

Fairness-Aware Community Detection and Clustering

A Thesis

submitted to the designated
by the Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Glykeria Toulina

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER
SYSTEMS ENGINEERING

WITH SPECIALIZATION
IN DATA SCIENCE AND ENGINEERING

University of Ioannina

School of Engineering

Ioannina 2026

Examining Committee:

- **Panayiotis Tsaparas**, Assoc. Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Evaggelia Pitoura**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Aristidis Likas**, Professor, Department of Computer Science and Engineering, University of Ioannina

TABLE OF CONTENTS

List of Figures	iv
List of Tables	v
List of Algorithms	vi
Abstract	vii
Εκτεταμένη Περίληψη	ix
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Overview and Contributions	2
1.3 Thesis Roadmap	4
2 Related Work	5
2.1 Fairness Metrics in Clustering and Community Detection	5
2.2 Methodologies for Mitigating Bias and Achieving Fairness	7
2.2.1 Fairness-Aware Clustering Algorithms	7
2.2.2 Fairness-Aware Community Detection Algorithms	11
3 Preliminaries	13
3.1 Community Detection	13
3.2 Data Clustering	14
3.3 Fairness Metric: Balance	14
3.4 A Physics-Inspired Framework for Fairness	15
4 Fair Label Propagation	19
4.1 The Standard Label Propagation Algorithm	19

4.2	The Fair Label Propagation Algorithm (FLP)	20
4.3	Experiments	22
4.3.1	Experimental Setup	22
4.3.2	Experimental results	24
4.4	Discussion	28
5	Force-Field Fair K-Means	29
5.1	Standard K-means	29
5.2	The Force-Field Fair K-Means (F3KM)	30
5.2.1	Centroid initialization strategies	32
5.2.2	Centroid update strategies	33
5.3	Experiments	34
5.3.1	Evaluation Metrics	34
5.3.2	Datasets	35
5.3.3	Results	38
5.4	Discussion	47
6	Fair Force-Field Clustering	51
6.1	Force-Field Clustering (F2C)	52
6.1.1	Gravitational interaction	52
6.1.2	Particle movement	53
6.1.3	Merging and convergence	53
6.2	Fair Force-Field Clustering (F3C)	54
6.2.1	Fairness interaction	54
6.2.2	Combined force	55
6.2.3	Algorithm	55
6.3	Experiments	56
6.3.1	Evaluation Metrics	56
6.3.2	Datasets	58
6.3.3	Experimental Setup	59
6.3.4	F2C Results	60
6.3.5	Effect of the Fairness Term	61
6.4	Discussion	64
7	Conclusions	66

LIST OF FIGURES

4.1	Fairness-Quality trade-off	25
4.2	Comparison on synthetic datasets of different hardness.	26
5.1	Synthetic Datasets: (a) basic, (b) circular, (c) circular_8, (d) elliptical . .	37
5.2	Results for synthetic datasets-different initialization	40
5.3	Results for synthetic datasets-different updates of centroids	41
5.4	Results for real datasets-different notions of balance	42
6.1	Representative synthetic datasets used in the experimental evaluation of F2C and F3C.	58
6.2	Effect of λ on Balance, Silhouette, and SSE for the synthetic datasets. .	62

LIST OF TABLES

4.1	Real Dataset Characteristics	23
4.2	Algorithm results on real datasets	27
5.1	Summary of Preprocessed and Sampled Real-World Datasets	38
5.2	Comparison of fair clustering methods on real datasets	45
5.3	Comparison of F3KM, BFKM, and FCA on synthetic datasets. For F3KM, we report the smallest value of λ among the tested configurations that gives the selected best result.	47
6.1	External clustering agreement for F2C and MeanShift on synthetic datasets.	60
6.2	Base and best fairness-aware F3C results on real datasets.	63

LIST OF ALGORITHMS

4.1	Fair Label Propagation (FLP)	22
5.1	Force-Field Fair K-Means (F3KM)	32
6.1	Fair Force-Field Clustering F3C	56

ABSTRACT

Glykeria Toulina, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2026.

Fairness-Aware Community Detection and Clustering.

Advisor: Panayiotis Tsaparas, Associate Professor.

Unsupervised learning methods, such as clustering and community detection, aim to partition data into meaningful groups. In clustering, objects are grouped according to their similarity in a feature space, so that points within the same cluster are more similar to each other than to points in different clusters. In community detection, nodes in a graph are grouped according to their connectivity patterns, so that nodes within the same community are more densely connected than nodes across different communities.

These methods are widely used in data mining and machine learning for tasks such as data summarization, dimensionality reduction, exploratory analysis, and pre-processing for downstream models. Because their outputs may influence later decisions, it is important to ensure that the produced clusters or communities are not only structurally meaningful, but also fair, balanced, and transparent with respect to sensitive attributes.

However, traditional algorithms are usually designed to optimize geometric or structural quality criteria without considering sensitive attributes. As a result, the produced clusters or communities may reinforce the demographic imbalance, isolate protected groups, or reflect existing biases in the data.

This thesis studies fairness-aware partitioning in both community detection and clustering.

We propose a unified physics-inspired framework in which clusters, communities are represented through fairness charges. The charge of a cluster or community

depends on its demographic composition

Under this view, imbalanced clusters or communities attract elements that improve their demographic balance and repel assignments that would increase imbalance. At the same time, the original assignment criterion is formulated as an "attraction force", capturing local connectivity in community detection and distance-based spatial proximity in clustering.

Based on this framework, we develop three methods. First, we introduce Fair Label Propagation, a fairness-aware variant of the Label Propagation algorithm for community detection. The method combines local connectivity force with an electrostatic fairness force in order to produce balanced communities. Then, we propose Force-Field Fair K-Means, which modifies the assignment step of K-means through a fairness-aware force score based on both distance to centroids and demographic imbalance. Last, we present Force-Field Clustering, starting from a standard clustering approach and extending it into a fairness-aware method, Fair Force-Field Clustering, by incorporating demographic forces into the same physics-inspired framework.

We evaluate the proposed methods on synthetic and real datasets using both fairness and quality metrics. The experimental results show that the proposed algorithms can improve demographic balance, while illustrating the trade-off between fairness and structural quality. Overall, the results demonstrate that fairness-aware force-based assignments can provide a flexible mechanism for incorporating demographic constraints into unsupervised partitioning methods.

ΕΚΤΕΤΑΜΕΝΗ ΠΕΡΙΛΗΨΗ

Γλυκερία Τουλίνα, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2026.

Ανίχνευση και ομαδοποίηση κοινοτήτων με γνώμονα τη δικαιοσύνη.

Επιβλέπων: Παναγιώτης Τσαπάρας, Αναπληρωτής Καθηγητής.

Ανίχνευση κοινοτήτων και ομαδοποίηση με γνώμονα τη δικαιοσύνη.

Η παρούσα διπλωματική εργασία μελετά το πρόβλημα της δικαιοσύνης σε μεθόδους μη επιβλεπόμενης μάθησης, με έμφαση στην ανίχνευση κοινοτήτων σε γράφους και στην ομαδοποίηση δεδομένων. Οι κλασικοί αλγόριθμοι ομαδοποίησης και ανίχνευσης κοινοτήτων συνήθως βελτιστοποιούν κριτήρια ποιότητας, όπως η γεωμετρική εγγύτητα, η συνοχή των ομάδων ή η πυκνότητα των συνδέσεων. Ωστόσο, δεν λαμβάνουν υπόψη ευαίσθητα χαρακτηριστικά, όπως το φύλο, η πολιτική προτίμηση ή άλλα δημογραφικά γνωρίσματα. Αυτό μπορεί να οδηγήσει σε ομάδες ή κοινότητες που είναι δομικά καλές, αλλά δημογραφικά άνισες.

Για την αντιμετώπιση αυτού του προβλήματος, η εργασία προτείνει ένα ενιαίο πλαίσιο εμπνευσμένο από φυσικές δυνάμεις, το οποίο εφαρμόζεται τόσο στην ανίχνευση κοινοτήτων όσο και στην ομαδοποίηση δεδομένων. Συγκεκριμένα, ομάδες στοιχείων, όπως clusters ή communities, λαμβάνουν φορτία δικαιοσύνης σύμφωνα με τη δημογραφική τους σύνθεση, ενώ η αλληλεπίδρασή του με άλλα στοιχεία, ομάδες ή κοινότητες δημιουργεί δυνάμεις δικαιοσύνης.

Εμπνευσμένοι από τον νόμο του Coulomb, θεωρούμε ότι στοιχεία ή ομάδες με αντίθετα φορτία έλκονται, ενώ στοιχεία ή ομάδες με ίδια φορτία απωθούνται. Στο πλαίσιο της εργασίας, η αρχή αυτή χρησιμοποιείται ώστε δημογραφικά ανισόρροπες ομάδες ή κοινότητες να ευνοούν αναθέσεις στοιχείων από την υποεκπροσωπούμενη ομάδα και να αποθαρρύνουν αναθέσεις που θα αύξαναν την ανισορροπία τους.

Παράλληλα, το αρχικό κριτήριο ανάθεσης του εκάστοτε αλγορίθμου διατηρείται και διατυπώνεται επίσης ως δύναμη έλξης. Στην ανίχνευση κοινοτήτων, η δύναμη αυτή εκφράζει την τοπική συνδεσιμότητα ενός κόμβου με μια κοινότητα, ενώ στην ομαδοποίηση εκφράζει τη χωρική εγγύτητα ενός σημείου.

Στο πλαίσιο αυτό, η εργασία προτείνει αρχικά τον αλγόριθμο Fair Label Propagation για την ανίχνευση κοινοτήτων. Ο αλγόριθμος βασίζεται στον κλασικό Label Propagation, αλλά τροποποιεί τον κανόνα ενημέρωσης των ετικετών προσθέτοντας μια δύναμη δικαιοσύνης. Έτσι, ένας κόμβος δεν ανατίθεται μόνο στην κοινότητα που εμφανίζεται συχνότερα στη γειτονιά του, αλλά λαμβάνεται επίσης υπόψη το κατά πόσο η ανάθεση αυτή βελτιώνει ή επιδεινώνει τη δημογραφική ισορροπία της κοινότητας.

Στη συνέχεια, η εργασία προτείνει τον αλγόριθμο Force-Field Fair K-Means για ομαδοποίηση σε μετρικούς χώρους. Ο αλγόριθμος διατηρεί τη βασική λογική του K-means, αλλά αντικαθιστά την απλή ανάθεση στο κοντινότερο κέντρο με έναν κανόνα ανάθεσης βασισμένο σε δυνάμεις. Η γεωμετρική δύναμη εξαρτάται από την απόσταση ενός σημείου από το κέντρο μιας ομάδας, ενώ η δύναμη δικαιοσύνης εξαρτάται από τη δημογραφική σύνθεση της ομάδας. Με αυτόν τον τρόπο, ο αλγόριθμος μπορεί να βελτιώνει τη δημογραφική ισορροπία χωρίς να αγνοεί πλήρως τη γεωμετρική δομή των δεδομένων.

Τέλος, παρουσιάζεται ο αλγόριθμος Force-Field Clustering, ο οποίος ξεκινά από μια βασική προσέγγιση ομαδοποίησης μέσω δυναμικών πεδίων και στη συνέχεια επεκτείνεται σε μια εκδοχή με γνώμονα τη δικαιοσύνη, τον Fair Force-Field Clustering. Η επέκταση αυτή ενσωματώνει δημογραφικές δυνάμεις στο ίδιο πλαίσιο, ώστε η διαδικασία ομαδοποίησης να λαμβάνει υπόψη τόσο τη γεωμετρική δομή των δεδομένων όσο και τη δημογραφική ισορροπία. Με αυτόν τον τρόπο, η ομαδοποίηση αντιμετωπίζεται ως μια δυναμική διαδικασία αλληλεπίδρασης μεταξύ σημείων, όπου η δομή των δεδομένων και η δικαιοσύνη επηρεάζουν από κοινού την εξέλιξη των αναθέσεων.

Η πειραματική αξιολόγηση πραγματοποιείται σε συνθετικά και πραγματικά δεδομένα. Τα αποτελέσματα δείχνουν ότι οι προτεινόμενες μέθοδοι μπορούν να βελτιώσουν τη δημογραφική ισορροπία των παραγόμενων ομάδων ή κοινοτήτων. Παράλληλα, αναδεικνύεται η αναμενόμενη αντιστάθμιση μεταξύ δικαιοσύνης και ποιότητας. Συνολικά, η εργασία δείχνει ότι τα force-based μοντέλα αποτελούν έναν ευέλικτο και ερμηνεύσιμο τρόπο ενσωμάτωσης κριτηρίων δικαιοσύνης σε αλγορίθμους

μη επιβλεπόμενης μάθησης.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Assoc. Professor Panayiotis Tsaparas, for his guidance, support, and encouragement throughout the course of this thesis. His valuable feedback and insightful suggestions were essential in shaping this work and helped me develop a deeper understanding of the research topic.

CHAPTER 1

INTRODUCTION

1.1 Motivation

1.2 Thesis Overview and Contributions

1.3 Thesis Roadmap

1.1 Motivation

The AI revolution of the past decade has led to a world where decisions affecting human lives are increasingly assisted by, or deferred entirely to, algorithmic systems trained on massive datasets. These decisions operate at multiple scales: from the individual level— influencing simple choices like where to dine or which movie to watch—to critical organizational and societal determinations, such as financial loan approvals, academic admissions, and sentencing decisions within the judiciary system.

Given the profound impact of these automated systems, there is growing concern regarding their ethical integrity. These concerns are corroborated by a substantial body of empirical evidence suggesting that algorithms often encode and amplify social biases. Prominent examples include the COMPAS recidivism prediction system, which was shown to exhibit bias against African-American inmates [1], and advertising algorithms that disproportionately displayed lower-paying job ads to women [2]. In response to these challenges, Algorithmic Fairness has emerged as a critical area of research within Responsible AI, aiming to design algorithms with formal fairness guarantees [3].

Although significant effort has been directed toward mitigating bias in supervised learning, attention has shifted to unsupervised learning tasks, particularly clustering, which serves as a fundamental step in data analysis. Clustering is extensively used in recommendation systems, medical diagnostics, and market segmentation, often acting as a pre-processing step for downstream predictive models. However, standard clustering algorithms are "colorblind"; they optimize for geometric objectives without regard for demographic composition. This can lead to "disparate impact" in high-stakes applications, where minority or under-represented groups are segregated or unfairly distributed across clusters.

1.2 Thesis Overview and Contributions

This thesis studies fairness-aware partitioning in two unsupervised learning settings: community detection and data clustering. In community detection the input is a graph, while in data clustering the input is a collection of data points in some multi-dimensional (metric) space. In both cases, we want to partition the input into cohesive groups, with respect to connectivity, or distance. The main goal is to incorporate demographic fairness into algorithms that normally optimize only structural or geometric quality.

We introduce a unified physics-inspired framework for this problem. In this framework, clusters and communities are assigned a fairness *charge* according to their demographic balance. The sign of the charge indicates the over-represented sensitive group, while the magnitude of the charge captures the degree of imbalance. These charges create a *fairness force* that pulls together nodes or points that improve demographic balance, and pushes away those that increase imbalance.

At the same time, the original objective of the underlying algorithm is preserved. In community detection, this objective is related to the local connectivity between nodes and communities. In clustering, it is related to distances and spatial proximity in the feature space. Therefore, the proposed framework does not replace the original algorithmic objective. Instead, it modifies the local assignment or movement rule by combining structural attraction with a fairness-aware force.

Within this framework, we develop three algorithms. First, we propose the Fair Label Propagation (FLP) algorithm for community detection. The Label Propagation

algorithm iteratively assigns the nodes in the graph, to the label that is most popular in their neighborhood, i.e., the community with the strongest pull. The (FLP) algorithm extends the standard Label Propagation algorithm by adding a fairness force to the label update rule. A node is no longer assigned only according to the most frequent label in its neighborhood, but also according to the effect that this assignment has on the demographic balance of the candidate community.

Second, we propose the Force-Field Fair K-Means (F3KM) algorithm for centroid-based clustering. The method keeps the main structure of K -means, but replaces the nearest-centroid assignment rule with a force-based score. This score combines the geometric attraction between a point and a centroid with a fairness correction based on the demographic balance of the corresponding cluster.

Third, we propose a force-based clustering method in which data points are treated as particles that move in the feature space. We first describe the base Force-Field Clustering method, which uses only geometric interactions between particles, without fairness considerations. We then extend it into Fair Force-Field Clustering (F3C), where each active particle has a fairness charge, and interacts with other particles through both geometric and fairness forces.

In summary, the main contributions of this thesis are:

- We introduce a novel unified physics-inspired framework for incorporating fairness into clustering and community detection through fairness charges, and fairness forces.
- We introduce FLP, a fairness-aware version of the Label Propagation algorithm for community detection, which combines neighborhood-based label updates with a fairness force.
- We introduce F3KM, a fairness-aware version of K -means, which modifies the point-to-centroid assignment step using a force-based fairness correction.
- We introduce F3C, a fairness-aware extension of a force-based clustering method, where moving particles represent groups of points and fairness is applied through interactions between these groups.
- We evaluate the proposed methods on synthetic and real datasets, studying the trade-off between demographic fairness and structural or clustering quality.

1.3 Thesis Roadmap

The rest of the thesis is organized as follows.

- Chapter 2 reviews related work on fairness, fairness-aware clustering and fairness-aware community detection.
- Chapter 3 introduces the basic definitions, the balance metrics, and the general physics-inspired fairness framework used throughout the thesis.
- Chapter 4 presents the Fair Label Propagation algorithm for community detection.
- Chapter 5 presents the Force-Field Fair K-Means algorithm for centroid-based clustering.
- Chapter 6 presents the Force-Field Clustering method and its fairness-aware extension, Fair Force-Field Clustering.
- Chapter 7 summarizes the findings and discusses possible directions for future work.

CHAPTER 2

RELATED WORK

2.1 Fairness Metrics in Clustering and Community Detection

2.2 Methodologies for Mitigating Bias and Achieving Fairness

This chapter reviews prior work on fairness in unsupervised learning, with emphasis on clustering and community detection. We discuss the main fairness notions, mitigation strategies, and algorithmic approaches used to produce partitions that are both structurally meaningful and demographically balanced.

2.1 Fairness Metrics in Clustering and Community Detection

Clustering is an important data mining procedure often used for summarization, dimensionality reduction, and data analysis. Thus, ensuring a fair, balanced, and transparent environment in clustering outputs is critical [4]. Fairness definitions are fundamentally divided into two distinct paradigms: *group fairness*, which focuses on demographic representation, and *individual fairness*, which ensures consistent treatment across similar entities. While algorithmic fairness in general clustering is well-established, the literature on fairness in community detection remains more limited and has predominantly focused on group fairness. In community detection, most existing work focuses on group fairness, since communities are evaluated as groups

of nodes and the main goal is to ensure that different demographic groups are fairly represented or fairly connected within the detected communities.

Group Fairness Metrics. Group fairness assumes that the input data points are partitioned into distinct demographic groups and requires equitable treatment across these groups. The various metrics proposed in the literature can be broadly categorized into three main approaches. First, *demographic representation* metrics focus on the composition of the clusters. The foundational notion of *balance* [5, 6] requires that different groups are proportionally represented within each cluster, preventing complete segregation. Extensions like *bounded representation* [7, 8] relax this by enforcing specific upper and lower bounds on the fraction of each group allowed in any given cluster. Second, *equitable center selection* metrics are designed for center-based algorithms, aim to enforce some form of proportional or constrained representation in the selection of centers, preventing majority groups from monopolizing cluster centers [9, 10]. Third, *cost distribution and outlier equity* metrics shift the focus from representation to the spatial cost of clustering. These approaches capture different aspects, such as minimizing the maximum clustering cost, bounding deviations from ideal group proportions, or ensuring that clustering costs (e.g., average distance to centers) are not disproportionately incurred by any single group [11, 12], or that discarded outliers do not disproportionately belong to a specific minority [13]. In community detection, group fairness metrics focus on the network’s structural topology and edge connectivity. Key metrics include *balance fairness* and *modularity fairness* [14], which measure demographic fractions and structural integration compared to a random null model. Additionally, metrics like *group modularity* and *diversity modularity* [15, 16] explicitly evaluate how well different groups connect within and across communities.

Individual Fairness Metrics. Individual fairness diverges from demographic categories and is rooted in the principle that “similar individuals should be treated similarly” [17]. A common approach relies on *probabilistic and distributional similarity*. Since deterministic clustering might forcefully separate two very similar data points lying on a cluster boundary, these metrics mandate that the probability of two similar individuals being placed in different clusters (or having different assignment distributions) is strictly upper-bounded by their actual similarity score [18, 19]. Other formulations capture different forms of *distance-based or neighborhood-aware fairness constraints*. They require that the distance of any individual to their assigned center is comparable to the distances experienced by their closest peers, thus respecting local

neighborhood densities and individual spatial requirements [20, 21, 22]. While individual fairness in community detection is still an emerging area, recent research has introduced metrics to quantify unfairness at both the node and graph levels without relying on spatial distances between nodes. For instance, the Individual Bias (IB) metric measures node-level unfairness by comparing a node’s ground-truth and predicted community contexts. These contexts are represented by rows of community co-occurrence matrices, and their discrepancy is computed using a vectorial distance, such as cosine distance. The graph-level extension, denoted by IBG, summarizes individual unfairness across the network by measuring the dispersion of the node-level IB scores. [23]

2.2 Methodologies for Mitigating Bias and Achieving Fairness

2.2.1 Fairness-Aware Clustering Algorithms

The methodologies for mitigating bias and achieving fairness in clustering generally follow the standard taxonomy of approaches: pre-processing, in-processing, and post-processing. Pre-processing methods typically transform the input dataset to satisfy specific fairness metrics before applying a standard clustering algorithm. In-processing strategies fundamentally modify the clustering objective or optimization routine to inherently produce fair partitions. Conversely, post-processing techniques adjust the output of a classical algorithm to meet fairness constraints.

Pre-processing Bias Mitigation: The foundational metric for pre-processing to reduce bias is Balance, which requires that the demographic ratio within each cluster reflect that of the overall dataset. The seminal work by Chierichetti et al. [5] introduced the concept of fairlets which are minimal balanced subsets of points that can be aggregated to form fair clusters. Their approach decomposes the dataset into fairlets using a minimum cost flow formulation, after which any classical algorithm (e.g., k -centers, k -median) can be applied to the fairlet centers.

While the original formulation focused on binary protected groups, subsequent research by Rösner and Schmidt [6] significantly expanded this framework. They generalized the fairlet decomposition to accommodate an arbitrary number of protected groups with privacy constraints, providing a highly efficient approximation

algorithm with strict performance guarantees for this multi-group setting. Further improving scalability, Backurs et al. [3] used tree metric embeddings to perform the decomposition in near-linear time.

To handle massive datasets, recent work has focused on developing fair coresets, small and weighted subsets of data that provably approximate the fairness and clustering properties of the full dataset, thereby enabling efficient processing of large-scale data and streams. Specifically, Bandyapadhyay et al. [24] developed fair coresets for Euclidean and general metric spaces, answering open questions from the foundational works on fair coresets previously introduced by Schmidt et al. [25] and Huang et al. [26].

In a significant recent preprint, Setzu and Guidotti [27] proposed clusterlets, a fuzzy clustering approach. Instead of rigid fairlets, it initially generates highly cohesive monochrome clusters and subsequently matches them to minimize deviation.

In-processing Bias Mitigation: In-processing methods modify the core algorithmic logic to optimize fairness and utility simultaneously. To address Social Fairness—which minimizes the maximum average cost incurred by any demographic group Ghadiri et al. [11] proposed Fair-Lloyd, a variation of the standard k -means heuristic that shifts the focus from proportional representation to equitable error distribution. By specifically modifying the center-update step to equalize the average clustering cost across groups, this method ensures that no single demographic population incurs a disproportionate burden of the spatial error.

For Individual Fairness, Jung et al.[21] proposed a method which constrains the clustering such that every point is served by a center within a distance comparable to its local neighborhood density. This line of inquiry was extended by [28], who provided a (β, γ) -bicriteria approximation to optimize this radius constraint, and by [20] who introduced α -equitable centers.

Probabilistic frameworks by Anderson et al.[19] and Esmaeili et al [29], effectively treat similar individuals similarly by assigning them to clusters with similar probability distributions. The in-processing landscape has recently expanded to address robustness, non-convex shapes, and causality. Duppalala et al. [30] introduced Robust Fair Clustering, identifying that demographic labels are often noisy; their algorithm uses "uncertainty sets" to provide theoretical guarantees even when group membership is uncertain.

The Fair Clustering via Alignment (FCA) algorithm by [31] is an innovative in-

processing approach designed to achieve an optimal trade-off between clustering utility and group fairness. Unlike traditional pre-processing methods that rely on sub-optimal fairlet construction, FCA is built upon a novel mathematical decomposition that translates the perfectly fair K-means objective into two distinct components: a transport cost between different protected groups and a clustering cost within a common aligned space. The algorithm operates iteratively in two alternating phases until convergence. In the first phase, it solves a modified Kantorovich optimal transport problem via linear programming to find a joint probability distribution that optimally aligns data points from different demographic groups. In the second phase, it updates the cluster centers within this new aligned space using standard clustering techniques. Simultaneously optimizing the matching map and the cluster centers, FCA systematically avoids complex fairness constraints, in this way achieving near-perfect fairness without numerical instability, thus the authors report improved trade-offs in their experimental evaluation.

Ziko et al. [32] proposes a general variational framework for fair clustering that integrates a Kullback–Leibler (KL) divergence based fairness penalty into standard clustering objectives such as K-means, K-medians, and Normalized Cut. Unlike prior combinatorial or spectral approaches, the proposed method allows explicit control of the trade-off between clustering quality and fairness through a regularization parameter. The authors derive a tight upper bound using a concave–convex decomposition and optimize it via a Majorize–Minimize procedure, leading to scalable, closed-form updates with convergence guarantees. The method is flexible, parallelizable, and applicable to large-scale datasets with multiple demographic groups, achieving competitive fairness and clustering performance compared to state-of-the-art approaches.

In a recent research by Pan et al. [33] the authors combine the standard K-means error with two mathematical penalties: one for unfair demographic distribution and one for unbalanced cluster sizes. Because this combined equation is complex to solve, they use a coordinate descent optimization technique. The algorithm evaluates each data point individually, assigning it to the cluster that minimizes the overall cost—balancing spatial distance with fairness and size. This step-by-step process repeats until the clusters achieve both demographic fairness and structural balance.

Furthermore, Gupta et al. [34] propose an in-loop post-processing method to address the computational and qualitative inefficiencies of strict demographic parity. To resolve this, they introduce a novel, generalized notion called τ -ratio fairness,

which bounds the representation of any protected group within a cluster using a tunable parameter τ , thus allowing a fine-grained trade-off between clustering quality and fairness. To implement this notion, they propose the Fair Round-Robin Algorithm for Clustering (FRAC), a heuristic approach that integrates directly into the standard K-means loop. Instead of assigning data points purely based on spatial proximity, FRAC employs a round-robin allocation strategy where cluster centroids iteratively select their closest available points from stratified demographic pools until the τ -ratio quotas are satisfied. This guarantees computationally efficient, scalable, and provably fair clustering without severely degrading the traditional spatial objective function.

Post-processing Bias Mitigation: Post-processing approaches offer flexibility by adjusting the output of standard algorithms to satisfy fairness constraints. Bera et al.[35], addressed the Bounded Representation metric, where groups must appear within specific proportion bounds. Their approach first computes a standard clustering and then reassigns data points by solving a linear program that enforces representation constraints for protected groups within each cluster. The method guarantees that fairness constraints are approximately satisfied while the clustering cost remains within a constant factor of the optimal solution, thus providing a principled trade-off between fairness and clustering quality.

Similarly, Kleindessner et al. [36], focused on the Fair Representation of Centers. The method begins by running a greedy K -center algorithm that selects centers iteratively: starting with a random point, it repeatedly chooses the point farthest from all currently selected centers. This ensures diverse and representative prototypes. To incorporate fairness, the algorithm then adjusts these centers through swaps within clusters to meet a predefined demographic ratio. All distances are pre-computed, enabling linear-time execution during the main algorithm. The approach guarantees a 5-approximation for two groups and scales to multiple demographics via recursive adjustments. Experiments show that it produces balanced summaries with minimal runtime overhead, making it practical for real-world applications like image search engines.

Additionally, Zhou et al. recently proposed a computationally efficient [37], two-stage approach to enhance fairness in K-means clustering by focusing on adjusting cluster memberships at the boundaries rather than solving a complex constrained optimization problem from scratch. In the first stage, the standard K-means algorithm is run to establish high-quality clusters. In the second stage, the method identifies

a small subset of "promising" data points located near the cluster boundaries and switches their assignments to balance sensitive subpopulations. Two specific algorithms are introduced for this selection: *fcNearForeign*, which targets points far from their own centroid but near a foreign cluster centroid, and *fcGini*, which uses the Gini index to detect highly "mixed" points in a local neighborhood. This boundary-focused strategy allows for substantial improvements in fairness metrics with minimal impact on overall clustering quality.

2.2.2 Fairness-Aware Community Detection Algorithms

The literature on fairness in community detection remains limited. The work most related to ours is that of Kleindessner et al. [38, 39], which proposes a fair spectral algorithm aiming to achieve balanced communities. According to their approach, a clustering is considered fair when every demographic group is proportionally represented in each cluster, following the ratio present in the overall dataset. Technically, this is achieved by expressing the fairness constraint as a linear constraint and projecting the graph's Laplacian matrix onto a fair nullspace before computing the eigenvectors.

Within a similar node-balance framework, Panayiotou and Magnani [40] introduced the Fair-mod algorithm, which incorporates demographic fairness constraints into modularity-based methods. This method modifies the Louvain algorithm to maximize a joint objective function, which is the weighted sum of the overall modularity and the fairness score of the sensitive attribute distribution per community.

To formally evaluate the presence of bias within network communities, Manolis and Pitoura [14] introduced two distinct evaluation metrics for networks with binary sensitive attributes: *balance fairness* and *modularity fairness*. While balance fairness quantifies the deviation of a protected group's representation within a community from its global fraction, modularity fairness specifically assesses the structural integration of the group by measuring its intra-community connectivity relative to a random null model. Through an extensive evaluation of both real-world and synthetic networks, they demonstrated that community fairness is heavily compromised by group size imbalances and homophily.

Building on this need to focus on structural integration, a recent line of research by Gkartzios et al. [15, 41, 16] fundamentally shifts the focus of fairness in com-

munity detection from simple node representation to structural edge connectivity. Initially, [15] introduced the concepts of *group modularity* and *diversity modularity* to evaluate whether specific demographic groups are equally well-connected within their assigned communities and whether they maintain sufficient cross-group interactions. To optimize these metrics for binary sensitive attributes, they proposed fairness-aware variations of the agglomerative Louvain algorithm.

This connectivity-based fairness framework was subsequently expanded to spectral clustering and deep learning methodologies [41]. By mathematically decomposing the adjacency and modularity matrices into sub-matrices that isolate intra-group and inter-group edges, two distinct algorithmic families were developed. The *input-based* methods modify the matrices fed into spectral algorithms and Graph Convolutional Networks, while the *loss-based* methods embed the group modularity and diversity constraints directly into the optimization loss function, allowing the models to learn fair community representations without altering the underlying graph structure.

Most recently, this approach was generalized to handle the increased complexity of networks with multi-valued sensitive attributes (e.g., multiple nationalities or education levels) [16]. To address settings with more than two groups, novel metrics were formalized, such as the *multi-fairness ratio*—which penalizes the connectivity gap between the most and least connected groups—and *group-aware diversity*, which ensures that every individual minority group participates in cross-group mixing. To optimize these new metrics, advanced deep community detection models (e.g., MultiDeepGroup and MultiDeepFairness) were introduced, successfully improving the internal connectivity and visibility of the most disadvantaged groups without significantly degrading the overall structural modularity of the network.

Finally, the problem of fairness has also been studied in the context of the densest subgraph problem by Anagnostopoulos et al. [42]. The goal there is to find a subgraph of maximum density while ensuring that no group is disparately impacted, containing an equal number of representatives from the protected labels. Their technique is also based on spectral relaxations, projecting the adjacency matrix into a “fair” subspace to enforce soft constraints. The densest subgraph problem is related to community detection, as both search for densely connected subsets of nodes, but it has a fundamentally different objective, as it does not aim to partition all nodes of the graph into distinct communities.

CHAPTER 3

PRELIMINARIES

3.1 Community Detection

3.2 Data Clustering

3.3 Fairness Metric: Balance

3.4 A Physics-Inspired Framework for Fairness

In this chapter, we introduce the formal definitions of community detection and data clustering, alongside the balance metric used to quantify demographic fairness. Building upon these foundations, the primary focus of this chapter is to present the physics-inspired framework for fairness that will serve as the core methodology throughout this thesis.

3.1 Community Detection

In the community detection problem, we assume that we have as input a graph $G = (V, E)$, with n nodes $V = \{v_1, v_2, \dots, v_n\}$ and edge-set $E = \{(v_i, v_j)\}$. The output of a community detection algorithm is a partition of the nodes into k disjoint subsets (communities) $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$, where $C_i \subseteq V$, $C_i \cap C_j = \emptyset$, and $\cup_{i=1}^k C_i = V$. The number of communities k may be given as input, or it may be decided by the algorithm. There is a variety of community detection algorithms that use different criteria to produce communities [43]. Typically, we require that the nodes within a

community are densely connected, while being sparsely connected across different communities.

3.2 Data Clustering

Data clustering seeks to partition a set of elements based on their similarity in a feature or metric space. We assume as input a set of data points $X = \{x_1, x_2, \dots, x_n\}$, typically residing in a d -dimensional space $\mathbb{R}^d$. The output is a collection of k disjoint clusters $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$, where $C_i \subseteq X$, $C_i \cap C_j = \emptyset$ for $i \neq j$, and $\bigcup_{i=1}^k C_i = X$.

Unlike community detection, which relies on topological connections (edges), clustering typically depends on a distance or similarity measure (e.g., Euclidean distance) to group data points such that intra-cluster similarity is maximized while inter-cluster similarity is minimized.

3.3 Fairness Metric: Balance

To establish a unified fairness notation, we refer to the input entities, whether they are spatial data points or graph nodes, generically as elements. Let $X = \{x_1, \dots, x_n\}$ denote the set of elements, and let $\mathcal{C} = \{C_1, \dots, C_k\}$ denote the resulting output partition, where each C_j represents either a cluster or a community.

The fairness metric used in this thesis assumes that the input elements are partitioned into distinct demographic groups, defined by a sensitive attribute associated with the elements. For simplicity, we assume a binary attribute and two groups, referred to as the red group and the blue group. For each cluster or community C_j , let C_j^r and C_j^b denote the subsets of red and blue elements in C_j , respectively.

The balance of a cluster or community C_j is defined as

$$\text{bal}(C_j) = \begin{cases} \min \left\{ \frac{|C_j^r|}{|C_j^b|}, \frac{|C_j^b|}{|C_j^r|} \right\}, & \text{if } |C_j^r| > 0 \text{ and } |C_j^b| > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

The value of $\text{bal}(C_j)$ lies in $[0, 1]$. A perfectly balanced group contains an equal number of red and blue elements, resulting in a balance value of 1. If a cluster or community contains elements from only one demographic group, its balance is defined as 0, since such a group is completely demographically imbalanced.

Given a partition $\mathcal{C} = \{C_1, \dots, C_k\}$, we define its average balance as

$$\text{bal}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C_j \in \mathcal{C}} \text{bal}(C_j). \quad (3.2)$$

This is an average measure, meaning that each cluster or community contributes equally to the final fairness score.

In datasets where the two demographic groups are not equally represented, we also use a normalized version of balance. Let X^r and X^b denote the red and blue elements in the full dataset X , respectively. The normalized balance of C_j is defined as

$$\overline{\text{bal}}(C_j) = \begin{cases} \min \left\{ \frac{|C_j^r|/|C_j^b|}{|X^r|/|X^b|}, \frac{|C_j^b|/|C_j^r|}{|X^b|/|X^r|} \right\}, & \text{if } |C_j^r| > 0 \text{ and } |C_j^b| > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.3)$$

This measure evaluates how closely the demographic composition of C_j matches the global demographic distribution of the dataset. The average normalized balance of a partition is then defined as

$$\overline{\text{bal}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C_j \in \mathcal{C}} \overline{\text{bal}}(C_j). \quad (3.4)$$

3.4 A Physics-Inspired Framework for Fairness

The methods proposed in this thesis are based on a common force-based principle. Given a set of elements $X = \{x_1, \dots, x_n\}$, where each element may correspond either to a data point or to a graph node, the goal is to construct a partition $\mathcal{C} = \{C_1, \dots, C_k\}$ that preserves structural quality while improving demographic fairness.

The framework is inspired by Coulomb’s law, according to which charged objects exert attractive or repulsive forces depending on the sign of their charges. In our setting, this physical principle is used as an algorithmic analogy. The key idea is that demographic imbalance creates a *fairness force*, that is attractive between elements or groups that improve balance, and repulsive when they increase imbalance.

We assume that each element is associated with a binary sensitive attribute. We define fairness charges at the level of groups of elements. A group may correspond to a cluster, a community, or a singleton element.

For each cluster or community C_j , let $\text{bal}(C_j)$ denote the balance score used by the algorithm. Depending on the setting, this score may correspond either to the standard

balance $\text{bal}(C_j)$ or to the normalized balance $\overline{\text{bal}}(C_j)$, as defined in Section 3.3. The specific choice is stated separately for each algorithm.

The demographic *imbalance* of C_j is defined as

$$\text{imb}(C_j) = 1 - \text{bal}(C_j). \quad (3.5)$$

Thus, $\text{imb}(C_j) \in [0, 1]$. A perfectly balanced cluster or community has $\text{imb}(C_j) = 0$, while a completely imbalanced one has $\text{imb}(C_j) = 1$.

The fairness charge of C_j is then defined as

$$q_j = \begin{cases} +\text{imb}(C_j), & \text{if the red group is over-represented in } C_j, \\ -\text{imb}(C_j), & \text{if the blue group is over-represented in } C_j, \\ 0, & \text{if } C_j \text{ is balanced.} \end{cases} \quad (3.6)$$

When standard balance is used, over-representation is determined directly by the numbers of red and blue elements in C_j . Therefore, $q_j > 0$ when C_j contains more red than blue elements, and $q_j < 0$ when it contains more blue than red elements. When normalized balance is used, over-representation is interpreted relative to the global demographic distribution of the dataset. In this case, the group whose proportion in C_j exceeds its global proportion is treated as the over-represented group.

The sign of q_j indicates which demographic group is over-represented in C_j , while $|q_j|$ indicates how imbalanced C_j is. If C_j is balanced, then $q_j = 0$, and no fairness force is exerted by that group.

The same definition also applies to groups that contain a single element. In algorithm-specific chapters, such one-element groups are used when a single node or data point interacts with a community or cluster.

Following the intuition of Coulomb's law, the interaction between two charged objects is proportional to the product of their charges and inversely proportional to the square of their distance:

$$F = K_e \frac{q_1 q_2}{r^2}. \quad (3.7)$$

In our setting, this law is used as an algorithmic analogy rather than as a physical model. Therefore, we set the proportionality constant to $K_e = 1$. The signed fairness

interaction between two groups C_i and C_j is then defined as

$$F_{ij}^e = -\frac{q_i q_j}{d_{ij}^2},$$

where q_i and q_j are the fairness charges of C_i and C_j , respectively, and d_{ij} denotes the distance or effective separation between the two groups.

This definition follows the Coulomb-style intuition that groups with opposite charges attract each other, while groups with the same sign repel each other. In terms of fairness, attraction encourages interactions between groups dominated by different demographic populations, while repulsion discourages interactions that would further reinforce demographic imbalance. If one of the two groups is balanced, then its charge is zero and no fairness force is applied through that group.

Fairness alone is not sufficient, since the produced groups must also preserve the structure of the data. Therefore, we also introduce a *structural attraction force*, denoted by F_{ij}^{attr} . This force models the original utility objective of the underlying algorithm. Its exact form depends on the setting: in centroid-based clustering, it is related to the distance between a point and a cluster, while in graph-based community detection, it is related to the connectivity between a node and a community.

The final assignment is based on a combined force score:

$$S_{ij} = \alpha F_{ij}^{attr} + \beta F_{ij}^e, \quad (3.8)$$

where $\alpha, \beta \in [0, 1]$ controls the relative contribution of the structural and fairness forces, respectively. The exact weighting of these two terms depends on the specific algorithm. In the following chapters, this combined score is adapted to the corresponding assignment or update rule.

Each element is assigned to the cluster or community that exerts the strongest combined force :

$$C(x_i) = \arg \max_{C_j \in \mathcal{C}} S_{ij}. \quad (3.9)$$

In each algorithm, the combined score is adapted to the corresponding assignment rule. When the fairness term is removed, the method reduces to the original algorithm. As the fairness term becomes stronger, assignments that improve demographic balance are favored, while assignments that increase imbalance are discouraged. Thus, the proposed framework keeps the original structural objective, but modifies the local assignment rule with a Coulomb-inspired fairness force.

In assignment-based algorithms, such as the methods presented in Chapters 4,5 the combined force is used to select the cluster or community that receives an element. In movement-based algorithms, such as the particle-based method presented in Chapter 6, the same principle is used to define the movement of active particles before they are merged into clusters.

CHAPTER 4

FAIR LABEL PROPAGATION

4.1 The Standard Label Propagation Algorithm

4.2 The Fair Label Propagation Algorithm (FLP)

4.3 Experiments

4.4 Discussion

In this Chapter, we apply the physics-inspired framework introduced in Chapter 3 to the problem of community detection in networks. Specifically, we propose the Fair Label Propagation (FLP) algorithm, which extends the popular Label Propagation (LP) algorithm [44] to produce demographically balanced communities.

4.1 The Standard Label Propagation Algorithm

The label propagation algorithm (LP) for community detection was first introduced by Raghavan et al. [44]. The algorithm starts by assigning each node a unique label, usually the ID of the node. Then it proceeds iteratively. In each iteration, each node updates its label, adopting the most frequent label in its neighborhood. If there are ties, the label with the largest ID is selected. The algorithm terminates when no node changes its label.

The result of the LP algorithm is that densely connected nodes are likely to adopt the same label. The algorithm is highly efficient, as each iteration runs in time linear

to the number of edges in the graph. It also has the advantage that it does not require the number of communities (k) to be specified in advance.

A known weakness of the algorithm is that, when updates are asynchronous, the resulting communities depend heavily on the random order in which the nodes are considered. Conversely, with fully synchronous updates, convergence is not guaranteed [45, 46]. To address these issues, the work in [46] proposed a variation of the LP algorithm that performs *semi-synchronous* updates.

The semi-synchronous algorithm first uses a greedy approach to obtain a coloring of the graph, such that no two connected nodes share the same color. Nodes with the same color can then update their labels synchronously, since the label of one does not directly affect the other in the same step. In the presence of ties, a predetermined order is followed (e.g., selecting the largest label ID) so that results remain consistent across different runs. The stopping criterion is met when all nodes in the graph hold the most frequent label of their neighborhood.

4.2 The Fair Label Propagation Algorithm (FLP)

In this section, we present the Fair Label Propagation algorithm (FLP). Applying the physics-inspired framework introduced in Chapter 3, our algorithm builds upon the vanilla LP algorithm. For the definition of our algorithm, we take a physics-inspired view of the LP algorithm. We assume that adjacent nodes (u, v) in the graph pull each other with force $F_p(u, v) = 1$. In an iteration of the LP algorithm, if $N_\ell(v)$ is the set of neighbors of v with label ℓ , then the community C_ℓ pulls node v with force $F_p(C_\ell, v) = |N_\ell(v)|$. Node v is assigned to the community C_ℓ that pulls node v the strongest, that is, the most popular community in the neighborhood of v .

To incorporate fairness into the LP algorithm, we follow the fairness-charge framework introduced in Chapter 3. In this chapter, we use the standard, unnormalized balance score defined in Eq. (3.4). Therefore, each community C_ℓ is assigned a fairness charge q_ℓ according to its current demographic composition, as described in Chapter 3: the sign of q_ℓ indicates the over-represented sensitive group, while its magnitude captures the imbalance of the community.

A node v is treated as a singleton group when it interacts with a candidate community. Thus, its charge is determined directly by its sensitive attribute: red nodes

have positive charge and blue nodes have negative charge.

In the general force-based framework, the fairness interaction between two charged objects depends on the product of their charges and on their distance(3.4). In the graph setting of Label Propagation, however, assignments are based on local neighborhoods rather than on Euclidean distances. Therefore, for each edge between v and a neighbor belonging to community C_ℓ , we use an effective graph distance equal to one and set the proportionality constant to one.

As a result, each neighbor of v with label ℓ contributes a fairness interaction proportional to $-q_v q_\ell$. Since there are $|N_\ell(v)|$ such neighbors, the total fairness force exerted by community C_ℓ on node v is defined as

$$F_e(C_\ell, v) = -q_v q_\ell |N_\ell(v)|.$$

The factor $|N_\ell(v)|$ captures the local connectivity between v and C_ℓ , while the sign of $-q_v q_\ell$ determines whether the assignment is encouraged or discouraged from a fairness perspective. If the candidate community is already balanced, then its charge is zero and the fairness term has no effect.

Thus, the electrostatic force captures the effect of assigning node v to community C_ℓ on the balance of that community.

The FLP algorithm computes the total force $F(C_\ell, v)$ that the community C_ℓ exerts on node v , by combining the pulling and electrostatic forces, that is:

$$F(C_\ell, v) = (1 - \lambda)F_p(C_\ell, v) + \lambda F_e(C_\ell, v),$$

where λ is a parameter of the algorithm. By combining the two forces, the algorithm aims to combine two objectives: The quality of the output communities, which is achieved by the pulling force, as in the vanilla LP algorithm, and the fairness of the output communities, which is achieved by the electrostatic force. The parameter λ controls the tradeoff between community quality and fairness. Higher values of λ place more emphasis on fairness. The value $\lambda = 0$ corresponds to the vanilla LP algorithm, while the case $\lambda = 1$ puts all the emphasis on fairness.

Putting everything together, the FLP algorithm operates exactly as the vanilla LP algorithm, iteratively updating the labels of the nodes, each time assigning a node v to the community C_ℓ that exerts the strongest force $F(C_\ell, v)$. The outline of the algorithm is shown in Algorithm 4.1. The algorithm complexity remains $O(Im)$, with m edges, and I iterations.

Algorithm 4.1 Fair Label Propagation (FLP)

Require: Graph $G = (V, E)$, group partition $\{G_b, G_r\}$, parameter λ .

```
1: Assign a unique label  $L(v) = v$  to every node  $v \in V$ .
2: repeat
3:   Update community charges  $q_\ell$ 
4:   for all  $v \in V$  do
5:     for all labels  $\ell$  do
6:        $N_\ell(v) = \{u \in N(v) : L(u) = \ell\}$ 
7:        $F_p(C_\ell, v) = |N_\ell(v)|$ ;  $F_e(C_\ell, v) = -q_v q_\ell |N_\ell(v)|$ 
8:        $F(C_\ell, v) = (1 - \lambda)F_p(C_\ell, v) + \lambda F_e(C_\ell, v)$ 
9:     end for
10:     $L(v) = \arg \max_\ell F(C_\ell, v)$ 
11:   end for
12: until No label change
```

4.3 Experiments

The goal of the experiments is two-fold: Explore the fairness–quality trade-off, by varying the parameter λ of the FLP algorithm; Compare against baselines and assess how dataset characteristics affect performance in terms of fairness and clustering quality.

4.3.1 Experimental Setup

Datasets: In our experiments, we use both real and synthetic datasets. We use the following real datasets:

- *Twitter* [47]: A collection of Twitter users, and their retweet actions.
- *Facebook* [48]: The mutual friendships between a collection of Facebook users, extracted from their ego networks.
- *Deezer* [49]: The mutual follow relationships between a collection of users on Deezer, an online music platform.
- *DrugNet* [50]: The acquaintance relationships between a collection of drug users in Hartford.

Table 4.1: Real Dataset Characteristics

Dataset	Nodes	Edges	Groups	bal(G)
<i>Twitter</i>	18,470	48,053	affiliation	0.63
<i>Facebook</i>	4,039	88,234	gender	0.61
<i>Deezer</i>	28,281	92,752	gender	0.80
<i>DrugNet</i>	185	265	gender	0.27
<i>Friendship</i>	127	396	gender	0.67

- *Friendship* [51]: The mutual friendships between students in a high school in Marseilles, France.

The sensitive attribute is the political affiliation, for the *Twitter* network, and the gender, for all other datasets. The characteristics of the datasets, and the balance, $\text{bal}(G)$, of the whole network are shown in Table 4.1.

We also use synthetic datasets, generated by a variant of the stochastic block model, defined in [38]. The model assumes that the nodes are partitioned into k planted *clusters*, $T = \{T_1, \dots, T_k\}$, and two groups $G = \{G_r, G_b\}$. The model is defined by four parameters: a, b, c , and d that determine the probability $\Pr(u, v)$ of an edge between two nodes u, v , depending on the group and cluster membership. Specifically: $\Pr(u, v) = a$, if u, v belong to the same cluster, and the same group; $\Pr(u, v) = b$, if u, v belong to different clusters, but the same group; $\Pr(u, v) = c$, if u, v belong to the same cluster, but in different groups; $\Pr(u, v) = d$, if u, v belong to the different clusters, and different groups.

We have $a > b > c > d$. The datasets are constructed so that traditional algorithms tend to produce monochromatic communities, by separating nodes from different groups. The goal is to study if fair community detection algorithms can generate fair (balanced) communities, ideally by recovering the planted clusters.

Algorithms: In our implementation of the FLP algorithm, we perform semi-synchronous updates of the node labels, adopting the update scheme in [46]. We also perform random permutations of the initial labels of the nodes, to deal with the randomness in the node ordering. We implemented FLP by adapting the open-source implementation provided by the NetworkX library.¹ Our code is publicly available.²

¹https://networkx.org/documentation/stable/_modules/networkx/algorithms/community/label_propagation.html

²<https://github.com/elidek-themis/flp>

We compare the FLP algorithm against the Fair Spectral algorithm (FSP) introduced in [38]. The algorithm incorporates the fairness constraints in the Laplacian matrix. It computes the eigenvectors of the k smallest eigenvalues, and performs k -means clustering on the resulting vectors. In our implementation we use the scalable variant of the algorithm [39], and we employ k -means++ for the clustering step.

Fairness and Quality metrics: We evaluate the output communities in terms of both fairness and quality. Fairness is measured using the balance metric defined in Equation 3.2. The quality of the communities is measured using *modularity* [52]. Modularity is a popular measure of community quality, that measures the divergence between the actual and the expected number of intra-community edges, if edges were generated at random such that the expected node degrees are preserved. Specifically, the modularity $Q(C)$ of a community $C \in \mathcal{C}$ is:

$$Q(C) = \frac{1}{2m} \left(\sum_{u,v \in C} A_{uv} - \frac{|N(u)||N(v)|}{2m} \right),$$

where A is the adjacency matrix of G , m the number of edges in G , and $|N(u)|$, $|N(v)|$ the degrees of node u and v respectively. The higher the modularity, the more cohesive the community. The modularity of a collection of communities is the sum of the modularities of the communities. Note that the modularity of the whole graph is zero.

4.3.2 Experimental results

Fairness-Quality trade-off: We first study the effect of the parameter λ on the performance of our algorithm. The parameter λ controls the contribution of the electrostatic force in the label assignment, and the influence of the fairness criterion on the community formation process. The values of λ between 0 and 1 implement a trade-off between finding well-connected communities, and achieving fairness.

To study the effect of λ , we use synthetic datasets, for which we have control over the dataset characteristics. We set the parameter values as follows: $a = 0.1$, $b = 0.01$, $c = 0.001$, and $d = 0.0001$. The number of groups is 2 (red/blue), and for the number of clusters we use the values $k \in \{2, 3, 4, 5\}$. Each cluster has 400 nodes, 200 of which are red and 200 are blue. The parameter λ takes values from 0 to 1, in steps of 0.1. We measure fairness, quality, and the number of clusters. Our measurements are averages over 10 different datasets, and 10 different label permutations per dataset.

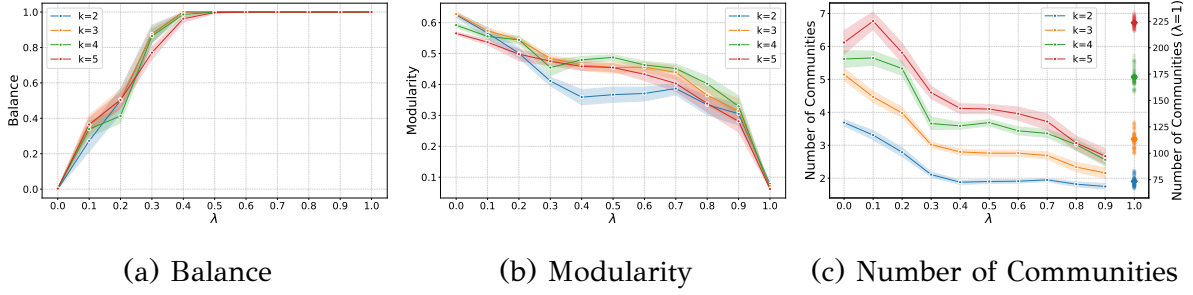


Figure 4.1: Fairness-Quality trade-off

Figure 4.1 shows our results. We observe that the vanilla LP algorithm ($\lambda = 0$) achieves the highest modularity, but very poor fairness, since the communities created are monochromatic. As the value of λ increases the fairness of the output improves, reaching the maximum for $\lambda \geq 0.4$. At the same time, modularity decreases slowly, up to $\lambda = 0.8$, and then drops sharply. For $\lambda = 0.9$, the strength of the electrostatic force results in grouping together nodes from different colors even when they belong to different clusters. These communities are sparsely connected, thus bringing modularity down. Sometimes, a single community is output, with modularity zero.

In the extreme case where $\lambda = 1$, the label selection is based solely on the electrostatic force. This forces bi-chromatic edges to merge, and same-color edges to split. Communities are built around bi-chromatic edges, resulting in a large number of communities, as shown in the right axis of Figure 4.1c, usually close to the number of bi-chromatic edges.

Overall, we observe that for λ in $[0.4, 0.7]$ we strike a good balance between fairness and quality. For the following, we use $\lambda = 0.5$ as the default value.

Comparison on Synthetic Datasets: We now evaluate the performance of the FLP algorithm, as we vary the “hardness” of the synthetic datasets, and we compare with the fair spectral algorithm (FSP). Specifically, we generate synthetic datasets where we vary the ratio b/c of the probability of connecting two nodes of the same color, that belong to different clusters, over the probability of connecting two nodes of different color that are in the same cluster. We think of the former edges as impeding the creation of fair communities, while the latter as facilitating the creation of fair communities. The higher the ratio, the harder for an algorithm to bring together nodes of different color in the same community.

We consider the values $\{5, 10, 15, 20, 25\}$ for the ratio b/c . We vary the ratio b/c

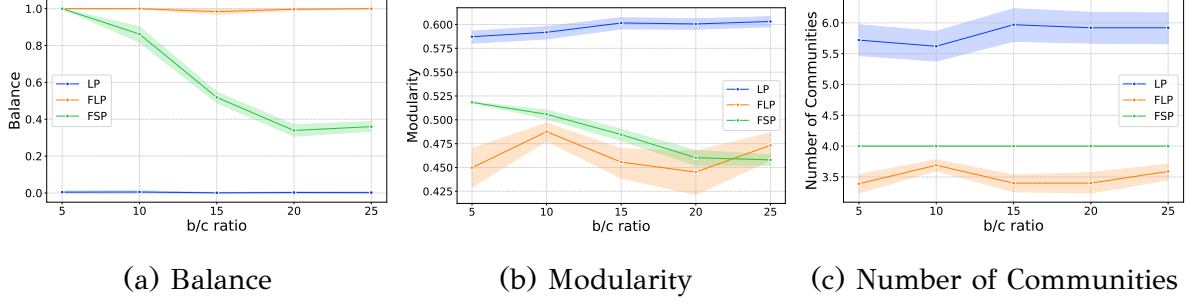


Figure 4.2: Comparison on synthetic datasets of different hardness.

by varying the parameter c . We use the default values for the parameters a, b, d (0.1, 0.01, 0.0001), and we vary the parameter c to take values 0.002, 0.001, 0.0006, 0.0005, 0.0004. We set the number of clusters to $k = 4$. We use $\lambda = 0.5$ for the FLP algorithm, and we also consider the vanilla LP algorithm. Again, we report averages over 10 different random graphs, and 10 label permutations per graph.

Figure 4.2 shows our results. We observe that our algorithm achieves almost perfect balance for all settings of the ratio b/c . On the other hand, the balance for the spectral algorithm deteriorates as the dataset becomes harder. As bi-chromatic edges become increasingly rare in comparison to single color edges, the FSP algorithm tends to bring together nodes of the same color and separate nodes of different color, leading to unfairness.

The FLP algorithm has lower modularity than the spectral algorithm. This is expected from the fairness-quality trade-off, since the FLP algorithm brings together nodes of different color, which are sparsely connected, while the FSP algorithm is more likely to create monochromatic communities, which have higher modularity.

We observe that the FLP algorithm on average outputs less than 4 communities, the number of planted clusters. This indicates that the FLP algorithm is more aggressive in merging clusters to preserve balance.

Results on Real Datasets: Finally, we compare our algorithms on the real datasets. Table 4.2 shows the results. For the Label Propagation algorithms the results are the averages of 10 different runs with different label permutations. Since we have no ground truth for the “correct” number of communities, we also perform 10 runs of the FSP algorithm with the number of communities output by the FLP algorithm in the corresponding run, and report average values.

The first observation is that the FLP algorithm performs best in terms of balance

Table 4.2: Algorithm results on real datasets

Dataset	Algorithm	Balance	Modularity	Communities
<i>Twitter</i>	LP	0.010	0.549	1189.3
	FLP	0.579	0.040	69.0
	FSP	0.018	0.083	69.0
<i>Facebook</i>	LP	0.460	0.734	38.2
	FLP	0.654	0.798	66.5
	FSP	0.563	0.596	66.5
<i>Deezer</i>	LP	0.479	0.469	1697.8
	FLP	0.767	0.462	1728.5
	FSP	0.493	0.452	1728.5
<i>DrugNet</i>	LP	0.218	0.602	43.0
	FLP	0.367	0.656	17.2
	FSP	0.345	0.683	17.2
<i>Friendship</i>	LP	0.390	0.662	16.1
	FLP	0.632	0.659	13.0
	FSP	0.398	0.659	13.0

on all datasets, sometimes being the clear winner (e.g., on the *Friendship* or *Deezer* networks). It is also competitive in terms of modularity: It has the best modularity for *Facebook* and close to the best in all datasets except for *Twitter*. Perhaps surprisingly, the LP algorithm does not always yield the best modularity, although its modularity is close to the best. This indicates that the fair algorithms are able to maintain community quality, while achieving fairness.

For the *Twitter* dataset, the FLP algorithm typically produces a very large community, and several smaller ones, yielding high balance, but low modularity. A similar behavior is also observed for the FSP algorithm, however, in this case, the small communities are monochromatic, bringing down the average balance.

4.4 Discussion

In this algorithm, we addressed fairness in community detection by proposing a physics-inspired fair variant of the Label Propagation algorithm. Experiments on real and synthetic datasets show that our method effectively identifies fair, high-quality communities, and outperforms existing baselines. In future work, we plan to explore alternative formulations of the pulling and electrostatic forces, as well as extensions of our algorithm for vector datasets.

We also experimented with normalized balance. Since the results were qualitatively similar to those obtained with the standard balance metric, we omit them for brevity.

CHAPTER 5

FORCE-FIELD FAIR K-MEANS

5.1 Standard K-means

5.2 The Force-Field Fair K-Means (F3KM)

5.3 Experiments

5.4 Discussion

In this chapter we apply the physics-inspired framework introduced in Chapter 3 to the problem of data clustering. Specifically, we propose the Force-Field Fair K-means algorithm, which integrates demographic balance into the assignment process of K-means through a fairness-aware force score.

5.1 Standard K-means

K-means is one of the most widely used unsupervised learning algorithms for data clustering. It is an iterative, centroid-based method that partitions a dataset into k groups of similar points based on their distance to cluster centers.

Given a set of data points $X = \{x_1, x_2, \dots, x_n\}$, where each point $x_i \in \mathbb{R}^d$, and a desired number of clusters k , the algorithm aims to partition the points into clusters $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ that minimize the sum of squared distances between points and

their assigned centroid:

$$\min_c \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|_2^2 \quad (5.1)$$

The standard K-means algorithm, as proposed by Lloyd [53], proceeds as follows:

- Select k initial centroids at random.
- Assign each data point to the cluster with the closest centroid.
- Update each centroid as the mean of the points assigned to its cluster.

This procedure repeats until the cluster assignments no longer change. Formally, the centroid update for each cluster is given by

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \quad (5.2)$$

5.2 The Force-Field Fair K-Means (F3KM)

Physics-inspired formulation. The proposed method builds on K -means but gives its assignment step a force-based interpretation. In standard K -means, each point x_i is assigned to the nearest centroid:

$$\arg \min_j \|x_i - \mu_j\|_2^2 = \arg \max_j \frac{1}{\|x_i - \mu_j\|_2^2}. \quad (5.3)$$

We reinterpret this distance-based assignment as a geometric attraction force. Let $d(x_i, \mu_j)$ denote the distance between point x_i and centroid μ_j . The geometric force is defined as

$$F_g(x_i, \mu_j) = \frac{1}{d(x_i, \mu_j)^2}. \quad (5.4)$$

Thus, closer centroids exert stronger geometric attraction, while the attraction decreases as the distance increases.

Fairness notation. We use the fairness-charge notation introduced in Chapter 3. In this chapter, the balance term used by the algorithm may correspond either to the standard balance or to the normalized balance, depending on the experiment.

In the clustering setting, each centroid μ_j represents a cluster C_j . Therefore, the centroid is assigned the fairness charge of the cluster it represents:

$$q_{\mu_j} = q_j.$$

The charge q_j is computed from the demographic composition of C_j , following the definition of Chapter 3. Its sign indicates the over-represented sensitive group, while its magnitude captures the imbalance of the cluster. When the standard balance is used, over-representation is determined from the red and blue counts inside C_j . When the normalized balance is used, it is interpreted relative to the global demographic distribution of the dataset.

Each point x_i is treated as a singleton group when it interacts with a candidate cluster. Its charge q_{x_i} is therefore determined directly by its sensitive attribute, according to the same charge convention.

Force interpretation of fairness. Following the group-level fairness interaction defined in Chapter 3, Force-Field Fair K-Means F3KM applies the fairness force between a point x_i and a candidate cluster C_j , represented by centroid μ_j . The signed fairness force is

$$F_e(x_i, \mu_j) = -\frac{q_{x_i}q_{\mu_j}}{d(x_i, \mu_j)^2}. \quad (5.5)$$

A positive value attracts x_i toward C_j , indicating that the assignment would improve the demographic balance of the cluster. A negative value repels x_i from C_j , indicating that the assignment would increase its imbalance.

Combined force and assignment rule. In F3KM, we keep the geometric attraction active and use λ to control the strength of the fairness correction:

$$\begin{aligned} F(x_i, \mu_j) &= F_g(x_i, \mu_j) + \lambda F_e(x_i, \mu_j) \\ &= (1 - \lambda q_{x_i}q_{\mu_j}) \frac{1}{d(x_i, \mu_j)^2}. \end{aligned} \quad (5.6)$$

Each point is assigned to the cluster that exerts the maximum total force:

$$l_{x_i} = \arg \max_j F(x_i, \mu_j). \quad (5.7)$$

When $\lambda = 0$, the method reduces to standard K -means. As λ increases, the fairness correction has a stronger effect on the assignment scores.

Algorithm. The F3KM algorithm is summarized in Algorithm 5.1. Starting from an initial cluster configuration, the algorithm alternates between a fairness-aware assignment step and a centroid update step. In each iteration, points are assigned to the clusters that exert the strongest combined force, and the centroids are then

updated according to the selected update rule. The algorithm terminates when the centroids no longer change up to a numerical tolerance, when a two-step centroid oscillation is detected, or when the maximum number of iterations is reached.

Algorithm 5.1 Force-Field Fair K-Means (F3KM)

```

1: Input: Dataset  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , sensitive attributes  $A$ , number of clusters  $k$ ,
   maximum iterations  $T_{\max}$ , parameter  $\lambda$ 
2: Output: Final clusters  $C_1, C_2, \dots, C_k$  and centroids  $\mu_1, \mu_2, \dots, \mu_k$ 
3: Initialize clusters, centroids, and clustering charges
4: for  $t = 1$  to  $T_{\max}$  do
5:   Step 1: Fairness-aware assignment
6:   for each point  $\mathbf{x}_i \in X$  do
7:      $l_{x_i} = \arg \max_j F(\mathbf{x}_i, \mu_j)$ 
8:   end for
9:   Update cluster compositions and charges
10:  Step 2: Centroid update
11:  for each cluster  $C_j, j \in \{1, \dots, k\}$  do
12:    Update  $\mu_j$  according to the selected update rule
13:  end for
14:  if convergence is reached or oscillation is detected then
15:    break
16:  end if
17: end for

```

Since initialization is a fundamental issue in clustering algorithms, we tested various **initialization strategies** to study the algorithm’s behavior. Similarly, we experimented with alternative approaches for **updating the centroids**. We describe these approaches in detail below. The complete code with all parameter configurations is available at ¹

5.2.1 Centroid initialization strategies

The k -means algorithm is sensitive to initial centroid selection, often leading to convergence to suboptimal local minima. Since the algorithm relies on random initialization,

¹<https://github.com/glyktln/f3km.git>

different starting points may result in substantially different clustering outcomes.

To systematically evaluate this sensitivity and ensure a robust performance assessment, we consider four distinct initialization strategies. Each configuration is evaluated over multiple independent runs to mitigate randomness effects.

1. **Random Centroids (RC):** Initial centroids are randomly selected from the dataset before the first fairness-aware assignment step. Since each initial centroid corresponds to a single data point, its initial charge is either $+1$ or -1 , depending on the sensitive group of that point.
2. **Random Labels (RL):** Each data point is randomly assigned to one of the k clusters. The centroids are then computed as the mean of the assigned points, resulting in a fully random initial partition of the data. The charge of each initial cluster is computed from the demographic composition of its assigned points.
3. **One Step K-means (KMs):** Randomly selected centroids are refined through a single standard k -means iteration prior to charge-balance computation.
4. **K-means++ initialization (KMPs):** The first centroid is selected uniformly at random. Each subsequent centroid is chosen with probability proportional to the squared distance from the nearest previously selected centroid (k -means++ initialization). A single standard k -means iteration is then performed to determine cluster assignments and centroid charges before applying our method.

5.2.2 Centroid update strategies

We consider two centroid update strategies. The first follows the standard k -means update rule, where each centroid is computed as the mean of the points assigned to its cluster. The second introduces a weighted update scheme.

Standard update of centroids. In the standard update rule, the centroid of cluster C_j is computed as the mean of its assigned points:

$$\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i. \quad (5.8)$$

This formula assigns equal weight to all points in the cluster.

Weighted update of centroids. To incorporate fairness into the centroid update step,

the F3KM algorithm assigns a dynamic weight w_{ij} to each point $\mathbf{x}_i$ assigned to cluster C_j . The weight depends on the current demographic imbalance of the cluster and on whether the point belongs to the over-represented or under-represented group of that cluster.

Using the same fairness correction term as in the assignment score, we define the weight of point $\mathbf{x}_i$ with respect to cluster C_j as

$$w_{ij} = 1 - \lambda q_{\mathbf{x}_i} q_{\mu_j}. \quad (5.9)$$

Here, $q_{\mu_j} = q_j$ is the charge of the cluster represented by centroid μ_j , and $\lambda \in [0, 1]$ controls the strength of the fairness correction. Since $q_{\mathbf{x}_i} \in \{-1, +1\}$ and $q_{\mu_j} \in [-1, 1]$, we have $w_{ij} \in [0, 2]$.

Under this formulation, points from the under-represented group of an imbalanced cluster receive weights greater than 1, increasing their influence on the centroid. In contrast, points from the over-represented group receive weights less than 1, reducing their influence.

The centroid μ_j is then updated as a weighted average:

$$\mu_j = \frac{\sum_{\mathbf{x}_i \in C_j} w_{ij} \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_j} w_{ij}}. \quad (5.10)$$

If $\lambda = 0$, or if the cluster is perfectly balanced so that $q_{\mu_j} = 0$, then all weights become $w_{ij} = 1$, and the update reduces to the standard K -means centroid update.

5.3 Experiments

5.3.1 Evaluation Metrics

To assess the quality of the clustering partition, we use two internal validation metrics:

- the Sum of Squared Errors (SSE)
- the Silhouette Coefficient (SI)

The Sum of Squared Errors[54, 53], often referred to as Inertia, quantifies the compactness of the clusters by measuring the deviation of data points from their respective centroids. Formally, for a set of k clusters, the SSE is calculated as the double summation of the squared Euclidean distances, defined as:

$$\text{SSE} = \sum_{j=1}^k \sum_{x_i \in \mathcal{C}_j} \|x_i - \mu_j\|_2^2 \quad (5.11)$$

where $\mathcal{C}_j$ represents the j -th cluster and μ_j denotes its centroid. The objective is to minimize this value, as a lower SSE indicates that data points are located closer to their cluster centers, implying high internal cohesion. While SSE focuses on compactness, the Silhouette Coefficient [55] evaluates both the cohesion and the separation of the clusters simultaneously. For a given data point x_i , the silhouette value is defined as:

$$\text{SI}(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$$

where $a(x_i)$ is the average distance between x_i and all other points in the same cluster, and $b(x_i)$ is the minimum average distance from x_i to all points in any other cluster. The resulting score ranges from -1 to +1, where a value close to +1 indicates that the instance is well-matched to its own cluster and poorly matched to neighboring clusters, validating that the clusters are distinct and well-separated. The overall Silhouette score for a clustering $\mathcal{C}$ is the mean of the individual Silhouette values:

$$\text{SI} = \frac{1}{n} \sum_{i=1}^n \text{SI}(x_i), \quad (5.12)$$

where n is the total number of points.

To evaluate demographic fairness, we use the *Balance* and *N-Balance* metrics. These correspond to the average bal and average $\overline{\text{bal}}$ scores across all clusters, respectively, as defined in Chapter 3.

5.3.2 Datasets

In our experiments, we use both real and synthetic datasets.

Synthetic Datasets: The synthetic datasets we created, shown in Figure 5.1, are designed to create a controlled conflict between geometric structure and demographic fairness. The points form several monochromatic natural spatial components, where each component mainly contains points from one sensitive group. Components with the same sensitive attribute are placed relatively close to each other, while components from different sensitive groups are placed farther apart. Across all synthetic datasets, the two demographic groups are equally represented, so the normalized and non-normalized balance measures give the same values.

Specifically, the datasets are constructed as follows:

- **Basic dataset** ($n = 800$): The points are arranged in two widely separated horizontal bands of distinct colors, forming four natural spatial components. These components are spatially well separated but demographically homogeneous.
- **Circular datasets**: We evaluate two circular variants. The standard Circular dataset ($n = 3200$) contains four natural spatial components, while the Circular_8 dataset ($n = 800$) contains sixteen natural spatial components. These datasets differ in their spatial dispersion and density, making them useful for testing the behavior of the algorithm under non-linear geometric structure.

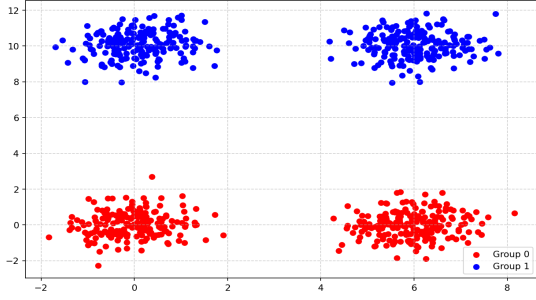
These natural spatial components are not used as the final target clusters. Instead, in our experiments, the target number of clusters is set to half the number of natural components. Therefore, each output cluster is expected to merge, on average, two natural components. This design makes the problem difficult for fair clustering. A standard distance-based method, such as K -means, tends to merge components that are close in the feature space. Since same-color components are often closer to each other, this leads to demographically homogeneous clusters. In contrast, a fairness-aware method must merge components from different demographic groups, even when they are farther apart, in order to improve balance.

Finally, to ensure a broader evaluation, we also use the standard Elliptical dataset ($n = 500$) from the BFKM paper [33].

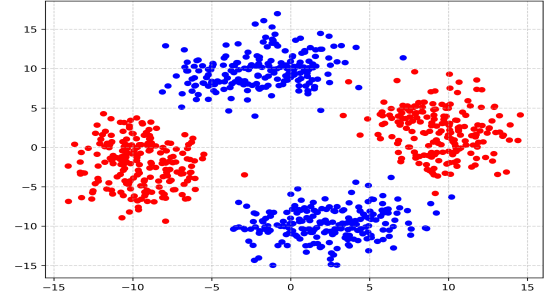
Real Datasets: In addition, for the evaluation of our methods, we employed several real-world datasets, shown in Table 5.1, that are widely used in the fairness and clustering literature. Prior to clustering, a preprocessing pipeline was applied to all datasets. This included data cleaning to remove invalid or missing rows, and the extraction of continuous numerical attributes. The protected attributes were isolated and encoded as binary variables.

For all datasets, each numerical feature is first standardized to have zero mean and unit variance. After standardization, we apply row-wise ℓ_2 normalization before running the clustering algorithms. This preprocessing is used because F3KM is distance-based, and therefore the scale of the input features directly affects the point-centroid distances used in the assignment rule.

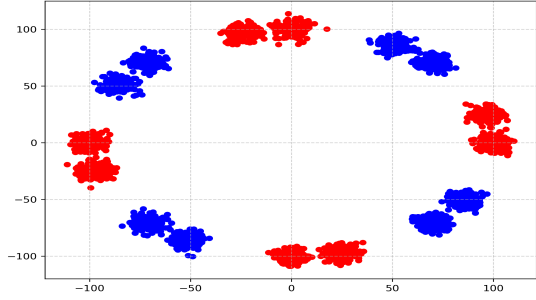
The datasets capture diverse application domains:



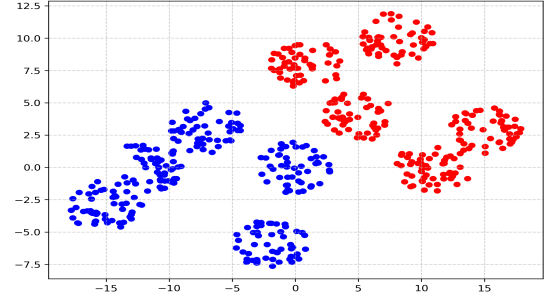
(a) Basic



(b) Circular



(c) Circular 8



(d) Elliptical

Figure 5.1: Synthetic Datasets: (a) basic, (b) circular, (c) circular_8, (d) elliptical

- The **Adult** dataset [56] contains demographic and income information from the U.S. census, utilizing Gender as the protected attribute and 6 numerical features.
- The **Bank Marketing** dataset [57] contains records of direct marketing campaigns from a Portuguese bank, with Marital Status (married vs. unmarried and divorced) as the sensitive attribute and 9 continuous features.
- The **Credit Card** dataset [58] contains financial transactions used to predict default probabilities, utilizing Gender as the protected attribute and 14 numerical features.
- Finally, the **Census 1990** dataset [59] comprises a large-scale socio-economic sample from the U.S. Census Bureau, from which 67 numerical features were extracted (excluding highly identifying attributes), with Gender serving as the protected attribute.

To reduce computational cost, we applied stratified sampling with respect to the protected attribute. This ensures that the sampled datasets preserve the demographic

group proportions of the original datasets. Specifically, exactly 10% of the original data volume was retained for the Adult, Bank, and Credit Card datasets, while a 2% stratified sample was extracted from the massive Census 1990 dataset.

Table 5.1: Summary of Preprocessed and Sampled Real-World Datasets

Dataset	Sampled Instances	Features	Protected Attribute	Ratio (Class 0:Class 1)
Adult	3 256	6	Gender	33.1% : 66.9%
Bank Marketing	4 111	9	Marital Status	39.4% : 60.6%
Credit Card	3 000	14	Gender	60.4% : 39.6%
Census 1990	49 166	67	Gender	48.5% : 51.5%

The average performance metrics across all methods and runs are reported to provide a comprehensive evaluation of clustering stability and quality.

5.3.3 Results

The experimental evaluation is structured as follows. First, we investigate the effect of different initialization strategies on synthetic datasets. Upon identifying the optimal initialization method, we proceed to compare the different centroid update strategies within the same synthetic environment. We then evaluate the effect of the two balance definitions, standard balance and normalized balance on the real datasets. We assess the performance of our algorithm compared to established baseline methods on both synthetic and real datasets. All the datasets are executed for 20 independent runs, with a maximum of 200 iterations per run. We report the average performance across runs. As previously mentioned, the datasets are processed using scaling prior to clustering.

Effect of different initializations In Figure 5.2, we examine the effect of different initialization methods on the synthetic datasets. Since these datasets contain equal numbers of red and blue points, the choice between simple and normalized balance does not affect the results. In all plots, the standard centroid update rule is used. Moreover, all custom synthetic datasets are scaled before applying the algorithm.

In general, the results suggest that the initialization method does not substantially affect the performance of the algorithm across most values of λ . However, Random Label (RL) initialization performs worse on the Circular_8 dataset compared to the other methods, so we do not consider it further as a reliable initialization strategy. For

this reason, we focus mainly on Random Centroids (RC) and K-means++ initialization (KMPs). The former is simple and computationally inexpensive, while the latter provides a more informed initialization by performing an initial standard K-means assignment before applying the fairness-aware updates.

Finally, we observe that for $\lambda \geq 0.9$, all initialization methods tend to produce lower balance values. An explanation is that, for such large values of λ , the fairness term has a very strong effect on the assignment scores of each cluster. Highly imbalanced clusters strongly attract points from the under-represented group and penalize points from their majority group. Since many points may be reassigned at the same time, the composition of the clusters can change abruptly. This can make the clusters change repeatedly across iterations, leading to oscillations instead of a stable balanced solution.

Effect of different centroid updates In Figure 5.3, the initialization method is set to RC technique, because it is simpler and performed similarly to KMPs. As established previously, the choice of balance metric does not impact the outcomes on the synthetic datasets, allowing us to focus exclusively on evaluating the different centroid update strategies. Observing the results across the synthetic datasets, the weighted update method generally outperforms the standard method, achieving its optimal performance at a lower value of λ . Overall, because the weighted update achieves a higher Balance score with a comparable drop in clustering quality, we select it as the optimal update strategy. This empirical finding aligns perfectly with our theoretical expectations: by design, the weighted approach increases the influence of points from the under-represented group of each cluster, shifting the centroid toward a more balanced partition.

Effect of different balance metric in real data To determine the most suitable balance metric for our proposed algorithm F3KM, we evaluated its performance across the real-world datasets 5.4. For this set of experiments, all datasets were standardized, and the algorithm was configured using the RC initialization alongside the weighted centroid update strategy. For the real datasets, we use $K = 10$ for Adult, Bank, and CreditCard, and $K = 20$ for Census1990. The datasets are standardized by removing the mean and scaling to unit variance, followed by row-wise ℓ_2 -normalization when required by the baseline implementation. Observing the results, we note that optimiz-

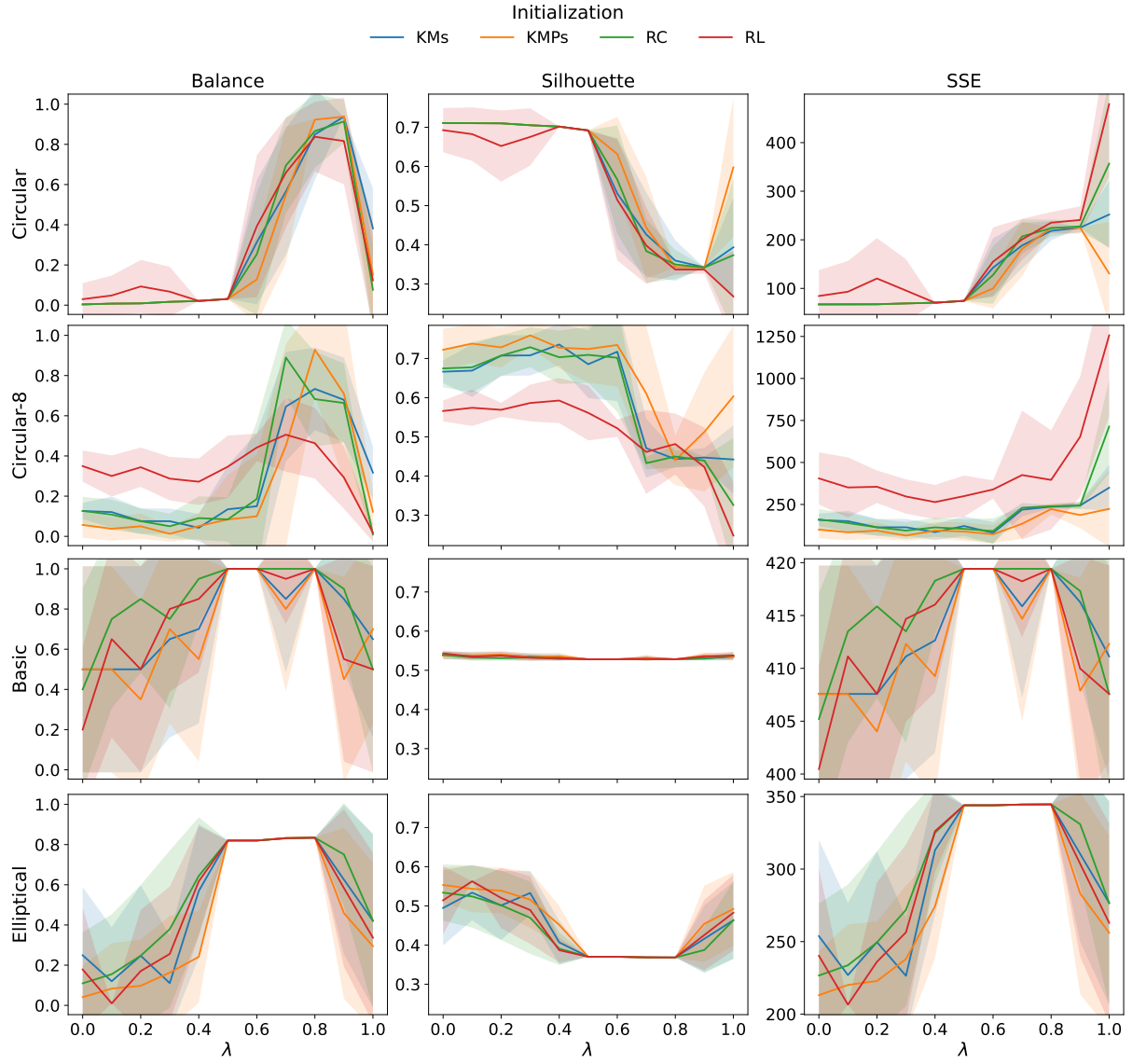


Figure 5.2: Results for synthetic datasets-different initialization

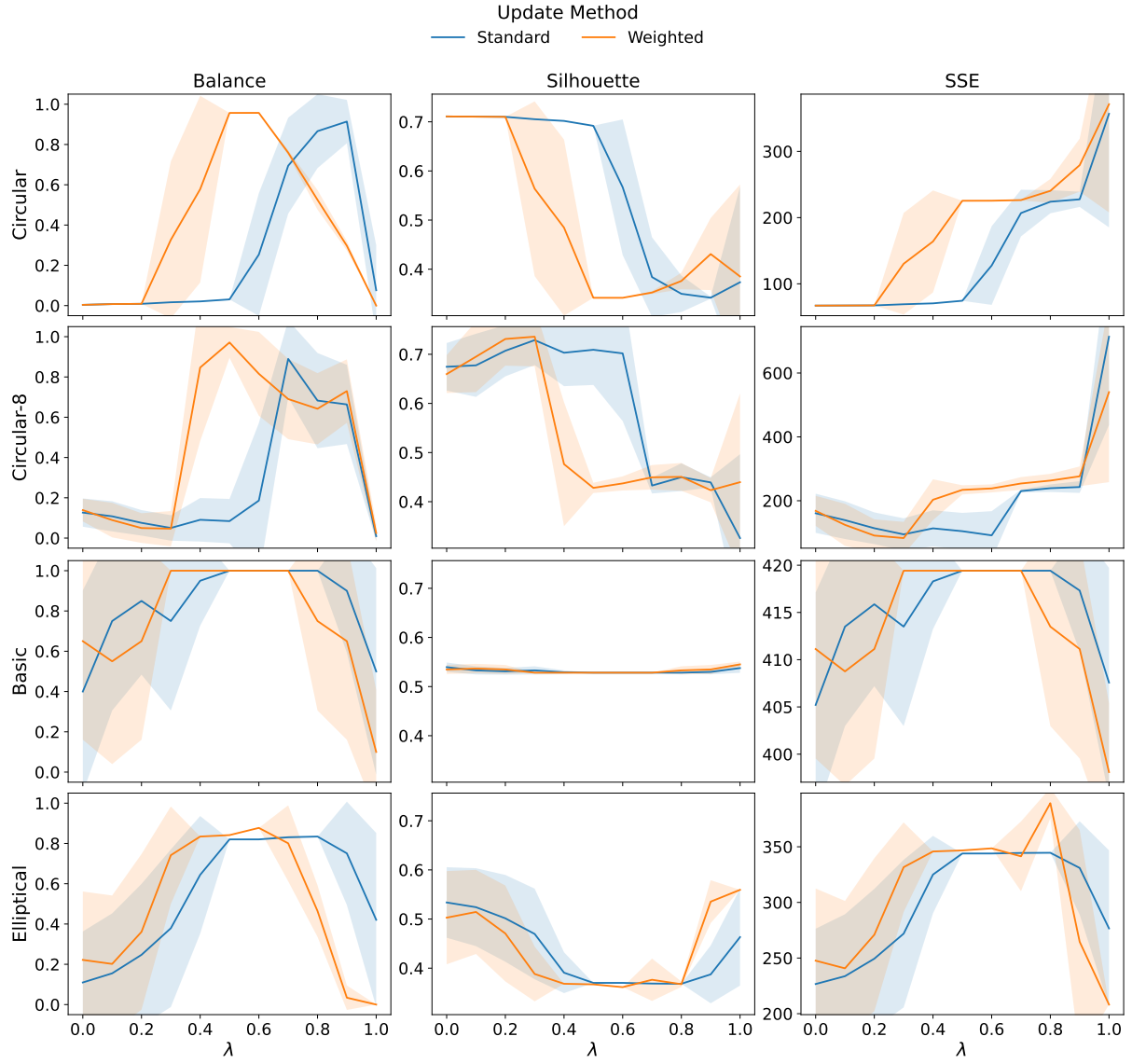


Figure 5.3: Results for synthetic datasets-different updates of centroids

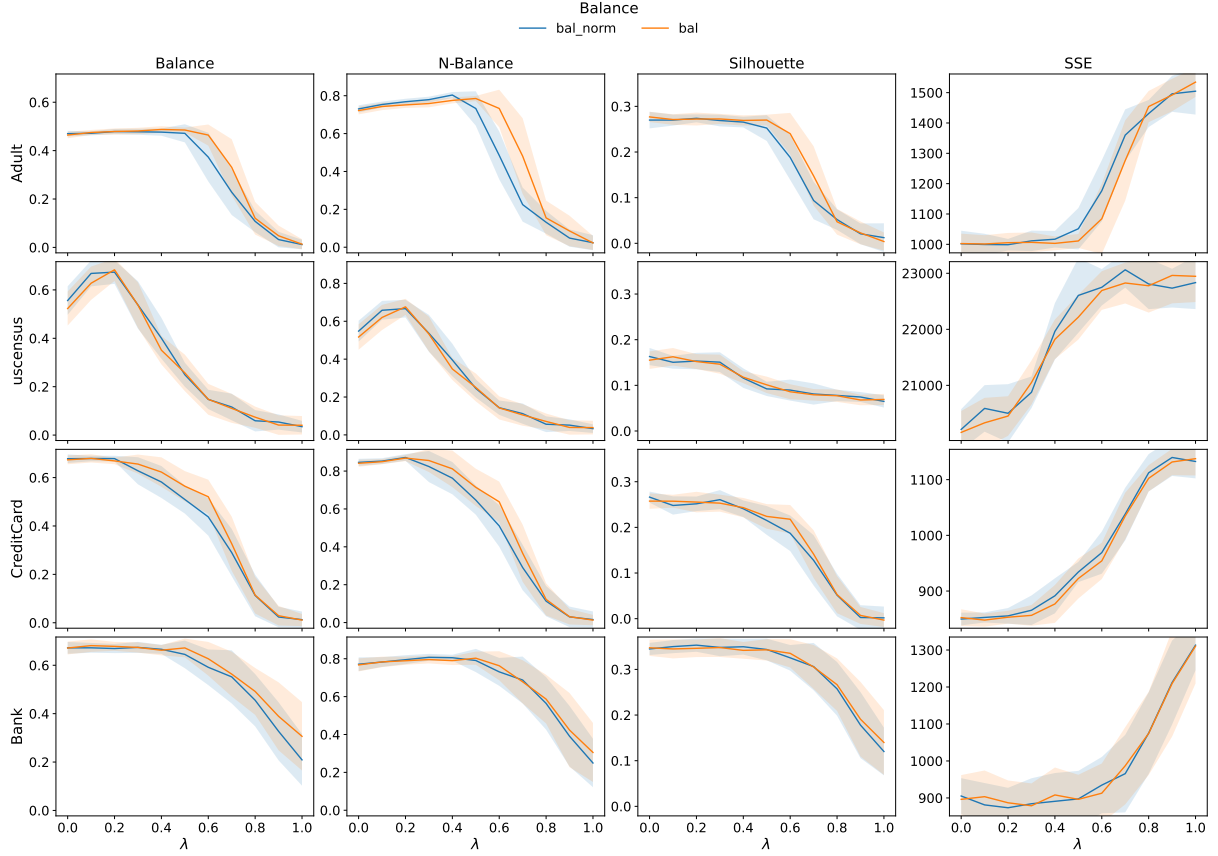


Figure 5.4: Results for real datasets-different notions of balance

ing for either the simple balance or the normalized balance during the algorithm’s execution yields virtually identical clustering outcomes. However, for evaluation purposes, plotting the Normalized Balance *N-Balance* provides a significantly clearer representation of the algorithm’s behavior. Specifically, it allows us to accurately assess how closely the resulting clusters approximate the true demographic fairness of the underlying dataset.

Comparison to Baselines To evaluate the performance of the proposed F3KM algorithm, we compare it against recent fair clustering baselines. In the main quantitative comparison, we include Balanced Fair K-Means (BFKM) [33] and Fair Clustering via Alignment (FCA) [31]. We also tested Variational Fair Clustering (VFC) [32], but we do not include it in the main results table because, under the parameter values used from the original experimental setting, its output was very close to the standard K -means initialization. Since no further parameter search was performed for VFC, we report this only as an implementation observation rather than as a full baseline

comparison.

For a focused comparison, we report only three metrics: N-Balance, SSE, and Silhouette score. N-Balance measures fairness relative to the global demographic distribution, while SSE and Silhouette measure clustering quality. Higher N-Balance values indicate better demographic balance, higher Silhouette values indicate better cluster separation, and lower SSE values indicate better within-cluster compactness. All methods are evaluated using the same metric definitions used throughout this thesis. Because we use our own sampled and preprocessed datasets, the results should be seen as comparisons within this thesis rather than exact reproductions of the original papers.

Experimental setup and parameters. As described above, for the real datasets, we use $K = 10$ for Adult, Bank, and CreditCard, and $K = 20$ for Census1990. The datasets are standardized by removing the mean and scaling to unit variance, followed by row-wise ℓ_2 -normalization when required by the baseline implementation.

For BFKM, we use the following dataset-specific penalty parameters:

Dataset	K	ρ	λ
Adult	10	800000	20000
Bank	10	16000	5000
Census1990	20	24000	10000
CreditCard	10	87000	40000

Here, ρ controls the fairness penalty and λ controls the cluster-size balance penalty in the BFKM objective. Therefore, BFKM is a penalty-based method whose performance depends on dataset-specific parameter choices.

FCA is treated separately because, unlike BFKM, it does not require an additional fairness-penalty parameter to tune. Instead, FCA enforces proportional fairness through its alignment-based formulation. In our experiments, we run FCA with the same K values as the other methods and with the implementation settings `iters` = 50, `numItermax` = 1000000, `epsilon` = 0, `max_iter` = 300, and `iters_inner` = 1. We use the non-gradient version of the method and average the results over 10 random initializations. These are computational settings of the implementation, not additional fairness hyperparameters.

For the proposed F3KM method, we report the best configuration according to maximum N-Balance from the λ -sensitivity experiments. In all reported cases, the

initialization method is the RC and the weighted centroid update is used. In contrast to BFKM, F3KM uses λ to control a local force-based correction in the assignment score, rather than adding a global penalty term to the objective.

VFC implementation note. We also tested the K -means version of VFC using fixed values of its fairness parameter λ . Where available, these values were based on the experimental settings reported in the original VFC paper. However, under our sampled and preprocessed datasets, the resulting assignments were very close to the initial standard K -means solution. In particular, on Census1990, even using a large value of λ , VFC produced almost identical SSE, Silhouette, and N-Balance values to its K -means initialization. Since VFC is known to be sensitive to the choice of λ , and since we did not perform an additional grid search over this parameter, we do not include VFC in the main quantitative comparison. Instead, we focus the baseline comparison on BFKM and FCA, for which the experimental setup was more directly aligned with our evaluation.

Table 5.2 shows that the methods follow different fairness-quality behaviours. FCA achieves the highest N-Balance on all real datasets, reaching almost perfect demographic balance. This is expected, since FCA is designed to enforce proportional fairness through an alignment-based formulation. However, this strong fairness improvement comes with a clear cost in clustering quality, especially in terms of Silhouette score. This effect is particularly visible on Census1990, where FCA achieves an N-Balance of 0.997, but gives the lowest Silhouette score.

BFKM also achieves very high N-Balance on Adult and CreditCard, but with a noticeable decrease in Silhouette. This suggests that BFKM enforces fairness aggressively through its penalty terms, but may distort the spatial structure of the clusters. On Bank and Census1990, BFKM improves N-Balance more moderately.

The proposed F3KM method provides a moderate trade-off between these behaviours. It does not force demographic balance as strongly as FCA or BFKM, but it consistently improves N-Balance over standard K -means on the reported datasets. At the same time, its Silhouette score remains close to the original K -means solution. For example, on Adult, F3KM increases N-Balance from 0.733 to 0.804, while maintaining a Silhouette score of 0.266. On Bank, it improves N-Balance from 0.784 to 0.808, with only a small reduction in Silhouette compared with K -means. On CreditCard, it improves N-Balance from 0.839 to 0.872, while keeping the Silhouette score close to the baseline. On Census1990, F3KM improves N-Balance from 0.488 to 0.667, providing

Table 5.2: Comparison of fair clustering methods on real datasets

Dataset	Method	N-Balance	SSE	Silhouette
Adult	K -means	0.733	986.39	0.276
	F3KM ($\lambda = 0.4$)	0.804	1016.81	0.266
	FCA	0.992	1066.88	0.244
	BFKM	0.996	1131.95	0.210
Bank	K -means	0.784	843.46	0.354
	F3KM ($\lambda = 0.3$)	0.808	884.29	0.348
	BFKM	0.841	851.29	0.343
	FCA	0.996	1000.57	0.307
CreditCard	K -means	0.839	847.87	0.263
	F3KM ($\lambda = 0.2$)	0.872	855.41	0.252
	BFKM	0.985	878.82	0.175
	FCA	0.992	860.86	0.254
Census1990	K -means	0.488	19661.74	0.166
	F3KM ($\lambda = 0.2$)	0.667	20501.36	0.153
	BFKM	0.565	19810.81	0.162
	FCA	0.997	22522.39	0.117

an intermediate solution between standard K -means and the more aggressive FCA baseline.

Overall, the results show that FCA and BFKM are stronger when the main objective is to maximize demographic balance. However, this often comes with a stronger loss in clustering quality. In contrast, F3KM acts as a softer, local assignment-level correction: it improves demographic balance while preserving more of the distance-based clustering structure.

Synthetic datasets.

Table 5.3 compares F3KM, BFKM, and FCA on the synthetic datasets using Balance, Silhouette, and SSE. For F3KM, we report the smallest value of λ among the tested configurations that gives the selected best result.

FCA achieves perfect Balance on all synthetic datasets. This is expected, since its alignment-based formulation is designed to enforce proportional fairness. However, this improvement in fairness is accompanied by a clear cost in clustering quality, especially on the non-convex datasets. In particular, on the Circular and Circular-8 datasets, FCA achieves perfect Balance, but with higher SSE and lower Silhouette compared to the geometry-preserving BFKM solutions.

For BFKM, we did not perform a systematic parameter search on the synthetic datasets. We tested only a limited set of parameter configurations. For the Elliptical dataset, we used the parameter setting considered in the original BFKM paper, while for the remaining synthetic datasets we used the fixed setting reported in Table 5.3. Under these tested settings, BFKM achieves perfect Balance on the Basic dataset and strong Balance on the Elliptical dataset. However, it performs poorly in terms of Balance on the Circular and Circular-8 datasets, despite achieving better SSE and Silhouette values. This suggests that, under these parameter choices, BFKM preserves the geometric structure well, but does not sufficiently improve demographic balance on the non-convex datasets.

As we see before, F3KM achieves high Balance across all synthetic datasets. On the Circular and Circular-8 datasets, it substantially improves fairness compared to BFKM, while obtaining Silhouette and SSE values close to FCA. On the Elliptical dataset, F3KM slightly improves both Balance and clustering quality compared to BFKM and FCA. Overall, these results illustrate the fairness–quality trade-off. FCA enforces perfect fairness, BFKM can perform well under suitable parameter settings, and F3KM provides a more consistent force-based fairness improvement across the

Table 5.3: Comparison of F3KM, BFKM, and FCA on synthetic datasets. For F3KM, we report the smallest value of λ among the tested configurations that gives the selected best result.

Dataset	Method	Parameters	Balance $\uparrow$	Silhouette $\uparrow$	SSE $\downarrow$
Basic	F3KM	$\lambda = 0.3, k = 2$	1.000	0.528	419.419
	BFKM	$p_1 = 200, p_2 = 100, k = 2$	1.000	0.528	419.419
	FCA	$k = 2$	1.000	0.528	419.419
Circular	F3KM	$\lambda = 0.6, k = 4$	0.957	0.342	225.527
	BFKM	$p_1 = 200, p_2 = 100, k = 4$	0.009	0.710	67.355
	FCA	$k = 4$	1.000	0.340	226.048
Circular-8	F3KM	$\lambda = 0.5, k = 8$	0.971	0.428	234.023
	BFKM	$p_1 = 200, p_2 = 100, k = 8$	0.158	0.649	184.367
	FCA	$k = 8$	1.000	0.425	229.357
Elliptical	F3KM	$\lambda = 0.6, k = 2$	0.877	0.361	348.587
	BFKM	$p_1 = 470, p_2 = 100000, k = 2$	0.858	0.344	352.004
	FCA	$k = 2$	1.000	0.340	353.829

synthetic datasets.

5.4 Discussion

The experimental results show that the proposed F3KM algorithm provides a moderate fairness-quality trade-off. The method is able to improve demographic balance compared with standard K -means, especially when the initial clustering is demographically imbalanced. However, the improvement in fairness is not uniform across all datasets and depends strongly on the structure of the data, the initial configuration, and the value of the fairness parameter λ .

The difference between the synthetic and real datasets is particularly important. On synthetic datasets, the demographic structure is controlled and often strongly separated, so the effect of the fairness force is more visible. In contrast, on several real datasets, standard K -means already produces relatively balanced partitions. In these cases, the possible fairness improvement is naturally limited, because the initial clustering is not highly unfair. Therefore, F3KM should not be interpreted as a method that always forces perfect demographic balance, but rather as a local correction mechanism that can improve balance when the assignment structure allows it.

Limitations. In this paragraph, we discuss the main limitations of the current formulation and highlight specific cases in which the fairness force may lead to unstable or less fair assignments.

A key limitation of the current formulation is that point assignments are updated simultaneously. When λ is large, many points from the same demographic group may move toward the same set of clusters at the same time. Since each point makes its assignment based on the current cluster composition, it does not account for the fact that many other points may make a similar move in the same iteration. This can lead to abrupt changes in cluster composition, uneven final assignments, and oscillatory behavior.

Another limitation is that the distance term has a strong effect on the assignments. Because the score uses the inverse squared distance, nearby centroids can dominate the assignment decision. In such cases, the fairness term may not be strong enough to change the assignment, especially when a point is much closer to one centroid than to the others.

In the extreme case where $\lambda = 1$, a fully monochromatic cluster gives zero attraction to points from its majority group and strongly favors points from the opposite group. In theory, this behavior can create oscillations, since clusters may attract points from the opposite group, change their demographic composition, and then reverse the direction of the fairness force. However, in practice, the balance does not necessarily remain equal to zero, because several effects can interrupt this idealized behavior.

Empty clusters. Strong fairness corrections may leave some clusters empty. When this happens, the cluster has no majority demographic direction, and the fairness correction is effectively deactivated. The cluster then behaves like a standard K -means cluster and can attract nearby points according to distance.

Initialization with centroids from the same group. If several initial centroids belong to the same demographic group, their initial charges are the same. In this case, the fairness term affects all these centroids in the same way for a given point. In the extreme case $\lambda = 1$, points from the same group as the centroids receive zero attraction toward all these clusters. Since their assignment scores are then equal, ties are resolved by random assignment among the tied clusters in our implementation. Points from the opposite group receive the same fairness amplification toward all these clusters, and are therefore also assigned mainly according to distance.

These effects are stronger in the two-cluster case ($k = 2$), because changes in one

cluster directly affect the balance and demographic direction of the other. As a result, even small reassignment changes can reverse the majority direction of the clusters and lead to oscillatory behavior.

Another common case occurs with random-label initialization. If the initial random partition already produces balanced or nearly balanced clusters, the fairness force is weak or zero from the beginning, and the assignments are driven mainly by the distance-based term.

Future Work While the proposed force-based approach introduces a promising direction for fairness-aware clustering, there are several avenues for future research that can further enhance its robustness and applicability.

First, a critical direction for future research involves developing a systematic methodology to select the optimal λ . As highlighted in our experimental evaluation, the choice of λ strictly dictates the trade-off between demographic parity and clustering quality, and its optimal value is not immediately obvious without exhaustive empirical tuning. Future research will focus on data-driven techniques to automatically determine optimal λ , adapting to the specific characteristics and fairness needs of each dataset.

Furthermore, an important direction for future research is extending our force-based framework to accommodate multiple sensitive attributes. To achieve this, we intend to adapt established multi-class reduction strategies, such as the One-vs-All approach. This would allow our algorithm to simultaneously balance several protected groups within the same clustering process.

In addition, to rigorously evaluate the computational efficiency and scalability of our algorithm, we intend to apply it to a more diverse and complex set of real-world datasets.

Also, a highly promising direction is the introduction of a dynamic "fairness deficit" metric. This metric would explicitly quantify the exact number of data points a cluster needs to acquire or release to achieve demographic parity. Based on this metric, we could implement a controlled assignment mechanism, restricting the percentage of points allowed to change clusters during a single iteration. This thresholding approach aims to mitigate severe centroid oscillations and guarantee smoother and more stable convergence. Another promising direction is to redefine the point charge as a function of its demographic group's global frequency, thereby assigning adaptive weights based

on group size.

CHAPTER 6

FAIR FORCE-FIELD CLUSTERING

6.1 Force-Field Clustering (F2C)

6.2 Fair Force-Field Clustering (F3C)

6.3 Experiments

6.4 Discussion

In this chapter, we study another application of the physics-inspired framework introduced in Chapter 3. In contrast to F3KM, where points are assigned to centroids, the method considered here follows a dynamic clustering process. Data points are treated as active particles that move in the feature space under the influence of Force-Fields. Clusters are formed gradually, as particles move toward each other and nearby particles are merged.

We first describe the base Force-Field Clustering method, denoted by F2C. This method uses only gravitational-like interactions between active particles and does not take sensitive attributes into account. It therefore serves as the underlying clustering mechanism on which the fairness-aware version is built.

We then extend this base method using the fairness-charge framework introduced in Chapter 3. The resulting algorithm, Fair Force-Field Clustering (F3C), incorporates demographic information through fairness forces between active groups of points. In this way, particle movement is influenced not only by spatial proximity, but also by the demographic imbalance of the groups represented by the particles.

6.1 Force-Field Clustering (F2C)

The base algorithm starts by treating each data point as an active particle. Let $z_i^{(t)} \in \mathbb{R}^d$ denote the position of active particle i at iteration t . Initially, each particle is located at the position of one original data point:

$$z_i^{(0)} = x_i.$$

Each initial particle has mass $m_i = 1$. When particles are merged, the resulting particle represents all original points contained in the merged group, and its mass is updated as the sum of the merged masses.

At any iteration, an active particle may therefore represent either a single original point or a group of points that have already been merged. The final clusters are obtained from the groups represented by the remaining active particles.

6.1.1 Gravitational interaction

The movement of particles in F2C is driven by gravitational-like attraction between pairs of active particles. Let z_i and z_j denote the current positions of two active particles. We define

$$r_{ij} = z_j - z_i$$

as the displacement vector from particle i to particle j , and

$$d_{ij} = \|r_{ij}\|$$

as their Euclidean distance.

The gravitational force exerted on particle i by particle j is

$$\mathbf{F}_{ij}^g = G \frac{m_i m_j}{d_{ij}^p} r_{ij}.$$

Here, G controls the strength of the gravitational attraction, m_i and m_j are the masses of the two particles, and p is a distance exponent. In our implementation, we set $G = 1$, and p is set to the dimensionality of the data.

Nearby particles and particles with larger mass exert stronger attraction. The total gravitational force acting on particle i is obtained by summing the pairwise forces from all other active particles:

$$\mathbf{F}_i^g = \sum_{j \neq i} \mathbf{F}_{ij}^g.$$

Optionally, the interaction can be restricted to particles within a fixed radius h . In that case, the sum is taken only over the neighboring particles of i inside this radius.

6.1.2 Particle movement

After the total force acting on each particle has been computed, the particle positions are updated. The movement rule is inspired by the physical notion of work. In physics, the work performed by a force during a displacement is given by the inner product between the force and the displacement. In our algorithm, this idea is used as an algorithmic analogy: each active particle is assigned the same fixed work budget during one movement step.

Let $W > 0$ denote this work budget. In our experiments, we set $W = 0.5$. For a particle i with total force vector $\mathbf{F}_i$, the unconstrained displacement is defined as

$$\Delta z_i^{work} = W \frac{\mathbf{F}_i}{\|\mathbf{F}_i\|^2}.$$

This displacement is in the direction of the total force.

where W controls the scale of the movement. If the force magnitude is below a small numerical threshold, the displacement is set to zero. The new position of the particle is then

$$z_i^{(t+1)} = z_i^{(t)} + \Delta z_i.$$

In the implementation, the displacement is also bounded using the nearest active neighbor of each particle. More specifically, if d_i^{near} denotes the distance from particle i to its nearest active neighbor, then the displacement is clipped so that

$$\|\Delta z_i\| \leq 0.5 d_i^{near}.$$

This keeps the movement local with respect to the current particle configuration and prevents excessively large updates.

6.1.3 Merging and convergence

After each movement step, nearby particles are merged. If the distance between two active particles is below a merging threshold τ , they are assigned to the same merged group. For a merged group C , the new particle position is computed as the mass-weighted average

$$z_C = \frac{\sum_{i \in C} m_i z_i}{\sum_{i \in C} m_i}.$$

The mass of the merged particle is

$$m_C = \sum_{i \in C} m_i.$$

In our implementation, the merging threshold is selected according to the experimental setting. In the synthetic datasets, the merging threshold is set equal to the minimum step threshold used by the algorithm. In the real datasets, where the number of points and the dimensionality make full pairwise interactions computationally expensive, the merging threshold is derived from the interaction radius h . The exact parameter settings used in each case are described in Section 6.3.3.

The process is repeated until one of the stopping criteria is satisfied. The algorithm stops when the desired number of clusters k_{goal} is reached, when only one active particle remains, when the active particle positions no longer change after an iteration, or when the maximum number of iterations is reached.

6.2 Fair Force-Field Clustering (F3C)

The base F2C method described above uses only geometric information. We now extend it by incorporating the fairness-charge mechanism introduced in Chapter 3. The resulting method is denoted by F3C.

At each iteration, every active particle represents a group of original points. Let C_i denote the group represented by active particle i . Instead of redefining the fairness quantities, we use the balance and imbalance notions introduced in Chapter 3. The fairness charge of an active group is determined by its current demographic composition: its sign indicates the over-represented sensitive group, while its magnitude is proportional to the imbalance of the group, so $q_i = +/ - \text{imb}(C_i)$.

Thus, balanced active groups have zero fairness charge, while imbalanced groups exert a stronger fairness influence. Since active particles are merged during the algorithm, these charges are recomputed after each merging step.

6.2.1 Fairness interaction

The fairness-aware extension adds an electrostatic-like interaction between active groups. For two active particles i and j , with charges q_i and q_j , the fairness force acting on particle i due to particle j is

$$\mathbf{F}_{ij}^e = -\frac{q_i q_j}{d_{ij}^p} r_{ij}.$$

following the fairness definitions in Chapter 3

With this sign convention, active groups dominated by different sensitive groups are attracted to each other, while groups dominated by the same sensitive group repel each other. Therefore, the fairness term encourages movements that bring demographically complementary groups closer to each other and discourages movements between groups that would reinforce demographic imbalance. These movements can then affect which particles become close enough to be merged.

The total fairness force acting on particle i is

$$\mathbf{F}_i^e = \sum_{j \neq i} \mathbf{F}_{ij}^e.$$

6.2.2 Combined force

The final force used by F3C combines the gravitational force of the base F2C algorithm with the fairness force:

$$\mathbf{F}_i = (1 - \lambda)\mathbf{F}_i^g + \lambda\mathbf{F}_i^e.$$

The parameter $\lambda \in [0, 1]$ controls the trade-off between geometric clustering and fairness. When $\lambda = 0$, the fairness term is inactive and F3C reduces to the base F2C algorithm. For $\lambda > 0$, the movement of active particles is influenced by both spatial proximity and demographic imbalance.

For large values of λ , the fairness component may dominate the geometric attraction. In particular, active groups with the same dominant sensitive group can exert repulsive forces on each other. This can push particles away from nearby geometric neighbors and may therefore affect the spatial compactness of the final clusters. This behavior reflects the fairness-quality trade-off of the method: stronger fairness forces can improve demographic balance, but they may also distort the geometric structure of the clustering.

6.2.3 Algorithm

The resulting F3C algorithm follows the same movement-and-merging structure as F2C, with the additional step of computing fairness charges for the active groups. The charges are updated throughout the process, because each active particle may represent a different group of original points after merging.

Algorithm 6.1 Fair Force-Field Clustering F3C

```
1: Input: Dataset  $X$ , sensitive attributes  $A$ , target number of clusters  $k_{\text{goal}}$ , maximum
   iterations  $T_{\text{max}}$ , parameters  $G, K, W, \lambda, \tau$ 
2: Output: Final clusters  $\mathcal{C}$ 
3: Initialize each point as an active particle with mass 1
4: for  $t = 1$  to  $T_{\text{max}}$  do
5:   Compute the fairness charges of the active groups
6:   Compute total forces  $\mathbf{F}_i = (1 - \lambda)\mathbf{F}_i^g + \lambda\mathbf{F}_i^e$ 
7:   Move particles according to the displacement rule
8:   Merge nearby particles and update their masses
9:   if  $k_{\text{goal}}$  is reached, only one particle remains, or convergence is detected then
10:    break
11:   end if
12: end for
13: return The clusters induced by the final active particles
```

6.3 Experiments

In this section, we evaluate the proposed force-based clustering approach. The experimental evaluation is organized in two stages. First, we study the behavior of the base F2C algorithm. This allows us to examine whether the movement and merging mechanism can produce meaningful clusters before fairness is introduced.

Second, we evaluate the fairness-aware behavior of F3C by varying λ , using $\lambda = 0$ as the internal baseline and values $\lambda > 0$ to activate the fairness interaction. In this setting, the movement of the active particles is affected not only by spatial proximity, but also by the demographic composition of the active groups.

6.3.1 Evaluation Metrics

We use the same evaluation metrics as in Chapter 5. For clustering quality, we report the Silhouette Score and the Sum of Squared Errors (SSE). Higher Silhouette values indicate better separated clusters, while lower SSE values indicate more compact clusters.

For fairness, we report the average standard *Balance* defined in Chapter 3. This metric measures how well the sensitive groups are mixed inside the produced clusters.

When reference cluster labels are available, we also report the Adjusted Rand Index (ARI) [60] and Normalized Mutual Information (NMI) [61]. These metrics compare the produced clustering with the known reference partition and are used only as external clustering-quality measures, not as fairness measures.

Let $\mathcal{C} = \{C_1, \dots, C_r\}$ be the clustering produced by an algorithm, and let $\mathcal{T} = \{T_1, \dots, T_s\}$ be the reference partition. Let $n_{ij} = |C_i \cap T_j|$ denote the number of points that belong both to cluster C_i and reference class T_j . Also, let

$$a_i = \sum_{j=1}^s n_{ij}, \quad b_j = \sum_{i=1}^r n_{ij}.$$

The Adjusted Rand Index (ARI) is defined as

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}.$$

Let $\mathcal{C}$ be the clustering produced by an algorithm and let $\mathcal{T}$ be the reference partition. The mutual information between $\mathcal{C}$ and $\mathcal{T}$ is defined as

$$I(\mathcal{C}; \mathcal{T}) = \sum_{i=1}^r \sum_{j=1}^s \frac{n_{ij}}{n} \log \left(\frac{n n_{ij}}{a_i b_j} \right),$$

where $n_{ij} = |C_i \cap T_j|$, $a_i = \sum_j n_{ij}$, $b_j = \sum_i n_{ij}$, and n is the total number of points.

The entropy of a partition $\mathcal{C}$ is

$$H(\mathcal{C}) = - \sum_{i=1}^r \frac{a_i}{n} \log \left(\frac{a_i}{n} \right),$$

and similarly for $H(\mathcal{T})$.

The Normalized Mutual Information (NMI) is then defined as

$$NMI(\mathcal{C}, \mathcal{T}) = \frac{2I(\mathcal{C}; \mathcal{T})}{H(\mathcal{C}) + H(\mathcal{T})}.$$

The NMI score lies in $[0, 1]$, where 1 indicates perfect agreement between the clustering and the reference partition.

For real datasets, where the sensitive groups are not necessarily equally represented, we also report normalized balance (NBalance). NBalance compares the red-to-blue ratio inside each cluster with the global red-to-blue ratio of the full dataset. This allows us to evaluate whether clusters preserve the global demographic proportions, instead of requiring an equal 50-50 composition in every cluster.

6.3.2 Datasets

For the experiments of this chapter, we use datasets taken from the public *Clustering-Datasets* repository,¹ which contains both synthetic and real-world datasets for clustering evaluation. In the first stage, we focus on selected synthetic datasets, since they provide known cluster labels and allow us to study whether the base F2C method can recover the expected cluster structure. Some representative datasets used in this stage are shown in Figure 6.1.

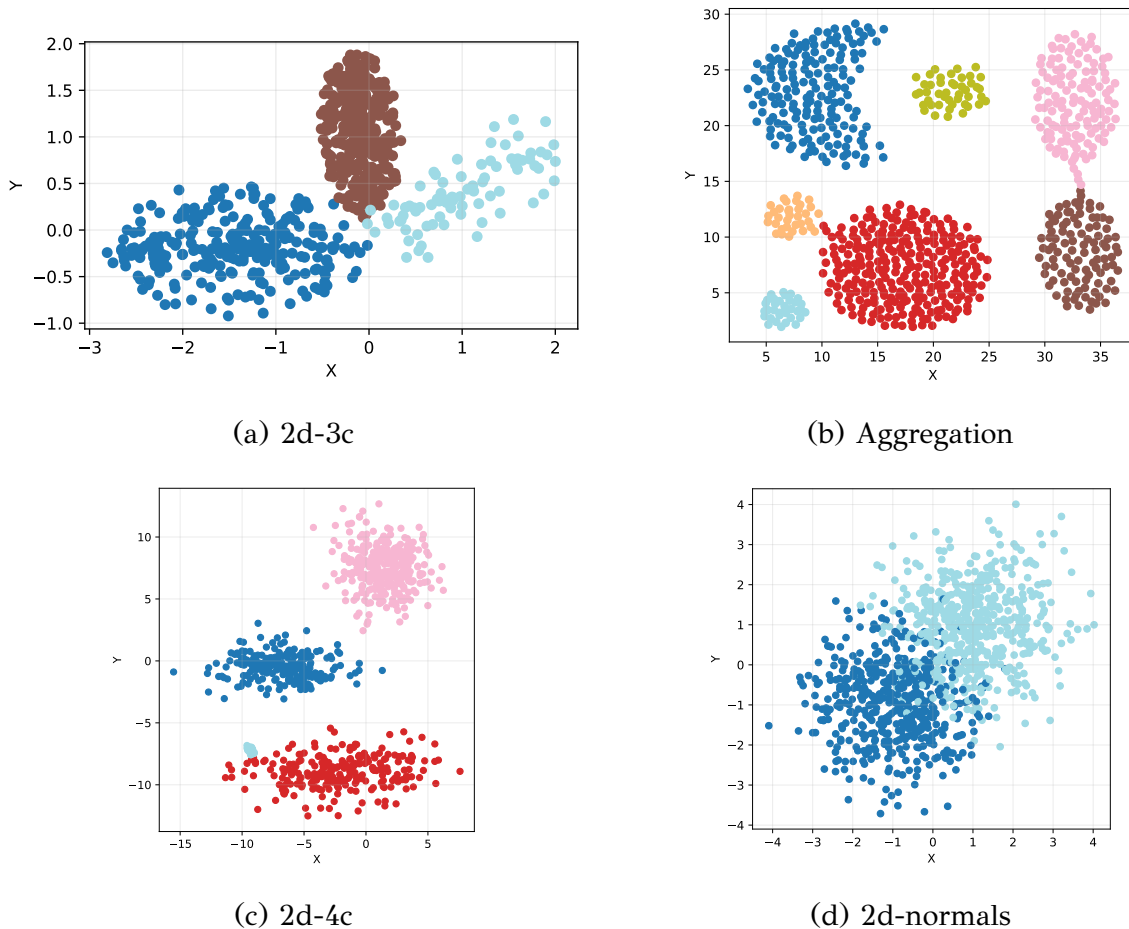


Figure 6.1: Representative synthetic datasets used in the experimental evaluation of F2C and F3C.

After studying the behavior of the method on these datasets, we also use the synthetic and real datasets introduced in Chapter 5. This keeps the experimental setting consistent with the evaluation of F3KM.

¹<https://github.com/milaan9/Clustering-Datasets>

6.3.3 Experimental Setup

For the base clustering experiments, we evaluate F2C, which corresponds to the case where no fairness force is used. We compare F2C with MeanShift [62], since both methods are based on the movement of points in the feature space and do not rely on point-to-centroid assignments in the same way as K-means. This comparison serves as an initial baseline for evaluating the behavior of the force-based clustering mechanism.

MeanShift is a density-based clustering method that iteratively moves each point toward the mean of the points in its local neighborhood. Given a bandwidth parameter h , the mean-shift update is

$$x^{(t+1)} = \frac{\sum_{i=1}^n x_i K\left(\frac{x^{(t)} - x_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x^{(t)} - x_i}{h}\right)}.$$

Points that converge to the same density mode are assigned to the same cluster.

For the fairness-aware experiments, we evaluate F3C mainly through an internal ablation study by varying the parameter λ . The case $\lambda = 0$ corresponds to the base F2C method, while values $\lambda > 0$ activate the fairness interaction between active groups. Thus, the experiments show how the proposed fairness term affects the behavior of the force-based clustering process. A full comparison of F3C against other fairness-aware clustering algorithms is left for future work.

For synthetic datasets, we report Silhouette Score, SSE, ARI, and NMI, since the reference clustering labels are known. For fairness-aware experiments, we additionally report standard Balance. For real datasets, where reference cluster labels may not represent a unique ground-truth clustering, we focus mainly on Balance, Silhouette Score, and SSE.

We use different parameter settings for synthetic and real datasets. For the synthetic datasets, we do not restrict the particle interactions using a local radius h . Therefore, at each iteration, every active particle can interact with all other active particles. In these experiments, the merging threshold is set using `threshold_merge_mode = min_step`, meaning that two active particles are merged only when their distance becomes smaller than the minimum distance of every point of dataset. Since the reference number of clusters is known for the synthetic datasets, we also use `kgoal` as a stopping criterion. Therefore the algorithm can also terminate when the number of active particles reaches the target number of clusters.

For the real datasets, running all pairwise interactions is computationally expensive because of the larger number of data points and the higher dimensionality of the feature space. For this reason, we restrict the interactions using a local radius h . The radius h is estimated using the maximum-distance bandwidth heuristic with quantile 0.3. For the merging step, we use this distance multiplied by 0.2.

6.3.4 F2C Results

We first evaluate the base F2C algorithm. In this case, the fairness force is inactive and the algorithm uses only the gravitational interaction between active particles. This stage isolates the behavior of the underlying force-based clustering mechanism.

Table 6.1 reports the comparison between F2C and MeanShift on the selected synthetic datasets.

Table 6.1: External clustering agreement for F2C and MeanShift on synthetic datasets.

Dataset	Method	ARI	NMI
2d-3c	MeanShift	0.757	0.795
	F2C	0.394	0.502
2d-4c	MeanShift	0.797	0.830
	F2C	0.900	0.888
2d-normals	MeanShift	0.000	0.000
	F2C	0.461	0.441
Aggregation	MeanShift	0.372	0.542
	F2C	0.370	0.489
Cassini	MeanShift	0.640	0.636
	F2C	0.739	0.724

Table 6.1 compares the base F2C method with MeanShift using ARI and NMI on synthetic datasets with known reference labels. The results show that the behavior of F2C depends on the geometry of the dataset. On 2d-4c and Cassini, F2C achieves higher agreement with the reference clustering than MeanShift. On 2d-3c, MeanShift performs better, while on Aggregation the two methods give similar ARI values, with MeanShift achieving a higher NMI.

For the 2d-normals dataset, MeanShift obtains ARI and NMI equal to zero. This corresponds to a degenerate solution where the method returns a single cluster, and therefore it does not recover the reference partition. In contrast, F2C produces a non-trivial clustering and obtains higher agreement with the reference labels.

Overall, these results indicate that F2C has moderate performance as a pure clustering algorithm. It can recover a meaningful cluster structure in some datasets, especially when the force-based movement and merging process is compatible with the geometry of the data. However, it is not consistently better than MeanShift. This suggests that there is room for further investigation and improvement. One of the benefits of our method is that it provides a dynamic force-based mechanism that can be extended with fairness forces.

6.3.5 Effect of the Fairness Term

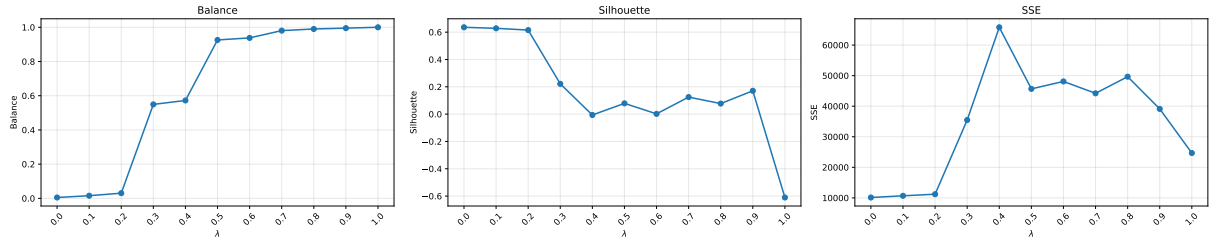
We now evaluate the fairness-aware behavior of F3C by varying λ . This experiment should be interpreted as an ablation study of the proposed fairness term, rather than as a full comparison against existing fairness-aware clustering algorithms. For $\lambda = 0$, the method is equivalent to the base F2C algorithm. For $\lambda > 0$, the fairness interaction is activated and the movement of particles is also influenced by demographic imbalance. In this way, we isolate the effect of the fairness force on Balance, *NBalance*, Silhouette, SSE, and the number of produced clusters.

F3C Results on Synthetic Datasets First we evaluate the fair synthetic datasets described in Section 5.3.2.

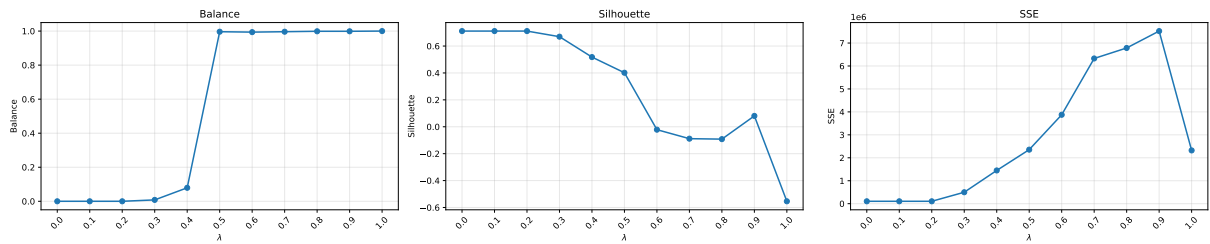
Figure 6.2 shows the effect of λ on the fairness and quality metrics for the synthetic datasets. As λ increases, Balance generally improves and, for most datasets, reaches the maximum value of 1 for $\lambda \geq 0.5$, remaining almost stable afterwards. This indicates that the fairness term becomes strong enough to produce perfectly balanced clusters.

At the same time, the improvement in fairness comes with the expected quality trade-off. The Silhouette score tends to decrease gradually, while SSE generally increases, showing that the clusters become less compact and less clearly separated as more weight is assigned to fairness.

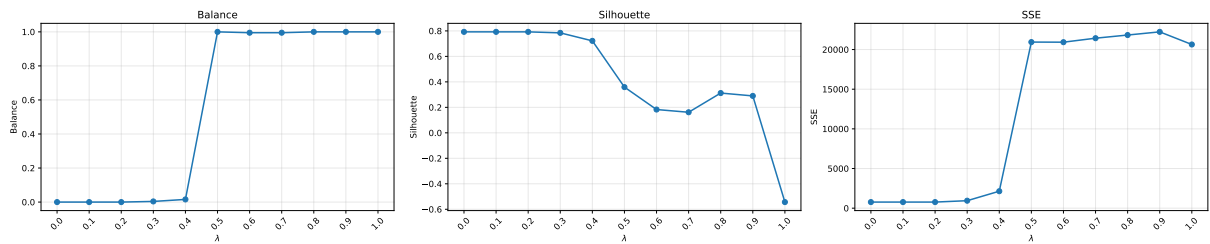
The Basic and Circular-8 datasets show similar behavior: before $\lambda = 0.5$, Balance



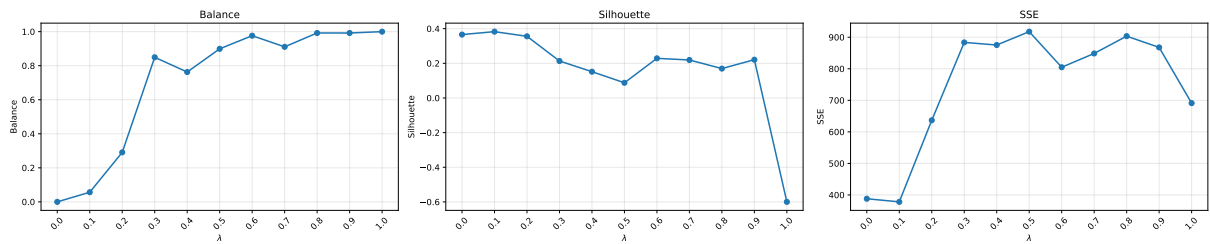
(a) Circular



(b) Circular-8



(c) Basic



(d) Elliptical

Figure 6.2: Effect of λ on Balance, Silhouette, and SSE for the synthetic datasets.

remains close to zero, while for $\lambda \geq 0.5$ it sharply increases to 1 and remains stable. Overall, the results are positive, since F3C substantially improves demographic balance while making the fairness-quality trade-off explicit.

For $\lambda = 0.5$, the first three synthetic datasets, Basic, Circular, and Circular-8, already achieve high Balance, while the decrease in clustering quality remains relatively small.

F3C Results on Real Datasets For the real datasets, we use the same preprocessed datasets described in Chapter 5. Since F2C and F3C are based directly on Euclidean distances between active particles, the real datasets are standardized before running the algorithm. This preprocessing follows the same methodology used in Chapter 5, since the real datasets contain features with different scales.

Running the force-based method on real datasets is more computationally demanding than running it on low-dimensional synthetic data. For this reason, we use a local interaction radius h in these experiments. Specifically, h is computed using the maximum-distance nearest-neighbor heuristic with quantile 0.3, so that the interaction radius is based on the closest 30% of nearest-neighbor distances. The merging threshold is then set to $0.2h$, and the maximum number of iterations is set to 10,000.

For each dataset, Table 6.2 reports the result for $\lambda = 0$, which corresponds to the base F2C method, and the best result for $\lambda > 0$, selected according to the highest NBalance. We use also the *NBalance* as the selection criterion because the real datasets have unequal demographic group proportions, and normalized balance better reflects whether the produced clusters match the global demographic distribution.

Table 6.2: Base and best fairness-aware F3C results on real datasets.

Dataset	λ	Clusters	NBalance	Balance	Silhouette	SSE
Adult	0.0	57	0.776	0.473	-0.184	2968.9
	0.8	84	0.794	0.498	-0.178	3128.7
Credit Card	0.0	106	0.788	0.588	-0.179	3894.3
	0.6	119	0.835	0.622	-0.185	3988.0
Bank	0.0	29	0.769	0.628	0.183	3219.6
	0.8	45	0.792	0.642	0.263	3395.0

Table 6.2 compares the base F2C behavior ($\lambda = 0$) with the best fairness-aware result obtained by F3C on each real dataset. In all three datasets, the best fairness-aware setting improves NBalance compared with the base method. For Adult, NBalance increases from 0.776 to 0.794, while Balance also improves from 0.473 to 0.498. For Credit Card, NBalance increases from 0.788 to 0.835 and Balance from 0.588 to 0.622. For Bank, NBalance increases from 0.769 to 0.792 and Balance from 0.628 to 0.642.

The effect on clustering quality is mixed. On Adult and Credit Card, the Silhouette Score remains negative, showing that the produced clusters are not well separated geometrically. On Bank, the Silhouette Score is positive and even improves for the fairness-aware setting, from 0.183 to 0.263. In all three datasets, the number of clusters increases when fairness is introduced, and this is the reason that the SI have decreased. This suggests that the fairness force encourages additional separations between active groups, improving demographic balance but also changing the geometric structure of the clustering.

The U.S. Census dataset is not included in the final table, because the algorithm did not complete within the available computational time. This is mainly due to the larger dimensionality and size of the dataset, which make the computation of particle interactions significantly more expensive.

6.4 Discussion

The experiments in this chapter study two aspects of the proposed method. First, the base F2C experiments examine whether the force-based movement and merging mechanism can recover meaningful cluster structures without using fairness. This is important because the fairness-aware method builds directly on this base mechanism.

Second, the F3C experiments study how the fairness force changes the behavior of the algorithm. As in the previous chapters, the goal is not to maximize fairness alone, but to study the trade-off between demographic balance and clustering quality. Increasing λ can improve Balance by encouraging interactions between groups with different demographic compositions. However, strong fairness forces may also affect the spatial structure of the clusters, which can lead to lower Silhouette values, higher SSE, or lower agreement with the reference labels.

Limitations. Although F3C provides a flexible way to incorporate fairness into a force-based clustering process, its behavior depends on several algorithmic parameters. The displacement rule controls how particles move at each iteration, while the merging threshold determines when nearby particles are combined into the same cluster. Therefore, different choices of these parameters may affect the stability of the process and the final clustering result.

Moreover, the fairness term does not only influence the demographic composition of the clusters, but can also affect the merging dynamics. In some cases, increasing the fairness parameter λ may improve Balance or *NBalance*, but it may also lead to a larger number of final clusters. For this reason, fairness improvements should be interpreted together with the number of produced clusters and the clustering-quality metrics, such as Silhouette and SSE.

A further direction for future work is a broader experimental comparison of F2C and F3C against both standard clustering algorithms and fairness-aware clustering methods. This would allow a more complete assessment of whether the observed fairness-quality trade-offs are specific to the proposed force-based formulation or also hold relative to existing fair clustering approaches.

Overall, this chapter shows how the same physics-inspired fairness framework can be applied beyond centroid-based clustering. In F3C, fairness acts directly between active groups of points, and the final clusters are formed through movement and merging rather than through point-to-centroid assignments.

CHAPTER 7

CONCLUSIONS

In this thesis, we studied the problem of fairness-aware partitioning in unsupervised learning, focusing on community detection and clustering. Traditional community detection and clustering algorithms usually optimize structural or geometric quality criteria, such as graph connectivity or spatial compactness, without considering the demographic composition of the produced groups. As a result, the output clusters or communities may be structurally meaningful, but demographically imbalanced.

To address this issue, we proposed a unified physics-inspired framework for fairness. The main idea is to represent clusters, communities, and active groups through fairness charges that depend on their demographic composition. The sign of a charge indicates the over-represented group, while its magnitude captures the degree of imbalance. These charges define a fairness force that encourages interactions or assignments that improve demographic balance, while discouraging assignments that would increase imbalance. At the same time, the original structural objective of each algorithm is preserved through a structural or geometric attraction term.

Within this framework, we developed three methods. First, we proposed Fair Label Propagation (FLP), a fairness-aware extension of the Label Propagation algorithm for community detection. The method modifies the label update rule by combining the standard neighborhood-based pulling force with an electrostatic fairness force. The experiments showed that FLP can substantially improve the balance of the detected communities.

Second, we proposed Forced-Field Fair K-Means (F3KM), a fairness-aware extension of K -means clustering. The method keeps the centroid-based structure of

standard K -means, but modifies the assignment step using a force-based score. This score combines the geometric attraction between points and centroids with a fairness correction based on the demographic imbalance of the candidate cluster. The experimental results showed that F3KM can improve demographic balance while preserving much of the original distance-based clustering structure. Compared with more aggressive fair clustering baselines, F3KM provides a softer local correction and therefore offers a moderate fairness-quality trade-off.

Finally, we studied a dynamic force-based clustering approach. We first presented F2C, a base method where data points move as particles under geometric forces and clusters are formed through merging. We then extended this mechanism to F3C by adding fairness forces between active groups of points.

Overall, the results of the thesis show that force-based fairness mechanisms can be a flexible way to incorporate demographic constraints into unsupervised partitioning methods. The proposed framework can be adapted to different settings, including graph-based community detection, centroid-based clustering, and particle-based clustering. In all cases, the main idea is the same: fairness is introduced as a local force that interacts with the original structural objective of the algorithm.

At the same time, the experiments show that fairness cannot be improved without considering its effect on quality. In community detection, improving balance may reduce modularity when the algorithm joins nodes that are not strongly connected. In clustering, improving demographic balance may increase SSE or reduce Silhouette when the fair assignment moves points away from their closest geometric cluster. Therefore, the goal is not only to maximize fairness, but to find a useful trade-off between fairness and structural quality.

There are several directions for future work. First, alternative definitions of the fairness force could be explored, including smoother or bounded versions of the force term. This may help reduce instability in cases where the fairness component becomes too strong. Second, the effect of the distance component could be further studied, especially in clustering methods where inverse-distance forces may dominate the final assignments. Third, the proposed framework could be extended to datasets with more than two sensitive groups, or to settings where multiple sensitive attributes are considered simultaneously. Finally, a more systematic parameter search could be performed for the proposed methods and for the baselines, especially in the particle-based clustering setting.

In conclusion, this thesis shows that physics-inspired force models offer an interpretable and adaptable mechanism for fairness-aware community detection and clustering. By combining structural attraction with demographic fairness forces, the proposed methods can improve the demographic balance of the produced partitions while maintaining a meaningful connection to the original clustering or community detection objective.

BIBLIOGRAPHY

- [1] A. W. Flores, K. Bechtel, and C. T. Lowenkamp, “False positives, false negatives, and false analyses: A rejoinder to “machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks”,” *Federal Probation*, vol. 80, no. 2, pp. 38–46, September 2016.
- [2] A. Datta, M. C. Tschantz, and A. Datta, “Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination,” 2015. [Online]. Available: <https://arxiv.org/abs/1408.6491>
- [3] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner, “Scalable fair clustering,” in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 405–413.
- [4] A. Chhabra, K. Masalkovaitė, and P. Mohapatra, “An overview of fairness in clustering,” *IEEE Access*, vol. 9, pp. 130 698–130 720, 2021.
- [5] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, “Fair clustering through fairlets,” *Advances in neural information processing systems*, 2017.
- [6] C. Rösner and M. Schmidt, “Privacy preserving clustering with constraints,” *CoRR*, vol. abs/1802.02497, 2018.
- [7] S. K. Bera, D. Chakrabarty, N. J. Flores, and M. Negahbani, “Fair algorithms for clustering,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.
- [8] S. Ahmadian, A. Epasto, M. Knittel, R. Kumar, M. Mahdian, B. Moseley, P. Pham, S. Vassilvitskii, and Y. Wang, “Fair hierarchical clustering,” *CoRR*, vol. abs/2006.10221, 2020. [Online]. Available: <https://arxiv.org/abs/2006.10221>

- [9] M. Kleindessner, P. Awasthi, and J. Morgenstern, “Fair k-center clustering for data summarization,” in *Proceedings of the 36th International Conference on Machine Learning, ICML*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 3448–3457.
- [10] X. Chen, B. Fain, L. Lyu, and K. Munagala, “Proportionally fair clustering,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 1032–1041.
- [11] M. Ghadiri, S. Samadi, and S. Vempala, “Socially fair k-means clustering,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT ’21, 2021, p. 438–448.
- [12] A. Chhabra and P. Mohapatra, “Fair algorithms for hierarchical agglomerative clustering,” 2023. [Online]. Available: <https://arxiv.org/abs/2005.03197>
- [13] S. Bandyapadhyay, T. Inamdar, S. Pai, and K. Varadarajan, “A constant approximation for colorful k-center,” 2019. [Online]. Available: <https://arxiv.org/abs/1907.08906>
- [14] K. Manolis and E. Pitoura, “Modularity-based fairness in community detection,” p. 126–130, 2024. [Online]. Available: <https://doi.org/10.1145/3625007.3627518>
- [15] C. Gkartzios, E. Pitoura, and P. Tsaparas, “Fair network communities through group modularity,” in *WWW*, 2025.
- [16] —, “Modularity-fair deep community detection with multi-valued sensitive attributes,” 2026.
- [17] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, “Fairness through awareness,” in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS ’12, 2012, p. 214–226.
- [18] B. Brubach, D. Chakrabarti, J. P. Dickerson, S. Khuller, A. Srinivasan, and L. Tsepenekas, “A pairwise fair and community-preserving approach to k-center clustering,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML’20, 2020.
- [19] N. Anderson, S. K. Bera, S. Das, and Y. Liu, “Distributional individual fairness in clustering,” *CoRR*, vol. abs/2006.12589, 2020.

- [20] D. Chakrabarti, J. P. Dickerson, S. A. Esmaeili, A. Srinivasan, and L. Tsepenekas, “A new notion of individually fair clustering: -equitable k -center,” in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022, pp. 6387–6408.
- [21] C. Jung, S. Kannan, and N. Lutz, “A center in your neighborhood: Fairness in facility location,” 2019. [Online]. Available: <https://arxiv.org/abs/1908.09041>
- [22] M. Kleindessner, P. Awasthi, and J. Morgenstern, “A notion of individual fairness for clustering,” *CoRR*, vol. abs/2006.04960, 2020.
- [23] F. Corriera, F. W. Takes, and A. Saxena, “Individual fairness in community detection: Quantitative measure and comparative evaluation,” 2026. [Online]. Available: <https://arxiv.org/abs/2602.16326>
- [24] S. Bandyapadhyay, F. V. Fomin, and K. Simonov, “On coresets for fair clustering in metric and euclidean spaces and their applications,” 2020. [Online]. Available: <https://arxiv.org/abs/2007.10137>
- [25] M. Schmidt, C. Schwiegelshohn, and C. Sohler, “Fair coresets and streaming algorithms for fair k -means,” in *Approximation and Online Algorithms*, E. Bampis and N. Megow, Eds. Cham: Springer International Publishing, 2020, pp. 232–251.
- [26] L. Huang, S. Jiang, and N. Vishnoi, “Coresets for clustering with fairness constraints,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/810dfbbebb17302018ae903e9cb7a483-Paper.pdf
- [27] M. Setzu and R. Guidotti, “Fair clustering with clusterlets,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.06259>
- [28] S. Mahabadi and A. Vakilian, “Individual fairness for k -clustering,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 6586–6596. [Online]. Available: <https://proceedings.mlr.press/v119/mahabadi20a.html>

- [29] S. Esmaeili, B. Brubach, L. Tsepenekas, and J. Dickerson, “Probabilistic fair clustering,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 743–12 755. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/95f2b84de5660ddf45c8a34933a2e66f-Paper.pdf
- [30] S. Duppala, J. Luque, J. P. Dickerson, and S. A. Esmaeili, “Robust fair clustering with group membership uncertainty sets,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.00599>
- [31] K. Kim, J. Lee, S. Park, and Y. Kim, “Fair clustering via alignment,” in *Proceedings of the 42nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR, 13–19 Jul 2025, pp. 30 363–30 399. [Online]. Available: <https://proceedings.mlr.press/v267/kim25t.html>
- [32] I. M. Ziko, J. Yuan, E. Granger, and I. Ben Ayed, “Variational fair clustering,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11 202–11 209, May 2021. [Online]. Available: <https://doi.org/10.1609/aaai.v35i12.17336>
- [33] R. Pan, C. Zhong, and J. Qian, “Balanced fair k-means clustering,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 5914–5923, 2024.
- [34] S. Gupta, G. Ghalme, N. C. Krishnan, and S. Jain, “Efficient algorithms for fair clustering with a new notion of fairness,” *Data Min. Knowl. Discov.*, vol. 37, no. 5, p. 1959–1997, Mar. 2023. [Online]. Available: <https://doi.org/10.1007/s10618-023-00928-6>
- [35] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, “Fair algorithms for clustering,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/fc192b0c0d270dbf41870a63a8c76c2f-Paper.pdf

- [36] M. Kleindessner, P. Awasthi, and J. Morgenstern, “Fair k-center clustering for data summarization,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 3448–3457. [Online]. Available: <https://proceedings.mlr.press/v97/kleindessner19a.html>
- [37] G. Zhou, H. Xu, H. Xu, C. Li, and D. Yan, “A computational approach to improving fairness in k-means clustering,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.22984>
- [38] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern, “Guarantees for spectral clustering with fairness constraints,” in *ICML*, 2019.
- [39] J. Wang, D. Lu, I. Davidson, and Z. Bai, “Scalable spectral clustering with group fairness constraints,” in *AISTATS*, 2023.
- [40] G. Panayiotou and M. Magnani, “Fair-mod: Fair modular community detection,” in *Complex Networks & Their Applications XIII*, H. Cherifi, M. Donduran, L. M. Rocha, C. Cherifi, and O. Varol, Eds. Cham: Springer Nature Switzerland, 2025, pp. 91–102.
- [41] C. Gkartzios, E. Pitoura, and P. Tsaparas, “Modularity-fair deep community detection,” in *2025 IEEE International Conference on Data Mining (ICDM)*, 2025, pp. 287–296.
- [42] A. Anagnostopoulos, L. Becchetti, A. Fazzzone, C. Menghini, and C. Schwiegelshohn, “Spectral relaxations and fair densest subgraphs,” in *ACM CIKM*, 2020.
- [43] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3–5, p. 75–174, Feb. 2010.
- [44] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Phys. Rev. E*, vol. 76, 2007.
- [45] X. Liu and T. Murata, “How does label propagation algorithm work in bipartite networks?” in *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, 2009, pp. 5–8.

- [46] G. Cordasco and L. Gargano, “Community detection via semi-synchronous label propagation algorithms,” in *BASNA*, 2010.
- [47] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *AAAI*, 2015.
- [48] J. Leskovec and J. Mcauley, “Learning to discover social circles in ego networks,” in *NeurIPS*, 2012.
- [49] B. Rozemberczki and R. Sarkar, “Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models,” in *ACM CIKM*, 2020.
- [50] M. R. Weeks, S. Clair, S. P. Borgatti, K. Radda, and J. J. Schensul, “Social networks of drug users in high-risk sites: Finding the connections,” *AIDS and Behavior*, vol. 6, no. 2, 2002.
- [51] R. Mastrandrea, J. Fournet, and A. Barrat, “Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys,” *PLOS ONE*, vol. 10, no. 9, 09 2015.
- [52] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E*, vol. 70, 2004.
- [53] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [54] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. University of California Press, 1967, pp. 281–297.
- [55] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377042787901257>
- [56] R. Kohavi *et al.*, “Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.” in *Kdd*, vol. 96, 1996, pp. 202–207.

- [57] S. Moro, P. Cortez, and P. Rita, “A data-driven approach to predict the success of bank telemarketing,” *Decision Support Systems*, vol. 62, pp. 22–31, 2014.
- [58] I.-C. Yeh and C.-h. Lien, “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients,” *Expert systems with applications*, vol. 36, no. 2, pp. 2473–2480, 2009.
- [59] C. Meek, B. Thiesson, and D. Heckerman, “The learning curve method applied to clustering,” in *International Workshop on Artificial Intelligence and Statistics*. PMLR, 2001, pp. 196–202.
- [60] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2, pp. 193–218, 1985.
- [61] A. Strehl and J. Ghosh, “Cluster ensembles—a knowledge reuse framework for combining multiple partitions,” *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.
- [62] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [63] G. Toulina and P. Tsaparas, “A fair label propagation community detection algorithm,” in *Social Networks Analysis and Mining*, A. An, A. Cuzzocrea, and H. Hu, Eds. Cham: Springer Nature Switzerland, 2026, pp. 42–52.

AUTHOR'S PUBLICATIONS

[63]