Response for #4148, “Coresets for Clustering with Fairness Constraints”

We thank the reviewers for their valuable comments. We start with answering questions raised by more than one reviewer and then answer reviewer-specific questions.

New experiments (Reviewers 1 and 3): Based on your suggestions, we conducted new experiments to study the speed-up our coresets obtain for recent fair clustering algorithms [3, 4]. We observed that simple adaption of our coresets already offers at least a 10x speed-up to **FairTree** [3] and a 2x speed-up to **FairLP** [4], even taking the coreset construction time into consideration; see Table 1a for some preliminary results. (The same coreset could be used for clustering with any fairness constraints, so its construction time would be averaged out if multiple fair clustering tasks are performed.) We believe that with a carefully crafted implementation of these results that integrate with our coresets would further accelerate the running time. We will include a thorough comparison in the final version of the paper.

Table 1: new empirical results

	T_{ALG} (s)	T'_{ALG} (s)	T_{COR} (s)
FairTree	444.31	22.70	26.14
FairLP	425.70	169.05	-

(a) speed-up of fair clustering using coresets on **Census1990** (n $\approx$ 2.5m). T_{ALG} is run time w/o coreset, T'_{ALG} is run time on top of our coreset, and T_{COR} is time to construct coreset.

ϵ	emp. err. Ours	**BICO**	**Uni.**	size	time (ms)
10%	0.28%	1.04%	10.63%	880	43.88
20%	0.55%	1.12%	28.73%	610	29.36
30%	1.17%	4.06%	19.91%	503	25.97
40%	2.40%	4.45%	48.10%	433	22.17

(b) performance of ε-coresets for fair k-means w.r.t. varying ε on **Adult** (n $\approx$ 30k), with gender and marital status as sensitive attributes.

We also conducted experiments on additional datasets including **Athletes** and **Diabetes**, and we employed a new uniform sampling baseline. We report partial results on **Adult** dataset in Table 1b, and complete results will appear in the final version of the paper. The results in the table show that the empirical error of our coreset is at most 50% of **BICO**, and 10% of **Uniform**. We note that the performance improvement to **BICO** is more significant than in our submission version due to the recently improved implementation of our coresets. Our coreset construction time scales roughly linearly with the size of the coreset, and as mentioned above, its efficiency leads to accelerated fair clustering algorithms (Table 1a). We also observe that the performance of **Uniform** is comparable to ours and **BICO** on certain datasets such as **Diabetes**, but its worst-case error is unbounded in general.

Novel contributions (Reviewers 2 and 3): Conceptually, we obtain the first coresets for fair clustering w.r.t. *multiple types* whose size is *independent* of the size of the dataset. Technically, to handle the multiple types, we show a general reduction that turns any fair coreset for a binary type into a coreset for multiple types. Although our coreset for a binary type is based on the framework of [25], [25] does not readily apply to fair clustering, due to the technical hurdle that data points may not be assigned to the nearest center in an optimal fair clustering. We cross this hurdle by showing new structural lemmas (Lemmas 4.1 and B.2) for 1-dimensional dataset X, that the optimal fair k-median/means clustering partitions X into $O(k)$ contiguous intervals. Empirically, as in Table 1a, our coreset may be applied to speed-up several very recent fair clustering algorithms [3,4] on various datasets, achieving more than 10x acceleration for [3].

Presentation (Reviewers 1 and 2): In the final version, we will add pseudocodes of our algorithms, a figure to visualize the 1D case in Section 4.1, and examples to illustrate the coreset construction. We will also make the paper more self-contained by adding a short “background” section that explains the framework of [25] and its technical limitations.

Reviewer 1: Regarding the terms “collections of groups” and “types”: We will clarify their definitions in the final version of the paper.

Reviewer 2: Regarding the k-center objective: Our coreset construction does work for k-center by combining Theorem 4.2 and the results of [24], which was also mentioned in Footnote 1.

Regarding the concern that the empirical error is superlinear with increasing ε: We didn’t quite understand your question. The only part that the empirical error seems to be superlinear is in Table 1 (Bank) for ε between 30% and 40%. Otherwise, the empirical error is roughly linear with increasing ε.

Reviewer 3: Regarding the re-definition that a weighted point may be partially assigned to more than one cluster: We do need the re-definition. The reason is that if each weighted point can only be assigned to one cluster, it is possible that the assignment constraint cannot be satisfied. For instance, suppose a coreset only contains one point with weight 100 and we require that each of two centers should be assigned by exactly 50 points. Then this point must be partially assigned to two centers.

Regarding the comment on Theorem 4.2: Theorem 4.2 actually implies an algorithm that constructs a coreset w.r.t. multiple types. We will clarify this and also discuss the running time in the final version of the paper.