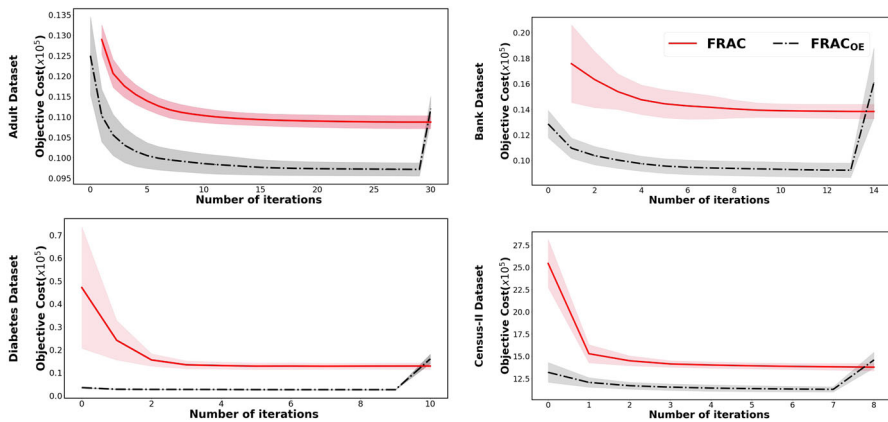


Fig. 9 The cost variation over the iterations for different approaches in $k(=10)$ -means

- Initially, the objective cost in Ziko et al. (2021) increases with dataset size but decreases on larger sizes (sensitive to hyper-parameter), but at significant degradation in fairness metrics.
- FRAC and FRAC_{OE} achieve strict fairness guarantees with a slight increase in objective cost from vanilla clustering.

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}

FRAC uses round-robin allocation after every clustering iteration. On the contrary, FRAC_{OE} applies the round-robin allocation only at the end of clustering. Both approaches will result in a fair allocation but might exhibit different objective costs. We experiment with the $k(=10)$ -means setting 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.

Observations: The plots in Fig. 9 indicate 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, we see a sudden jump overshooting the cost of FRAC (to accommodate fairness constraints). Thus, applying fairness constraints after every iteration is better than applying them 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.

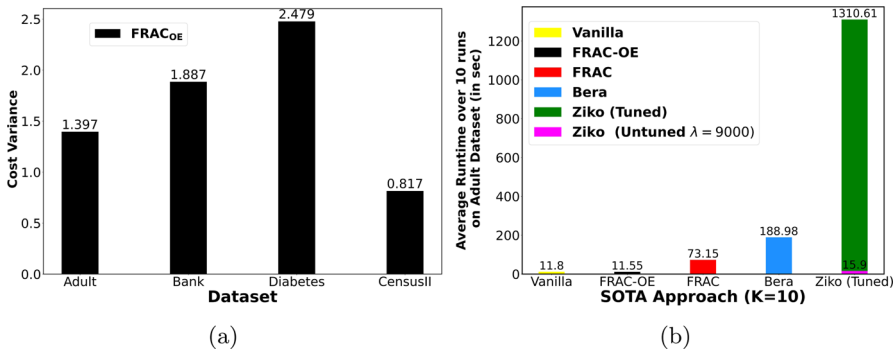


Fig. 10 **a** Bar plot shows the variance in objective cost over different 100 random permutations of converged centers returned by vanilla k -means clustering in $FRAC_{OE}$. **b** k -means runtime analysis of different SOTA approaches on Adult dataset for $k=10$ (Color figure online)

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 experiment with varying the order of the centers to see the impact on the objective cost for the $k(=10)$ -means clustering version. We report the objective cost variance computed across 100 permutations of the ten 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 objective costs (averaged over ten trials) is shown in Fig. 10a.

Observations: It is evident from the plot that the variance is consistently negligible 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 the 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 ratio. 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. It is the only SOTA approach that allows for the desired τ -ratio fairness in a restrictive manner. Bera et al. reduce the degree of freedom using a δ parameter that controls the lower and upper bound on the 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 the 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 a baseline only for certain τ_ℓ values. In Table 2, we present results for the τ corresponding to $\delta=0.2, 0.8$.

Table 2 k -means objective cost for τ -ratio for Adult and Bank dataset for $k=10$ clusters

Dataset	τ - vector	FRAC Objective cost		FRAC _{OE} Objective cost		Bera et al. δ Value	Objective cost
		FRAC Objective cost	FRAC Objective cost	FRAC _{OE} Objective cost	FRAC _{OE} Objective cost		
Adult	$< 0.133, 0.066 >$	9804.65 \pm 221.05	9616.51 \pm 111.49	0.8	9515.30 \pm 19.94		
	$< 0.535, 0.264 >$	10,010.39 \pm 211.27	10,011.78 \pm 239.73	0.2	9788.73 \pm 23.32		
	$< 0.25, 0.12 >$	9870.93 \pm 261.24	9714.06 \pm 157.45	<i>Cannot be computed</i>			
Bank	$< 0.121, 0.056, 0.022 >$	9210.38 \pm 640.76	9043.51 \pm 461.23	0.2	9588.30 \pm 48.82		
	$< 0.485, 0.225, 0.089 >$	10,982.63 \pm 1228.28	11,317.61 \pm 1310.32	0.8	8472.65 \pm 37.30		
	$< 0.25, 0.10, 0.04 >$	9548.68 \pm 540.86	9465.35 \pm 476.88	<i>Cannot be computed</i>			

Observation: Our algorithms can achieve any generalized τ vectors like $[0.25, 0.12]$. Such vectors make more sense in real-world applications, like requiring at least 25% male and 12% female points in each cluster. The objective cost obtained by FRAC and FRAC_{OE} is comparable to Bera et al. (2019), but the work by Bera et al. (2019) is extendible to the multiple protected attributes.

7.4 Run-time analysis

Finally, we compare the runtime of the different approaches for the $k(=10)$ -means clustering versions on the Adult dataset. The average runtime over ten different runs is reported in Fig. 10b.

Observations:

- Runtime of FRAC is significantly better than the fair SOTA approaches.
- Ziko et al. (2021) (tuned) runtime is quite high due to hyper-parameter tuning.
- Ziko et al. (2021) (untuned) is comparable to vanilla clustering but with a deterioration in fairness (seen in previous sections).
- FRAC_{OE} has a marginal difference from vanilla runtime as it applies a single round of fair assignment.
- Bera et al. (2019) being LP formulation has higher complexity and requires double the time of FRAC.

Motivated by Kriegel et al. (2017), we further study the runtime behavior across a varying number of data points and a varying number of clusters. For the scalability study, we perform the analysis using Census-II as it is the largest dataset. We use the same hyper-parameter value ($\lambda=500,000$) for Ziko et al. (2021) in this study.

7.4.1 Runtime comparison with number of cluster(k)

In this study, we experiment to find the variation in runtime as the 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) has a significantly high execution time. Thus, we limit the results to $k(=5, 10)$ -clustering. As pointed out in the previous section Bera et al. (2019), LP fails to converge for $k=2$.

Observations:

- FRAC_{OE} has a runtime close to vanilla clustering.
- Ziko et al. (2021), even in the untuned version, the runtime is close to FRAC. Tuning will result in a significant increase in overall runtime.
- Bera et al. (2019) have a significantly higher runtime.

7.4.2 Runtime comparison across varying data set size

We study the scalability of different approaches with the increased data set size for $k=10$. For Bera et al. (2019), plots in Fig. 12 reveal that the run time significantly increases even with $500k$ points in the dataset. So we limit the study to this size.

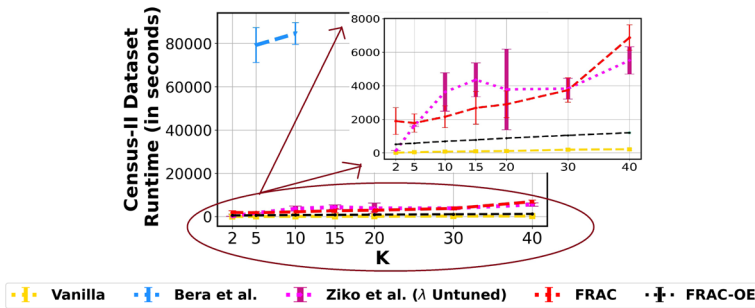


Fig. 11 The line plot shows runtime variation over varying number of clusters(k) for k -means setting on complete dataset. The hyper-parameter $\lambda = 500,000$ is taken the same as that reported in Ziko et al. paper for the Census-II dataset due to expensive computational requirements. For similar 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) (Color figure online)

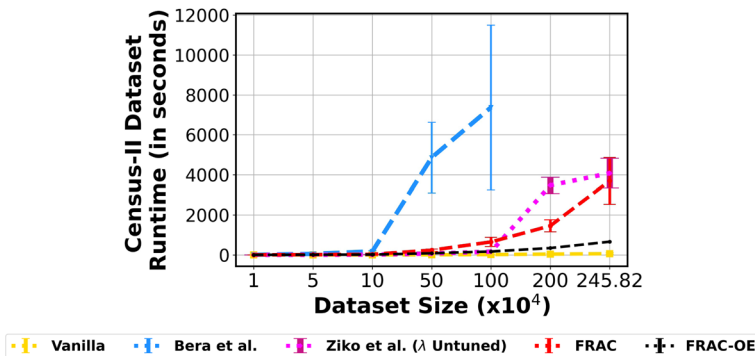


Fig. 12 The line plot shows runtime variation over varying dataset sizes (upto the total dataset size of 245.82×10^4) for $k=10$ -means setting. The hyper-parameter $\lambda = 500,000$ is taken the same as that reported in Ziko et al. paper for the Census-II dataset due to expensive computational requirements. For similar reasons, Bera et al. results for Census-II are evaluated for dataset sizes of $10k$, $50k$ and $100k$. (Best viewed in color)

Observations:

- Ziko et al. (2021) (untuned) runtime is close to vanilla clustering. However, the gap increases after a certain dataset size.
- $FRAC_{OE}$ follows a trend slightly close to vanilla clustering and does not deteriorate with the size showing its efficiency.
- FRAC has a run time larger than vanilla clustering but is comparable to untuned Ziko et al. (2021).
- Tuning 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 algorithm, FRAC_{OE} , (i) achieves a $2(\alpha + 2)$ -approximate solution up to three clusters, and (ii) achieves $2^{k-1}(\alpha + 2)$ -approximate guarantees to general k with $\tau = \{1/k\}_{\ell=1}^m$. Current proof techniques for $k \leq 3$ requires intricate case analysis, which becomes intractable for larger k . However, our experiments show that FRAC_{OE} and FRAC outperforms SOTA approaches in terms of objective cost and fairness measures even for $k > 3$. We also prove the cost approximation for the general τ vector and show convergence analysis for FRAC_{OE} . An immediate future direction is to analytically prove a $2(\alpha + 2)$ -approximation guarantee for general k .

It is worth noting that the τ -ratio fairness ensures the Balance property. However, if one uses 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 the immediate future directions mentioned above, 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 while maintaining group fairness are interesting research problems.

Funding The research is funded by Department of Science & Technology, India under grant number SRG/2020/001138 (Recipient name- Dr. Shweta Jain). The authors would like to thank the Prime Minister Research Fellowship, the Ministry of Education, Government of India for generously funding the primary author.

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

Declarations

Conflict of interest No potential competing interest was reported by the authors.

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, Bhaskara A, Venkatasubramanian S (2021) Fair clustering via equitable group representations. In: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, pp 504–514

- Abraham SS, Padmanabhan D, Sundaram SS (2020) Fairness in clustering with multiple sensitive attributes. In: EDBT/ICDT 2020 joint conference, pp 287–298
- Ahmadian S, Epasto A, Kumar R, Mahdian M (2019) Clustering without over-representation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 267–275
- Ahmadian S, Epasto A, Kumar R, Mahdian M (2020) Fair correlation clustering. In: Chiappa S, Calandra R (eds) Proceedings of the twenty third international conference on artificial intelligence and statistics, volume 108 of proceedings of machine learning research, pp 4195–4205
- Anderson N, Bera SK, Das S, Liu Y (2020) Distributional individual fairness in clustering. [arXiv:2006.12589](https://arxiv.org/abs/2006.12589)
- Anegg G, Angelidakis H, Kurpisz A, Zenklusen R (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, Indyk P, Onak K, Schieber B, Vakilian A, Wagner T (2019) Scalable fair clustering. In: International conference on machine learning, pp 405–413
- Bandyapadhyay S, Fomin FV, Simonov K (2020) On coresets for fair clustering in metric and euclidean spaces and their applications. [arXiv:2007.10137](https://arxiv.org/abs/2007.10137)
- Bandyapadhyay S, Inamdar T, Pai S, Varadarajan K (2019) A constant approximation for colorful k-center. [arXiv:1907.08906](https://arxiv.org/abs/1907.08906)
- Banerjee A, Ghosh J (2006) Scalable clustering algorithms with balancing constraints. *Data Min Knowl Discov* 13(3):365–395
- Barocas S, Selbst AD (2016) Big data's disparate impact. *California Law Rev*, 671–732
- Baumann E, Rumberger JL (2018) State of the art in fair ML: from moral philosophy and legislation to fair classifiers. *CoRR* [arXiv:1811.09539](https://arxiv.org/abs/1811.09539)
- Bera S, Chakrabarty D, Flores N, Negahbani M (2019) Fair algorithms for clustering. *Adv Neural Inf Process Syst* 32:4954–4965
- Bercea IO, Groß M, Khuller S, Kumar A, Rösner C, Schmidt DR, Schmidt M (2018) On the cost of essentially fair clusterings. [arXiv:1811.10319](https://arxiv.org/abs/1811.10319)
- Böhm M, Fazzone A, Leonardi S, Schwiegelshohn C (2020) Fair clustering with multiple colors. [arXiv:2002.07892](https://arxiv.org/abs/2002.07892)
- Bose A, Hamilton W (2019) Compositional fairness constraints for graph embeddings. In: International conference on machine learning, pp 715–724
- Bottou L, Bengio Y (1994) Convergence properties of the k-means algorithms. *Advances in neural information processing systems* 7
- Brubach B, Chakrabarti D, Dickerson J, Khuller S, Srinivasan A, Tsepenekas L (2020) A pairwise fair and community-preserving approach to k-center clustering. In: International conference on machine learning, pp 1178–1189
- Brubach B, Chakrabarti D, Dickerson JP, Srinivasan A, Tsepenekas L (2021) Fairness, semi-supervised learning, and more: A general framework for clustering with stochastic pairwise constraints. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 6822–6830
- Byrka J, Penschyl T, Rybicki B, Srinivasan A, Trinh K (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 AN, Wu X (2022) The fairness field guide: perspectives from social and formal sciences. [arXiv:2201.05216](https://arxiv.org/abs/2201.05216)
- Chakrabarti D, Dickerson JP, Esmaeili SA, Srinivasan A, Tsepenekas L (2022) A new notion of individually fair clustering: α -equitable k-center. In: International conference on artificial intelligence and statistics, pp 6387–6408
- Chan TH, Guerqin A, Sozio M (2018) Fully dynamic k-center clustering. In: Proceedings of the 2018 World Wide Web conference, pp 579–587
- Chen X, Fain B, Lyu L, Munagala K (2019) Proportionally fair clustering. In: International conference on machine learning, pp 1032–1041
- Chhabra A, Masalkovaitė K, Mohapatra P (2021) An overview of fairness in clustering. *IEEE Access* 9:130698–130720. <https://doi.org/10.1109/ACCESS.2021.3114099>
- Chhabra A, Singla A, Mohapatra P (2021) Fair clustering using antidote data. [arXiv:2106.00600](https://arxiv.org/abs/2106.00600)
- Chierichetti F, Kumar R, Lattanzi S, Vassilvitskii S (2017) Fair clustering through fairlets. In: Proceedings of the 31st international conference on neural information processing systems, pp 5036–5044

- Chikahara Y, Sakaue S, Fujino A, Kashima H (2021) Learning individually fair classifier with path-specific causal-effect constraint. In: International conference on artificial intelligence and statistics, pp 145–153
- Chlamtáč E, Makarychev Y, Vakilian A (2022) Approximating fair clustering with cascaded norm objectives. In: Proceedings of the 2022 annual ACM-SIAM symposium on discrete algorithms (SODA), pp 2664–2683. SIAM
- Cho J, Hwang G, Suh C (2020) A fair classifier using mutual information. In: 2020 IEEE international symposium on information theory (ISIT), pp 2521–2526
- Correa J, Cristi A, Duetting P, Norouzi-Fard A (2021) Fairness and bias in online selection. In: International conference on machine learning, pp 2112–2121
- 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, Ravi S (2020) Making existing clusterings fairer: algorithms, complexity results and insights. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 3733–3740
- Deepak, Abraham SS (2020). Representativity fairness in clustering. In: 12th ACM conference on web science
- Deepak JM, Jose SV (2020) Fairness in unsupervised learning. In: Proceedings of the 29th ACM international conference on information & knowledge management, CIKM '20, New York, NY, USA, pp 3511–3512. Association for Computing Machinery
- Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2012) Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference, pp 214–226
- Elzayn H, Jabbari S, Jung C, Kearns M, Neel S, Roth A, Schutzman Z (2019) Fair algorithms for learning in allocation problems. In: Proceedings of the conference on fairness, accountability, and transparency, pp 170–179
- Esmaili S, Brubach B, Srinivasan A, Dickerson J (2021) Fair clustering under a bounded cost. *Adv Neural Inf Process Syst* 34:14345–14357
- Esmaili S, Brubach B, Tsepenekas L, Dickerson J (2020) Probabilistic fair clustering. *Adv Neural Inf Process Syst* 33:12743–12755
- Feng Z, Kacham P, Woodruff D (2021) Dimensionality reduction for the sum-of-distances metric. In: International conference on machine learning, pp 3220–3229
- Ghadiri M, Samadi S, Vempala S (2021) Socially fair k-means clustering. In: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, pp 438–448
- Ghadiri M, Singh M, Vempala SS (2022) Constant-factor approximation algorithms for socially fair k -clustering. *arXiv preprint arXiv:2206.11210*
- Gong S, Liu X, Jain AK (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, Jaiswal R (2021) Tight fpt approximation for socially fair clustering. *arXiv:2106.06755*
- Han L, Xu D, Xu Y, Yang P (2022) Approximation algorithms for the individually fair k -center with outliers. *J Global Optim*, 1–16
- Harb E, Lam HS (2020) KFC: a scalable approximation algorithm for k -center fair clustering. *Adv Neural Inf Process Syst* 33:14509–14519
- Harris DG, Pensyl T, Srinivasan A, Trinh K (2019) A lottery model for center-type problems with outliers. *ACM Trans Algorithms* 15(3):1–25
- Huang L, Jiang S, Vishnoi N (2019) Coresets for clustering with fairness constraints. *Adv Neural Inf Process Syst* 32:7589–7600
- Jia X, Sheth K, Svensson O (2020) Fair colorful k -center clustering. In: International conference on integer programming and combinatorial optimization, pp 209–222. Springer
- Jones M, Nguyen H, Nguyen T (2020) Fair k -centers via maximum matching. In: International conference on machine learning, pp 4940–4949
- Julia A, Larson J, Mattu S, Kirchner L (2016) Propublica–machine bias. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>. [Online; accessed 13-August-2021]
- Jung C, Kannan S, Lutz N (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, Medya S, Mandal D, Silva A, Dey P, Sanyal S (2021) Feature-based individual fairness in k -clustering. *arXiv:2109.04554*

- Kleindessner M, Awasthi P, Morgenstern J (2019) Fair k -center clustering for data summarization. In: International conference on machine learning, pp 3448–3457
- Kleindessner M, Awasthi P, Morgenstern J (2020) A notion of individual fairness for clustering. [arXiv:2006.04960](https://arxiv.org/abs/2006.04960)
- Kleindessner M, Samadi S, Awasthi P, Morgenstern J (2019) Guarantees for spectral clustering with fairness constraints. In: International conference on machine learning, pp 3458–3467
- 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 HP, Schubert E, Zimek A (2017) 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, Ntoutsis E (2021) Towards fair, explainable and actionable clustering for learning analytics. In: EDM
- Le Quy T, Roy A, Iosifidis V, Zhang W, Ntoutsis E (2022) A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1452
- Lee JK, Bu Y, Rajan D, Sattigeri P, Panda R, Das S, Wornell GW (2021) Fair selective classification via sufficiency. In: International conference on machine learning, pp 6076–6086
- Li B, Li L, Sun A, Wang C, Wang Y (2021) Approximate group fairness for clustering. In: International conference on machine learning, pp 6381–6391
- Li P, Zhao H, Liu H (2020) Deep fair clustering for visual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
- Liu S, Vicente LN (2021) A stochastic alternating balance k -means algorithm for fair clustering. [arXiv:2105.14172](https://arxiv.org/abs/2105.14172)
- Lohaus M, Perrot M, Luxburg UV (2020) Too relaxed to be fair. In: III HD, Singh A (eds) Proceedings of the 37th international conference on machine learning, vol. 119 of proceedings of machine learning research, pp 6360–6369
- Mahabadi S, Vakilian A (2020) Individual fairness for k -clustering. In: International conference on machine learning, pp 6586–6596
- Makarychev Y, Vakilian A (2021) Approximation algorithms for socially fair clustering. [arXiv:2103.02512](https://arxiv.org/abs/2103.02512)
- Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A (2021) A survey on bias and fairness in machine learning. *ACM Comput Surv* 54(6). <https://doi.org/10.1145/3457607>
- Micha E, Shah N (2020) Proportionally fair clustering revisited. In: 47th international colloquium on automata, languages, and programming (ICALP 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik
- Negahbani M, Chakrabarty D (2021) Better algorithms for individually fair k -clustering. *Adv Neural Inf Process Syst* 34:13340–13351
- Ntoutsis E, Fafalios P, Gadiraju U, Iosifidis V, Nejdil W, Vidal ME, Ruggieri S, Turini F, Papadopoulos S, Krasanakis E et al (2020) Bias in data-driven artificial intelligence systems-an introductory survey. *Wiley Interdiscip Rev Data Min Knowl Discov* 10(3):e1356
- Padmanabhan D (2020) Whither fair clustering? In: AI for social good: Harvard CRCS Workshop
- Quy TL, Roy A, Friege G, Ntoutsis E (2021) Fair-capacitated clustering. [arXiv:2104.12116](https://arxiv.org/abs/2104.12116)
- Ranzato F, Urban C, Zanella M (2021) Fair training of decision tree classifiers. [arXiv:2101.00909](https://arxiv.org/abs/2101.00909)
- Rösner C, Schmidt M (2018) Privacy preserving clustering with constraints. [arXiv:1802.02497](https://arxiv.org/abs/1802.02497)
- Schmidt M, Schwiigelshohn C, Sohler C (2019) Fair coresets and streaming algorithms for fair k -means. In: International Workshop on Approximation and Online Algorithms, pp 232–251. Springer
- Schmidt M, Wargalla J (2021) Coresets for constrained k -median and k -means clustering in low dimensional euclidean space. [arXiv:2106.07319](https://arxiv.org/abs/2106.07319)
- Song H, Li P, Liu H (2021) Deep clustering based fair outlier detection. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, New York, NY, USA, pp 1481–1489. Association for Computing Machinery
- Thejaswi S, Ordozgoiti B, Gionis A (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, Yalciner M (2022) Improved approximation algorithms for individually fair clustering. In: International Conference on Artificial Intelligence and Statistics, pp 8758–8779. PMLR
- Wang B, Davidson I (2019) Towards fair deep clustering with multi-state protected variables. [arXiv preprint arXiv:1901.10053](https://arxiv.org/abs/1901.10053)
- Zhang H, Davidson I (2021) Deep fair discriminative clustering. [arXiv preprint arXiv:2105.14146](https://arxiv.org/abs/2105.14146)

- Zhang W, Bifet A, Zhang X, Weiss JC, Nejdil W (2021) Farf: A fair and adaptive random forests classifier, *Advances in Knowledge Discovery and Data Mining*, 245–256. Springer International Publishing
- Ziko IM, Yuan J, Granger E, Ayed IB (2021) Variational fair clustering. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 35, pp 11202–11209

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.