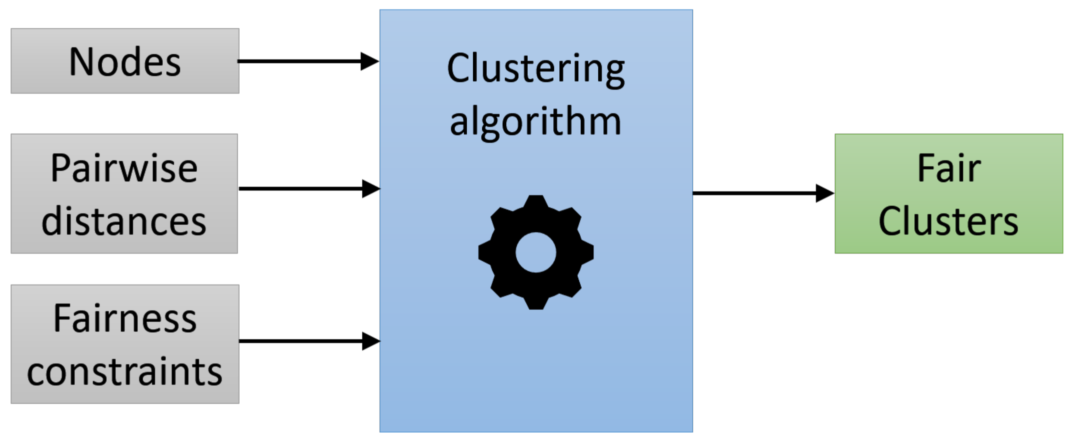
# Learning to Generate Fair Clusters from Demonstrations

Sainyam Galhotra, Sandhya Saisubramanian, and Shlomo Zilberstein

College of Information and Computer Sciences,
University of Massachusetts Amherst

## Motivation

### Fair Clustering




### Challenges

- Many different ways to define and measure fairness
- Difficult to fine tune constraint parameters like fairness thresholds
- Inadvertent incomplete specification of fairness metrics leads to biased outcomes when deployed

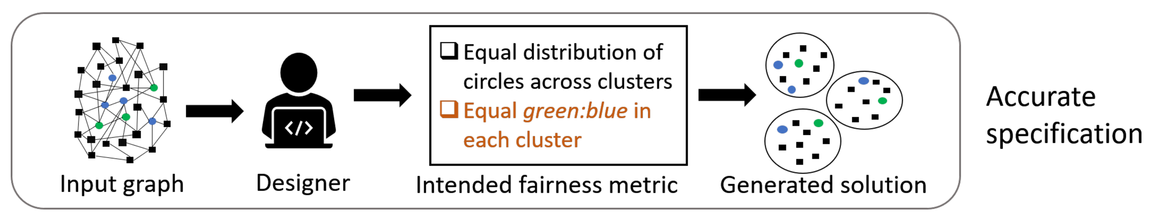



Figure: An illustration of ideal setting with accurate specification.

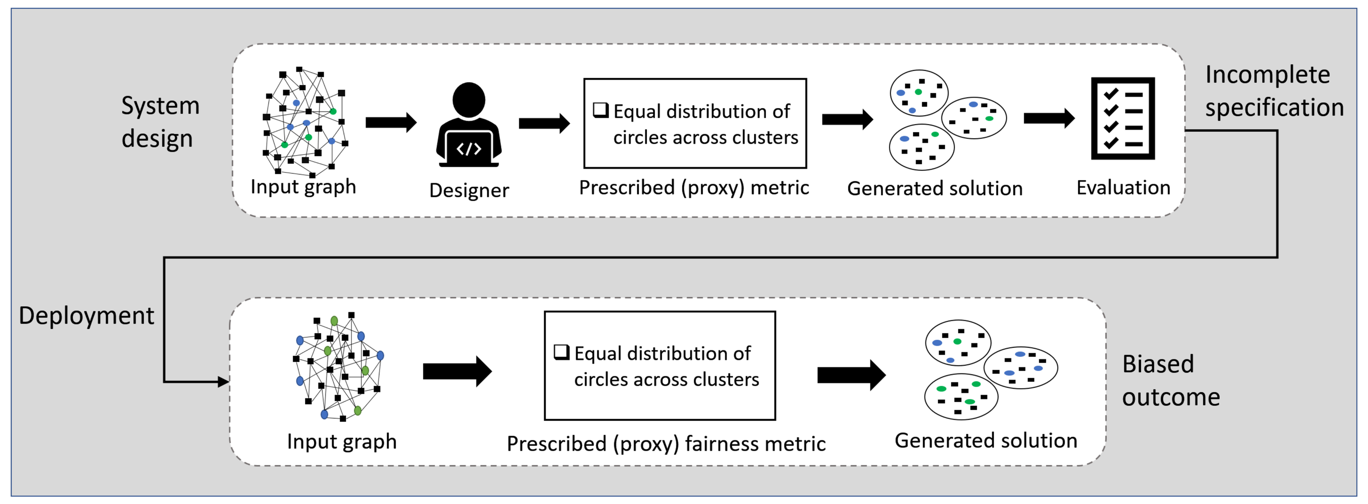



Figure: An illustration of incomplete specification of fairness metric resulting in biased output– unequal distribution of green and blue nodes in each cluster.

How to correctly identify the fairness metric that the designer intends to optimize for a problem?

| Symbol | Formula | Parameter |
|---|---|---|
| $\omega_{GF}$ | Ratio of each feature value $\in [\alpha, \beta]$ | $\alpha, \beta$ |
| $\omega_{EQ}$ | Relative distribution of a specific feature value | $\beta$ |
| $\omega_{IC}$ | Homogeneity of clusters | $\beta$ |

Table: Example fairness and interpretable constraints.

## Problem Setting

An oracle generates example demonstrations on a subset of nodes to guide the search for desired fairness constraint.

- A **clustering demonstration** $\lambda$ provides the inter-cluster and intra-cluster links for a subset of nodes from the dataset $T \subseteq V, |T| \geq 2$, by grouping them according to the underlying objective function and constraints, $\lambda = \{C_1, \ldots, C_t\}$ with each $C_i$ denoting a cluster such that $\cup_i C_i = T$ and $t \leq k$.
- A **Globally informative demonstration** provides the true cluster affiliation of a subset of nodes, $T \subseteq V$, and is denoted by $\lambda_g = \{\langle u_1, \gamma(u_1)\rangle, \ldots, \langle u_t, \gamma(u_t)\rangle\}$, $\forall u_i \in T$ with $\gamma(u)$ indicating the cluster affiliation of node $u$.

**Assumption:** Nodes in each demonstration are randomly selected and clustered according to ground-truth fairness constraints

**Objective:** Given a finite set of candidate fairness metrics ($\Omega$) and a finite set of clustering demonstrations ($\Lambda$), identify a fairness metric $\omega_F \in \Omega$ required to be satisfied by the clusters when optimizing objective $o$.

## Contributions

- Formalizing the problem of learning to generate fair clusters from demonstrations
- Presenting two algorithms to identify the fairness constraints, generate fair clusters, and analyzing their theoretical guarantees
- Empirically demonstrating the effectiveness of our approach in identifying the clustering constraints on three data sets
- Generating fair and interpretable clusters with our approach

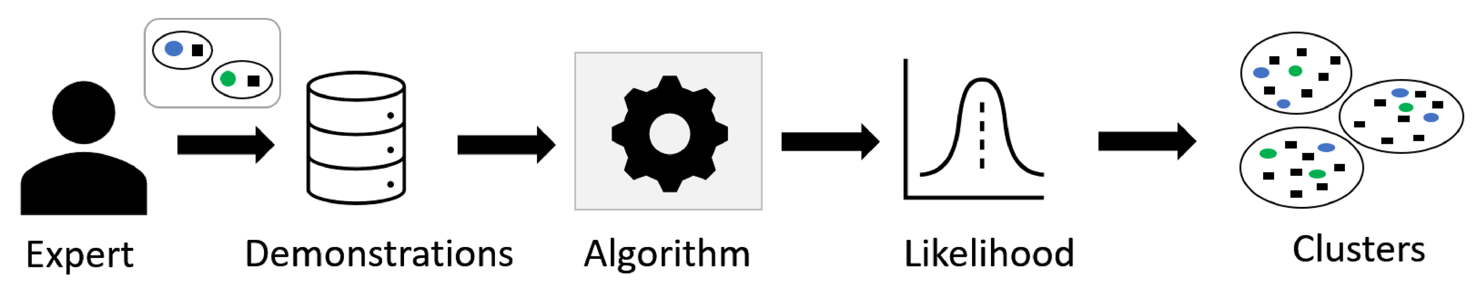



Figure: Overview of our approach.

## Algorithm Intuition

**Maximum Likelihood estimation:** Assume access to techniques that optimize fairness objectives $\omega \in \Omega$

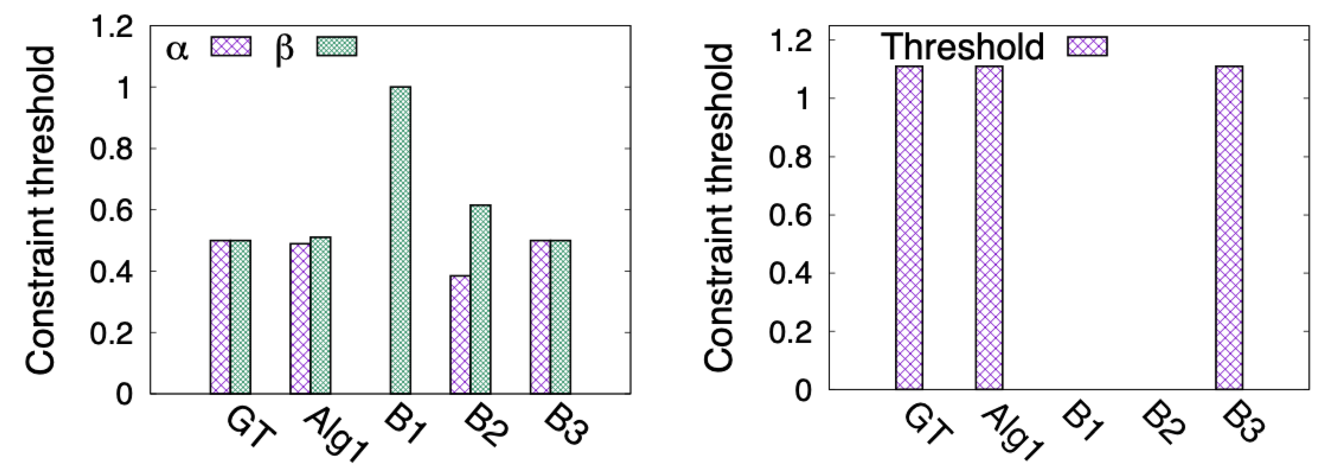
1. Initialize the set of clusters according to the demonstrations $\lambda$
2. Greedily merge closest pair of clusters until $k$ clusters are left
3. Calculate constraint threshold for each fairness constraint and feature combination
4. Run traditional fair clustering algorithm for each constraint with estimated threshold values
5. Choose the final clustering that has maximum likelihood of generating $\lambda$

**Greedy Clustering:** Initialize all nodes in a separate singleton cluster. Iteratively merge nodes to form $k$ clusters. Perform local search to satisfy most likely constraint estimated using maximum likelihood.

## Experimental Results

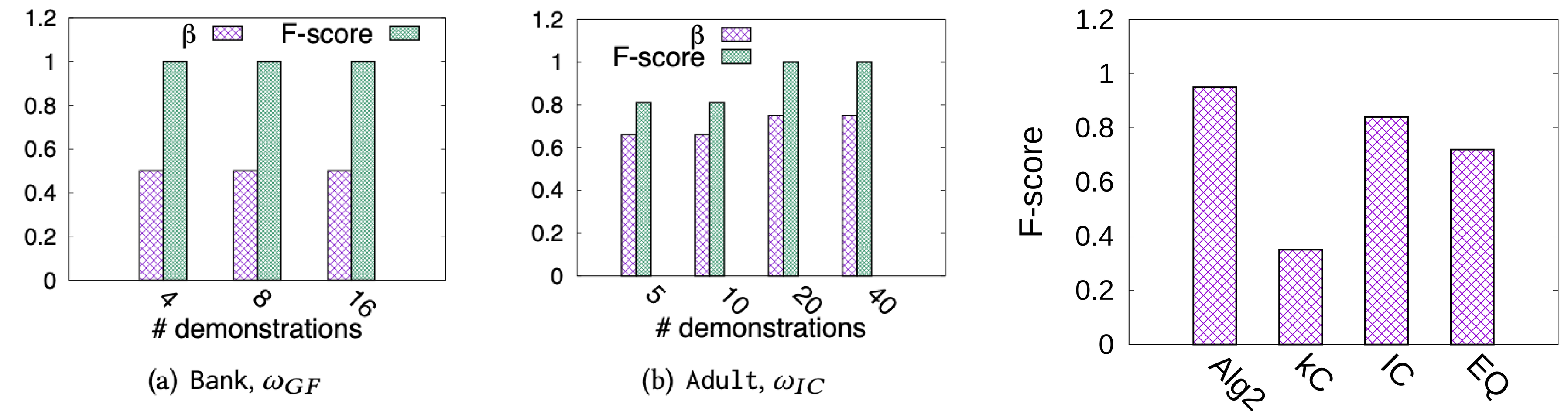
Empirically tested on 3 domains with various baselines. Additional results in paper.

### 1. Comparison of estimated constraints for different techniques




(a) Bank, $\omega_{GF}$ (b) Adult, $\omega_{EQ}$

### 2. Effect of #demonstrations and multi-constraint setting




(a) Effect of #demonstrations (b) Fair and Interpretable clusters

### Key Takeaways:

- Our approach identifies the fairness constraint in less than $2 \log n$ demonstrations
- Our algorithms construct the desired set of clusters and are highly efficient