Supplemental Material
MANI-Rank: Multiple Attribute and Intersectional Group Fairness for Consensus Ranking

1 Case Study of Computer Science Department Rankings

While group fairness concerns typically emerge when people are being being ranked, they also apply for other ranked entities. We illustrate the effectiveness our framework on removing bias in Computer Science Department rankings using the publicly available CSRankings from csrankings.org [1]. We collected rankings of 65 departments in the US over the period of $2000 - 2020$ (utilizing the relative order presented on csrankings.org as the ranking for each year) to generate a 20-year consensus ranking. To examine geographic and private vs. public institutional bias, we used the location (Northeast, South, West, Mid-West) and type (Public or Private) of each institution as protected attributes.

In Table 1, we observe that the rankings over the years indeed exhibit bias, namely, a strong bias in the location attribute (high ARP)−which stems from institutions in the Northeast region commonly appearing at the top of the rankings, along with those in South region exhibiting the opposite trend. Additionally, there is a significant amount of intersectional bias − resulting from the base rankings having Private and Northeast colleges highly ranked.

When using fairness-unaware Kemeny to create the 20-year consensus ranking, we observe that this bias is amplified, with the Location ARP score resembling that of base rankings with higher such scores, and the IRP score close to 0.6. By utilizing our *MANI-Rank* criteria and setting the group fairness threshold $\Delta = .05$, it can be seen that we were able to remove the bias in the consensus ranking. All our proposed methods remove the bias toward Northeast and Private universities.

Table 1: **CSRankings Study:** Attribute values columns (e.g, Northeast, Prive) indicate FPR scores, and Location, and Type indicate ARP scores.

Ranking	***Northeast***	***Midwest***	***West***	***South***	***Location***	***Private***	***Public***	***Type***	***IRP***
2000	0.692	0.373	0.574	0.315	0.377	0.642	0.358	0.284	0.536
2001	0.712	0.370	0.598	0.270	0.442	0.608	0.392	0.217	0.582
2002	0.700	0.451	0.610	0.203	0.497	0.569	0.431	0.138	0.574
2003	0.689	0.392	0.589	0.287	0.402	0.562	0.438	0.124	0.557
2004	0.658	0.481	0.570	0.265	0.392	0.610	0.390	0.221	0.520
2005	0.680	0.457	0.552	0.278	0.402	0.607	0.393	0.213	0.480
2006	0.707	0.451	0.550	0.255	0.452	0.582	0.418	0.164	0.476
2007	0.691	0.435	0.601	0.236	0.455	0.607	0.393	0.213	0.553
2008	0.724	0.416	0.550	0.265	0.459	0.610	0.390	0.221	0.515
2009	0.693	0.469	0.599	0.205	0.488	0.591	0.409	0.181	0.570
2010	0.687	0.426	0.582	0.268	0.419	0.608	0.392	0.217	0.521
2011	0.714	0.426	0.573	0.245	0.470	0.607	0.393	0.215	0.559
2012	0.694	0.460	0.574	0.237	0.457	0.582	0.418	0.164	0.542
2013	0.738	0.436	0.533	0.249	0.489	0.606	0.394	0.211	0.545
2014	0.674	0.476	0.561	0.259	0.416	0.582	0.418	0.164	0.495
2015	0.726	0.469	0.532	0.236	0.490	0.599	0.401	0.197	0.561
2016	0.711	0.470	0.517	0.267	0.445	0.606	0.394	0.211	0.512
2017	0.688	0.478	0.540	0.264	0.424	0.547	0.453	0.095	0.448
2018	0.661	0.504	0.559	0.253	0.409	0.551	0.449	0.103	0.457
2019	0.680	0.499	0.531	0.264	0.416	0.566	0.434	0.132	0.467
2020	0.666	0.445	0.540	0.318	0.348	0.579	0.421	0.158	0.415
Kemeny	0.712	0.445	0.570	0.233	0.479	0.602	0.398	0.203	0.570
Fair-Kemeny	0.532	0.499	0.532	0.432	0.100	0.498	0.502	0.004	0.099
Fair-Schulze	0.529	0.501	0.529	0.436	0.093	0.494	0.506	0.012	0.089
Fair-Borda	0.519	0.499	0.534	0.445	0.089	0.496	0.504	0.008	0.094
Fair-Copeland	0.531	0.501	0.531	0.432	0.099	0.494	0.506	0.012	0.099

2 Pseudocode for Fair-Copeland, Fair-Schulze, and Fair-Borda

Pseudocode for Fair-Copeland.

Algorithm 1: Fair-Copeland

Input : Base rankings R, candidate database X, thresholds Δ
Output: fair consensus ranking π^{C*}

```
W_mat ← precedence_matrix(R)
copeland_scores ← empty Dictionary /* with candidates as keys
    */

foreach x_i ∈ X do
    foreach x_j ∈ X do
        if x_i != x_j then
            x_i_wins = W_mat[x_j, x_i]
            x_j_wins = W_mat[x_i, x_j]
            if x_i_wins ≥ x_j_wins then
                copeland_scores[x_i] += 1
            end
        end
    end
end
copeland_π = sort(copeland_scores) /* sort candidates by
    decreasing copeland score                              */
return Make-MR-Fair(copeland_π, X, Δ)
```

Pseudocode for Fair-Schulze.

Algorithm 2: Fair-Schulze

Input : Base rankings R, candidate database X, thresholds Δ
Output: fair consensus ranking π^{C*}

$W_{mat} \leftarrow precedence\_matrix(R)$
$P_{mat} \leftarrow$ empty array `/* n by n array for path strengths */`

foreach $x_i \in X$ **do**
  foreach $x_j \in X$ **do**
    if $W_{mat}[x_j, x_i] > W_{mat}[x_i, x_j]$ **then**
      $P_{mat}[x_j, x_i] = W_{mat}[x_j, x_i]$
    end
    else
      $P_{mat}[x_j, x_i] = 0$
  end
end
foreach $x_i \in X$ **do**
  foreach $x_j \in X$ **do**
    if $x_j! = X_i$ **then**
      foreach $x_k \in X$ **do**
        if $x_i! = x_k x_j! = x_k$ **then**
          $P_{mat}[x_k, x_j] =$
          $\max(\mathrm{P}_{mat}[x_k, x_j], min(P_{mat}[x_i, x_j], P_{mat}[x_k, x_i]))$
        end
      end
    end
  end
end
schulze_scores = columnsum(Pmat)
`/* index is candidate id */`
schulze_π = sort(schulze_scores) `/* sort candidates by decreasing schulze score */`
return Make-MR-Fair(schulze_π, X, Δ)

Pseudocode for Fair-Borda.

Algorithm 3: Fair-Borda

```
Input  : Base rankings R, candidate database X, thresholds Δ
Output: fair consensus ranking π^{C*}

borda_scores ← empty Dictionary /* with candidates as keys */

points_pos ← array range(n, -1, 0) /* array representing the
    points alloted to each position */

foreach r ∈ R do
    foreach i ∈ range(0, n) do
        item = R[r, i]
        borda_scores[item]+ = points_pos[i]
    end
end
borda_π = sort(borda_scores) /* sort candidates by decreasing
    borda score */
return Make-MR-Fair(borda_π, X, Δ)
```

References

[1] E. Berger, “Csrankings: Computer science rankings,” 2018.