

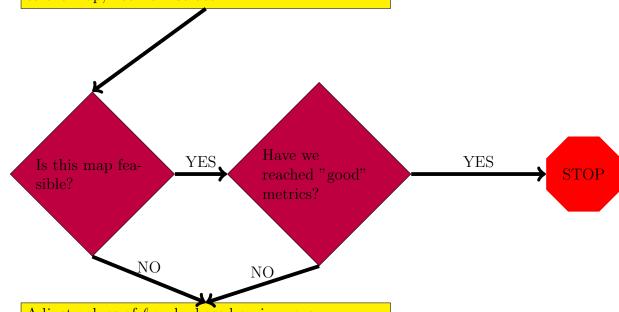
INITIALIZATION: Start with existing district map

- 1. Run metrics of interest on this map, noting this is an infeasible map as population constraints do not hold.
- 2. Set $\ell_{START} \geq k = \#$ of districts.
- 3. Set $\epsilon_{START} \leq 1$ to be the percent difference allowed from target population.

Run Recom with current value of ℓ and tolerance ϵ :

- Select ℓ adjacent districts to combine.
- Find a spanning tree of combined nodes.
- Find a list of edges that are "cuttable". (For $\ell > 2$, this will be a pair/triple/etc. of cuttable edges.)
- Separate districts by cutting/removing a randomly selected cuttable edges(s).

Update metrics by the changes that have been done to the map, not from scratch.



Adjust values of ℓ and ϵ based on improvement/temperature functions/iteration number/etc.

Check list of codes needed:

- Parts dealing with Recom algorithm:
 - Combine ℓ districts into one pool
 - Find spanning tree
 - Find list of "cuttable" edges, given ℓ and ϵ
 - Update map given separations formed by removing selected edges
- Dealing with Metrics:
 - Update metrics based on changes to maps, these methods may vary depending on the metric.
 - Some metrics we may want to include will deal with:
 - * Competitiveness
 - * Bias
 - * Compactness

We will also need to make decisions on:

- How to choose which districts to combine and "work-on"
- How will the updates to ℓ and ϵ work?