SCALABLE APPROXIMATIONS OF CAPACITATED k-MEDIANS FOR POLITICAL DISTRICTING

A PREPRINT

Wes Gurnee
Cornell University
rwg97@cornell.edu

ABSTRACT

We present a novel approach for efficiently finding high quality solutions to the uniformly capacitated k-medians problem (UCKMP) with optimality bounds for political districting. Our approach is to solve the LP relaxation to construct a weighted bipartite graph with weight equal to the value of the associated LP variable. This bipartite graph is used as the input for a spectral partitioning subroutine which picks the final k centers. These centers are then used in a transportation problem to do the final assignment of census tracts to centers in polynomial time. Our approach is scalable because the most costly part of our algorithm, solving the relaxation, is highly parallelizable and because the tracts exist in (near) euclidean space, there are a number of natural variable pruning heuristics that greatly cut down the total number of variables without affecting solution quality. We present an extensive empirical analysis of using our approach to create congressional districts for American states our of census tracts.

***K*eywords** Approximation Algorithms · k-medians · Spectral methods · Districting

1 Introduction

The problem of partitioning a space into a disjoint set such that each set satisfies some capacity constraint appears in a wide variety of application domains. In machine learning, clustering techniques are used to find hidden categorical structure in data embedded in continuous vector space. In the physical world, splitting up a city or state is a common problem that arises when creating police precincts, schools districts, political districts, and more. In most of these domains, the natural objective function is to minimize a distance metric of the assignment of points or land areas to clusters or districts. This gives rise to the k-medians problem where the objective is to minimize the sum of distances between points in the cluster and the cluster centroid.

In this study, we focus on political districting problems. In the United States, districting is a political decision. Every 10 years, after the decennial census is completed, state legislatures are responsible for drawing the districts responsible for electing members of the US House of Representatives as well as the state house and senate maps. As one might imagine, letting politicians pick their voters has undesirable consequences. In particular, the party in power at the time of redistricting "cracks" and "packs" areas of opposition voters so that their influence is diluted across multiple districts or they are all crammed into one district they win by a landslide. Furthermore, this leads to many districts that are politically very safe for each party. This creates a dynamic in which the only threat to an incumbent is from the ideological extremes and compromise becomes a bug rather than a feature. It is not hard to predict the gridlock that logically ensues.

The remedy is therefore to take control of the redistricting process out of the hands of politicians and let voters pick their politicians rather than the reverse. Seeing as districts must be contiguous and compact, a natural objective function is to maximize the compactness of the districts. However, there are many different metrics of compactness that one could reasonably use [1]. The conventional wisdom is that compact districts are politically fair (no party is advantaged) because they are generated without political consideration. Unfortunately, this turns out not to be true because of the sharp party divide between rural and urban voters [2]. Nevertheless, having fast routines for generating compact districts will be an important step in any algorithm that seeks to create fair political districts and represents an important baseline.

For geographic k-medians problems like school or political districting, the problem input is a set of atomic geographic units (tracts) that are each assigned to centers. Tracts are defined by a geographic centroid and a population. The set of possible centers is also the set of tracts and therefore the problem is to pick which tracts are centers and which tracts get assigned to each center. More formally the general k-medians problem is

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{i\in N} y_i = k \\
& \sum_{i\in N} x_{ij} = 1 && \forall j \in N \\
& x_{ij} \le y_i && \forall i,j \in E \\
& y_i,\ x_{ij} \in \{0,1\}, && \forall i,j \in E
\end{aligned}$$

where N is the set of tract indices and E is (for now) $N \times N$. x_{ij} and y_i are binary decision variables where $y_i = 1$ indicates that tract t_i is a center and $x_{ij} = 1$ indicates that tract t_j is assigned to the center tract t_i. In most application domains, certainly political districting, there is an additional set of capacity constraints that bounds the quantity of some tract attribute (population in our case) for each district. Therefore we add the constraints

$$y_i L \le \sum_{j\in N} p_j x_{ij} \le y_i U, \quad \forall i \in N$$

where p_j is the population of tract j and U, R are the upper and lower bounds for the population of a district. In general, c_{ij} corresponds to a distance metric between t_i and t_j. In our formulation

$$c_{ij} = p_j \|t_i - t_j\|_2^\alpha$$

which is the moment of inertia measure of compactness described by Young[1]. α is an additional parameter that dictates how harshly we penalize distance.

k-medians is a specific instance of a more general problem called the facility location problem (FLP). The FLP arises in situations where you need to fulfill demand by assigning various consumers to facilities that can satisfy all of the demand with minimum cost. An example is when there are a group of customers with demand for a certain resource and an operator must decide where to open production facilities to most cost effectively meet this demand. The integer program has a few differences compared to k-medians. Instead of having fixed k constraint, there is an additional term in the objective function to account for the fixed costs of opening a facility, and hence the problem is in both assignment and balancing the cost savings by having facilities closer to customers with the cost of opening more facilities in choosing k. Furthermore, it is usually natural to allow for a single customer to be served by multiple facilities, and so x_{ij} need not be binary.

Both the FLP and CKMP are NP-hard. Therefore for large problems, it is necessary to use approximation algorithms and heuristics to find good solutions.

2 Related Work

Given the great practical importance of the FLP and capacitated k-medians problem, significant previous work exists [3]. The history has evolved around solving uncapaticated k-medians, which was built on work for the FLP, and the capacity constraints are added later.

The work in uncapacitated k-medians can largely be placed in three buckets: LP-rounding [4], primal-dual[5], or local search analyses [6]. Unfortunately, Jain et al. proved that the uncapacitated k-median problem is hard to approximate within a factor of 1 + 2/e. Furthermore, the natural LP-relaxation of the UKMP has an integrality gap of at least 2 and with an upperbound of 3 [7]. The best known approximation algorithm is a (2.675 + ϵ)-approximation based on dependent rounding [8].

Things get worse when adding capacity constraints. The integrality gap of the natural capacitated relaxation is unbounded [3]. Moreover, much of the current work does not guarantee satisfaction of the capacity constraints or the k centers constraint. For example, Byrka et al. give an $O(1/\epsilon^2)$-approximation by violating the capacity constraint by a factor of (2 + ϵ) [9]. Shi Li instead presents an $O(exp(1/\epsilon^2))$-approximation that violates the k centers constraint by a factor of $(1 + \epsilon)$ [10]. It should also be noted that there are slightly different variants of the CKMP. First there is a distinction between uniform and nonuniform CKMP where the capacity constraints either are or are not the same for all

centers or facilities. Second, there is a hard and soft version of CKMP where hard implies that a center or facility can be opened just once and for soft more than once. In districting, we are solving the hard uniform capacitated problem.

There is also extensive literature in using algorithms for political districting also known under the more general term of supervised regionalization methods [11]. The taxonomy of methods is first split into methods that explicitly enforce spatial contiguity and those that do not. Although political districts are required to be contiguous, in optimizing compactness it is very rare that an output will ever not be contiguous and so we do not have an explicit enforcement mechanism in our approach. In terms of technique, most approaches are based on clustering, optimization models of compactness, or local search approaches. See Duque's excellent survey for references to specific papers. [11].

3 Approach

Our approach was motivated by the observation that we could significantly decrease solve times of the capacitated k-median integer program by not using all census tracts as centers. In particular, the time it took to solve the relaxation, and then solve the integer program with only the centers that were nonzero in the relaxation solution, was about 5-10x faster than solving the full IP from scratch with almost no loss in solution quality. However solving the relaxation was still time consuming even for smaller problems and so to scale to California with over 8000 census tracts the objective was to first reduce the size of the problem.

3.1 Pruning

In our analysis we work with 2 flavors of variable pruning, one to reduce the set of centers and one to reduce the set of assignment variables. As a reminder, in the full problem we have a set N of tracts and a set E, that is $N \times N$ in the full problem, referring to the assignment of tracts to centers. It is useful to think of this as a complete directed (where the arcs indicate the direction of assignment) graph where the problem is to prune edges that will not be needed in the final assignment.

The first style of pruning is distance based pruning of the x_{ij} assignment variables. In California, a census tract in San Francisco and a tract in Los Angeles will never be in the same district. However, the question becomes how far away do two tracts have to be from each other before we prune them? A distance too short will create an infeasible model and a distance too far will be intractable to solve. Our approach is motivated by the fact that the infeasible models are infeasible because of the capacity constraint. Therefore we threshold the radius based on the total population of tracts within that radius. More formally, let the function

$$R(t_i, p) := \min_r(\sum_{j:d(t_i,t_j)<r} p_j) > p$$

denote the minimum radius around t_i that contains at least population p. For a center t_i we let the set of tracts T_i allowed to be assigned to t_i be

$$T_i^{\beta} = \{\ j \mid \quad d(t_i, t_j) < R(t_i, \beta P/k)\}$$

where P is the total population of all census tracts, k is the number of districts, and β is the pruning parameter which we analyze empirically (results in section 4). In practice to compute function R, we do a binary search over the radius with precomputed distances. The cost is minimal especially given the savings in both model construction and solving the relaxation.

The other style of pruning we perform is randomized pruning of centers. It is overkill to allow every census tract to be a center, given that for a particular solution, one could change every center (and corresponding assignment) to a neighboring tract and the solution would be approximately the same. [1] Since the number of variables and constraints increases by a factor of the number of centers, randomly cutting centers gives a significant decrease in execution time without much of a cost to solution quality. This leads to an additional parameter ϕ, the probability that we delete a center. Each center is assigned a number uniformly at random between 0 and 1 and those y_i with random value less than ϕ are pruned, along with x_{ij} variables that corresponded to assignment to center i.

[1] It is possible that the solution stay the exact same but the objective function increases because the assignment is still optimal, there just happens to be a choice of centers that have smaller total distance.

Putting it all together, given parameters β and ϕ instead of variable sets N and E, we have a center set

$$C^{\phi} = \{\ i \mid \quad \text{urandom}(i) > \phi\}$$

and pruned edge set

$$E^{\phi\beta} = \{\ (i,j) \mid \quad i \in C^{\phi},\ j \in T_i^{\beta}\}.$$

3.2 LP solve

With aggressive initial variable pruning, the LP is solved much more efficiently [2]. However, simplex and its variants still struggle to converge for even modestly sized problems. In our experiments, we found the barrier method, an interior point optimization scheme, much more effective at quickly converging. The problem with interior point methods is that they do not maintain a basis like simplex, and therefore can't be used directly to warmstart the IP without an expensive crossover step. Fortunately, we only use the LP solution to inspect nonzero centers to further prune the set of centers down to the k final centers before solving the IP. Therefore we don't need a basis and hence can avoid crossover.

There are a few more drawbacks associated with this approach. First, pruning makes the associated LP bound only valid for the pruned problem, and not the full problem and so we lose our ability to bound the full solution. With $\phi = 0$ and β near k a bound would still hold but this doesn't achieve much cost savings and could take hours or days to compute for California. Furthermore, the barrier method is memory intensive so both time and space become practical constraints. Additionally, as an interior point method, barrier solutions typically have many near 0 y_i activations. We employ optimistic rounding here to threshold near 0 values to assume they are 0. However, the benefits of this overall approach still outweigh the costs. In particular, the barrier method is parallelizable, and therefore simply throwing more computing power at the problem will go a long way in scaling our approach to larger problem instances.

3.3 Spectral selection

Even with random center pruning at the onset, and further center pruning from setting near 0 centers in the LP solution to 0, for large problems it is still too expensive to solve the integer problem to be useful in practice. In particular, simplex still has trouble even solving the root of the branch-and-bound tree and to use the barrier method we now need to perform crossover which is often more expensive than getting the LP to converge. Setting k fixed centers transforms this problem from k-median to a transportation problem [3] which is solvable in polynomial time. Therefore to make the integer program as fast as possible to solve, the natural strategy is to fix the k centers.

To do this we construct the following weighted bipartite graph

$$B = (C^{\phi} \cup N,\ E^{\phi\beta} \mid \text{weight}(e_{ij}) = x_{ij})$$

where C^{ϕ} is the set of left vertices corresponding to centers, N is the set of right vertices corresponding to each tract, and the edge weights are the activations of the corresponding assignment variable for center i and tract j. If this graph were constructed from the optimal integer solution then it would be the case that there are simply k connected components (if we remove centers with no nonzero incident edges) and the minimum weight cut to form a k-partition is trivially the empty set. Since this graph is formed from the relaxed solution, there are many low weight fractional edges and there is probably only one connected component. However, it is still the case that there exists a very low weight edge set that could be cut to form k connected components. Spectral partitioning is just the tool to find this k-partition.

With the k components returned by spectral partitioning, each component has a set of center nodes $S_i^c \subset C^{\phi}$ and a set of tract nodes $S_i^t \subset N$ for partition i. The sets $S_1^t, \ldots, S_k^t$ form a compact and contiguous districting (empirically) but there is nothing to enforce population balance of these sets. Instead we just use the sets S_i^c, and for each set we take a weighted average of the cluster, weighted by the activation of the y_i variable from the LP solution corresponding to each center in the set. Finally we find the census tract with centroid closest to each of the cluster averages, and call this collection of tracts the final collection of centers C^{final}. Note that we only consider non pruned centroids. This is just to be consistent with the LP so the bound is still valid but for situations where the bound is irrelevant this isn't necessary. Figure 1 depicts what this process looks like on a synthetic example.

[2] It makes a significant different in model construction too which is not a trivial cost component.

[3] Named so because it comes from an application domain where you have k supply points and many demand points and the problem is to decide what demand gets satisfied from which supply points.

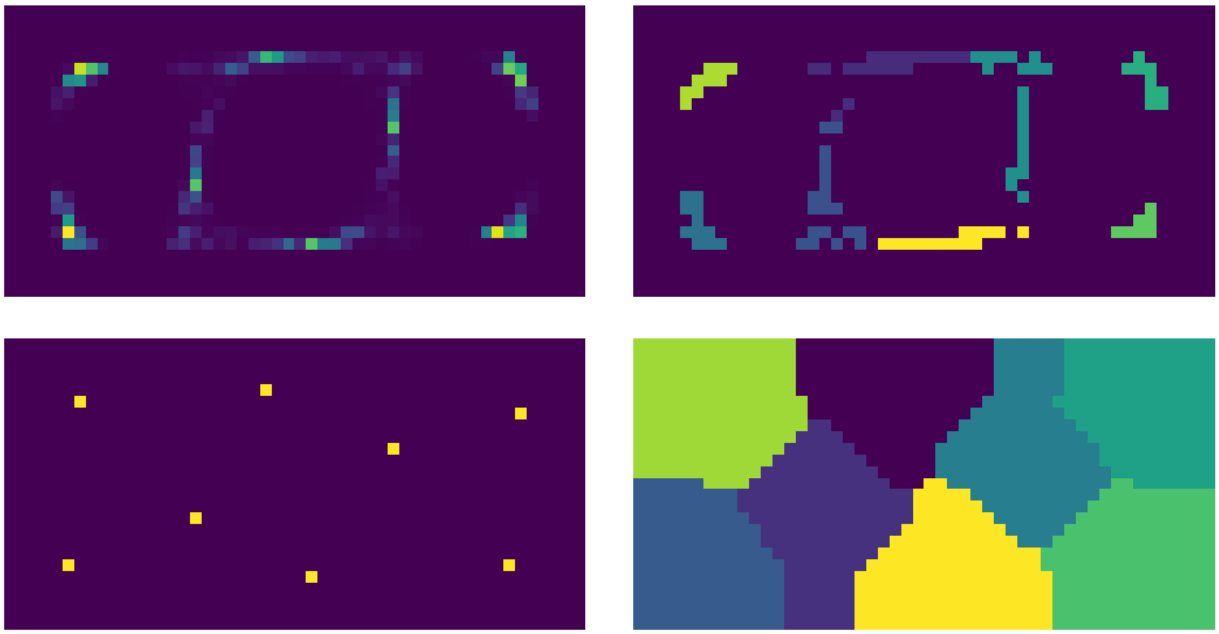

Figure 1: Top left is activations of y_i variables in the LP solution (with no pruning) of a synthetic graph with 1250 nodes and 8 districts. Top right is the resulting clusters. Bottom left are the centers resulting from the weighted average. Bottom right are the population unbalanced S_i^t sets. Note the synthetic graph does not have uniform population which is why for instance the yellow district is larger than the bottom left blue district.

3.4 IP solve

With the k centers selected, solving the IP now becomes very efficient because it is a transportation problem solvable in polynomial time. There is now no need to for y_i variables and the variable set becomes

$$E^{IP} = \{x_{ij} |\ i \in C^{final},\ j \in T_i^{\beta}\}$$

The final assignment is then the final districting and this is guaranteed to be population balanced. In some cases, if the population radius factor parameter β is too small or the population tolerance is too small for the average number of tracts per district, the model will be infeasible. In this case raising β, raising the population tolerance, or using a subtract atomic geographic unit instead will help.

Now for a complete description of the algorithm. Let

LP_KMED :			**IP_TP** :		
minimize	$\sum_{(i,j)\in E^{\phi\beta}} p_j d_{ij}^{\alpha} x_{ij}$		minimize	$\sum_{(i,j)\in E^{IP}} p_j d_{ij}^{\alpha} x_{ij}$	
subject to	$\sum_{i\in C^{\phi}} y_i = k$		subject to		
	$\sum_{i\in C^{\phi}} x_{ij} = 1$	$\forall j \in N$		$\sum_{i\in C^{final}} x_{ij} = 1$	$\forall j \in N$
	$\sum_{j\in T_i^{\beta}} p_j x_{ij} \le y_i(1+\epsilon)\frac{P}{k}$	$\forall i \in C^{\phi}$		$\sum_{j\in T_i^{\beta}} p_j x_{ij} \le (1+\epsilon)\frac{P}{k}$	$\forall i \in C^{final}$
	$\sum_{j\in T_i^{\beta}} p_j x_{ij} \ge y_i(1-\epsilon)\frac{P}{k}$	$\forall i \in C^{\phi}$		$\sum_{j\in T_i^{\beta}} p_j x_{ij} \ge (1-\epsilon)\frac{P}{k}$	$\forall i \in C^{final}$
	$0 \le x_{ij} \le y_i \le 1,$	$\forall i,j \in E^{\phi\beta}$		$x_{ij} \in \{0,\ 1\},$	$\forall i,j \in E^{IP}$

where **LP_KMED** and **IP_TP** return their variable values.

Algorithm 1: k-medians approximation using spectral selection

$C^{\phi} = \{\ i \mid \quad \text{urandom}(i) > \phi\}$;
$T_i^{\beta} = \{\ j \mid \quad d(t_i, t_j) < R(t_i, \beta P/k)\} \quad \forall i \in C^{\phi}$;
$E^{\phi\beta} = \{\ (i,j) \mid \quad i \in C^{\phi},\ j \in T_i^{\beta}\}$;
$\{x_{ij}\},\ \{y_i\}$ = solve($\mathbf{LP\_KMED}(E^{\phi\beta},\ k)$) ; `// solve using barrier method`
$B = (C^{\phi} \cup N,\ E^{\phi\beta} \mid \text{weight}(e_{ij}) = x_{ij})$;
$S_1^c, \ldots, S_k^c$ = spectral_partition(B) ; `// keep only the left vertex set - the centers`
$C^{final} = argmin_{i \in C^{\phi}}\{\ dist(t_i,\ mean(S_{\ell}^c)\ \} \quad \forall \ell \in [1:k]$;
$E^{IP} = \{x_{ij} |\ i \in C^{final},\ j \in T_i^{\beta}\}$;
return solve($\mathbf{IP\_TP}(E^{IP})$) ; `// district i is comprised of tracts` j `where` x_{ij} `= 1`

4 Results

To test our algorithm, we chose to use the exact distribution of problems we were designing for - American states with k set to the current number of congressional, state senate, and state assembly representatives. The atomic geographic units are the from the TIGER shapefiles used by the census to store census tracts [12]. Additionally, data from the 2017 American Community Survey was used to get population estimates for each census tract [13].

We performed our computational experiments using a Dell R620 with two Intel Xeon E5-2680 2.70GHz 8-core processors and 96GB of RAM. For software, Gurobi v9.0.1 was used to solve the k-medians relaxation using the barrier method, and to solve the transportation problem integer program. We used Sklearn for the spectral partitioning implementation and NetworkX for managing graphs. Code, data, and results are made publicly available [4].

In our experiments we did a random order grid search over the following parameter space:

- Population factor pruning radius - $\beta \in [2, 4, 6, 8]$
- Random center pruning ratio - $\phi \in [0, 0.5, 0.75]$
- Cost exponential - $\alpha \in [1, 1.5, 2]$
- Population tolerance - $\epsilon \in [0.01, 0.025, 0.05]$

Furthermore, we only ran a (state, k) trial if the number of census tracts in the state divided by k was greater than 75. This gave us 49 (state, k) combinations and with a parameter space of 108 combinations that is 5292 trials in total. In our analyses we make the distinction between big, medium, and small problems where $k \geq 20$, $5 < k < 20$, $k \leq 5$ respectively.

Finally, as a baseline we also try solving the transportation problem with centers selected from a k-means heuristic. That is we input each tract with its centroid location and population and compute a k-means approximation using Lloyd's algorithm where points are weighted by their population. Like before, we then find the tracts that are closest to the exact centers and solve the transportation problem with these centers. Since this is fast, we perform it multiple times (uniformly at random between 1 and 20) and take the best solution.

4.1 Feasibility

The first important analysis is understanding feasibility. Too aggressive x_{ij} pruning for large problems will result in a search radius that is too small and will be infeasible. The worst case is when optimal centers are near dense urban areas. These are centers that probably won't have any assignment from the city tracts but because they are so close to the city, the available assignment radius is too small for it to reach tracts that should be assigned to it in the optimal solution.

Population tolerance also plays a role in feasibility because for a state that barely makes our cutoff, there might only be about 75 tracts per district which could be a problem when you have a 1% tolerance. However, in general a stricter tolerance only forces minor perturbations of the boundary tracts to make sure everything is balanced. Table 1 shows how feasibility for the transportation problem integer program varies with population tolerance ϵ and the pruning radius parameter β for both our spectral selection approach and the k-means baseline.

[4] `https://github.com/wesg52/Kmedians`

Spectral selection				
$\epsilon,\ \beta$	2	4	6	8
0.01	38.83	79.63	84.03	94.13
0.025	43.12	81.51	87.99	95.00
0.05	43.78	84.5	90.60	96.09

k-means selection				
$\epsilon,\ \beta$	2	4	6	8
0.01	16.76	83.33	90.91	93.37
0.025	18.52	81.51	90.08	93.50
0.05	18.91	83.98	92.17	94.27

Table 1: Percent feasible given population tolerance and distance pruning

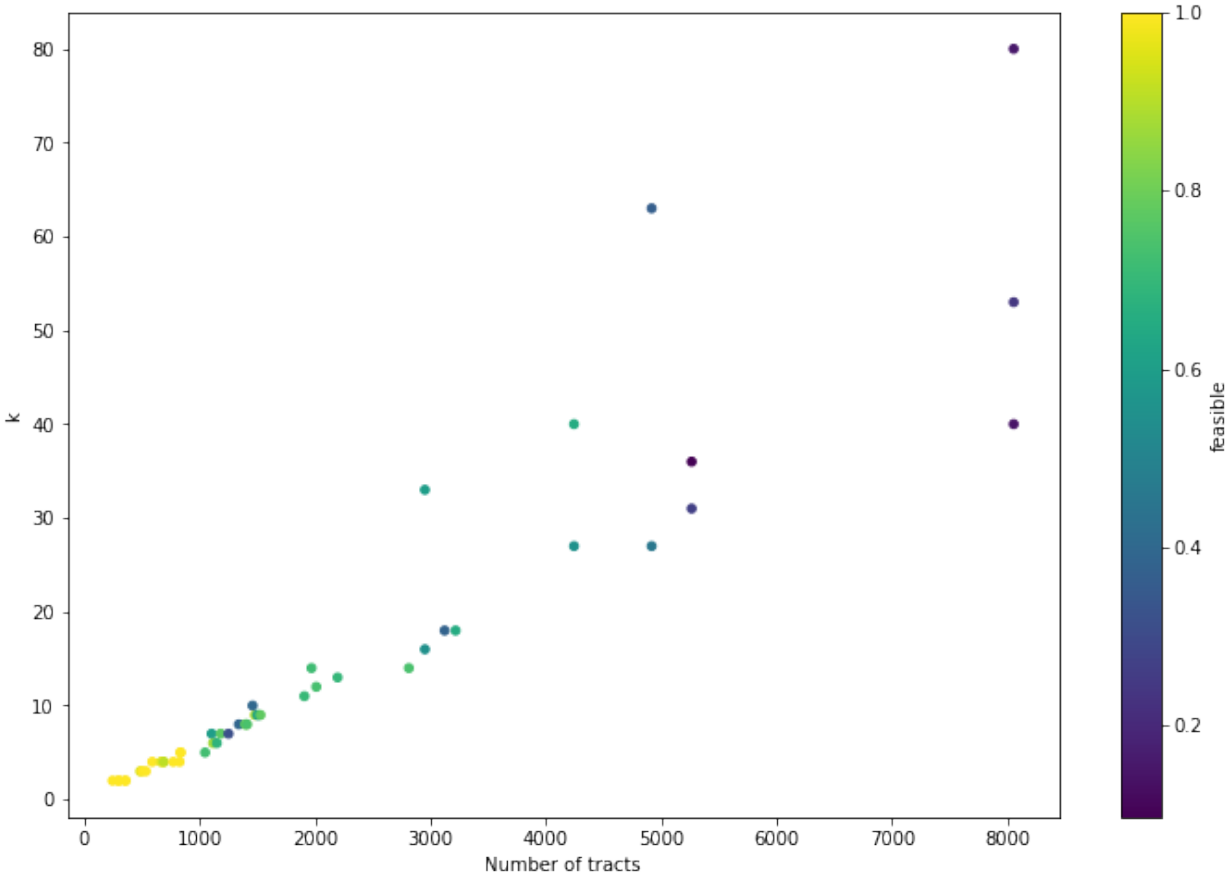

Figure 2: Percent feasible given number of tracts and number of districts

As one might expect, larger states are infeasible the most as shown by figure 2. In particular, they are basically always infeasible with pruning radius less than 6. Only states with less than about 4 districts are feasible with radius 2, the lowest pruning radius we tried. Unfortunately, one has to solve the full LP relaxation (usually >95% of the total run time cost) before we can know the feasibility of the IP and therefore one would want to dynamically set the tolerance conservatively. Since the k-means feasibility results are similar and much faster to compute, finding β with a feasible k-means solution could serve as a good approximation.

Lastly to help population tolerance feasibility issues one should use a smaller atomic geographic unit. The census also offers census blocks which are of subtract granularity with population estimates. For smaller states this is necessary to compute districts for state legislatures, but the scaling properties should all be the same.

4.2 Barrier Convergence

As was discusses earlier, we use the barrier method, an interior point, method because it is much faster to converge and benefits from parallelism. Unlike simplex, which has a very large number of very fast iterations, barrier has very few iterations, each of which can take on the order of seconds. In figure 3 we plot the cumulative counts of when the barrier method passes a primal-dual gap threshold for small, medium, and large problems ($k \leq 5$, $5 < k < 20$, $k \geq 20$).

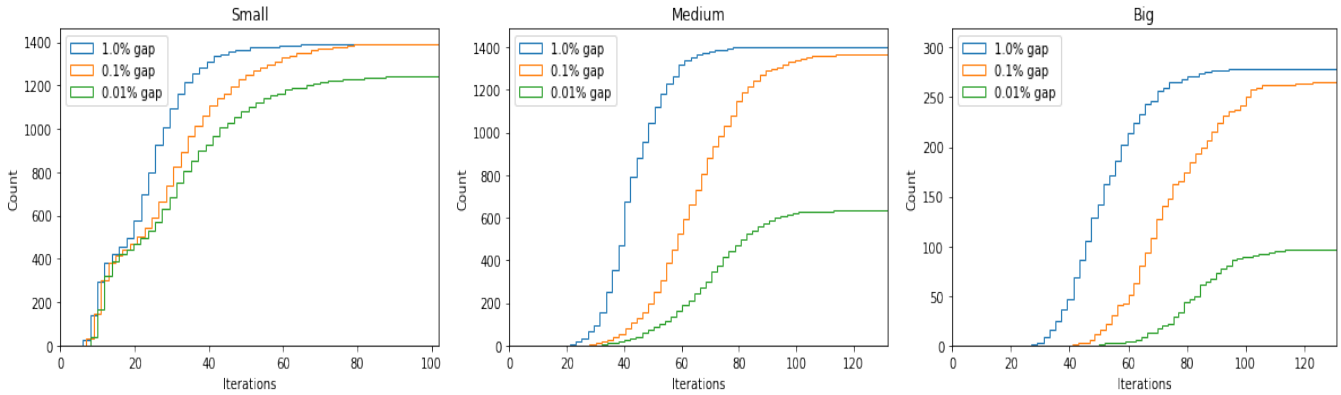

Figure 3: Cumulative counts of barrier method primal-dual threshold gap by problem size

It is important to note that for every experiment we set the barrier solver to stop if it reached a solution with gap less that 0.01% or 30 minutes elapsed, whichever comes first. Since the LP is only used for finding good centers, it isn't necessary to solve to as small a gap as we did, and one could save a lot of time in the process. However, we noticed that the clusters aren't as tight when the gap is larger, and therefore the quality of the centers produced by spectral selection might deteriorate. A line of future work is to investigate this tradeoff.

4.3 Objective Gaps

Our most exciting result is the performance of our spectral selection method as judged by the gap between the best integer solution and the best fractional solution, especially compared to the k-means selection baseline.

gap	total	β=2	β=4	β=6	β=8	ϕ=0.0	ϕ=0.5	ϕ=0.75	ϵ=0.01	ϵ=0.025	ϵ=0.05	big	medium	small
0.1	34.83	59.83	35.07	30.02	28.35	35.73	35.51	33.33	35.15	34.53	34.82	0.0	7.38	71.09
1.0	51.92	79.71	50.27	47.66	45.35	52.82	52.03	50.98	50.6	52.64	52.48	0.24	28.86	88.32
5.0	72.64	92.68	73.43	68.71	66.99	72.51	72.71	72.68	70.38	72.42	75.0	28.1	63.95	92.99
20.0	94.89	98.74	95.01	94.74	93.29	94.19	95.0	95.45	93.7	94.3	96.62	81.43	94.55	98.8
50.0	99.86	100.0	100.0	100.0	99.55	99.57	100.0	100.0	99.83	99.83	99.92	98.81	100.0	100.0

Table 2: Percent trials below IP-LP gap using spectral selection

gap	total	β=2	β=4	β=6	β=8	ϕ=0.0	ϕ=0.5	ϕ=0.75	ϵ=0.01	ϵ=0.025	ϵ=0.05	big	medium	small
0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.16	1.6	0.23	0.0	0.0	0.0	0.0	0.48	0.19	0.19	0.09	0.0	0.0	0.36
5.0	4.69	26.74	3.58	3.37	2.97	3.91	4.13	6.03	3.99	4.48	5.59	0.0	0.0	10.5
20.0	23.5	64.71	22.26	21.86	18.85	23.83	22.74	23.95	22.18	23.27	25.02	0.0	11.87	40.47
50.0	63.38	97.86	67.13	63.13	54.6	62.6	63.72	63.79	60.8	64.17	65.12	26.97	56.4	78.43

Table 3: Percent trials below IP-LP gap using k-means selection

However these results come with a few remarks. The first is that the gap is with respect to the pruned IP and the pruned LP and not with respect to a global LP bound. This was done to preserve consistency and make comparisons fair, but also because it would be cost prohibitive to find a global LP bound for some medium and all large problems. That being said, because the x_{ij} distance pruning is based on distance, which is precisely the objective function, it is unlikely that after increasing β well past the point of feasibility there will exist better integer solutions.

The second is that clearly spectral selection does not achieve small gaps on larger problems. However, because the integrality gap of the capacitated k-median problem is unbounded, it is not clear how well we could be doing. That is to say, this is the gap of our solution and the theoretically best fractional solution rather than the theoretically best integer solution. Since it is cost prohibitive to compute the optimal integer solution for large problems, it is difficult to know whether the integrality gap is to blame or if the spectral centers are just poorly chosen, perhaps because of poor spectral cluster density.

4.4 Run time

The last aspect we analyze is overall execution time. Figure 4 depicts the average contribution of each stage of the algorithm. Preprocessing here is just the time it takes to make the linear program. The MIP timeslice captures both creating and solving the MIP as it is such a small component. From figure 4 it is obvious the importance of reducing the runtime of barrier. It also shows that are pruning techniques were helpful in this regard. This is an advantageous place to be algorithmically speaking, because barrier is parallelizable therefore by Amdahl's law we can speed up execution time significantly with more CPU cores.

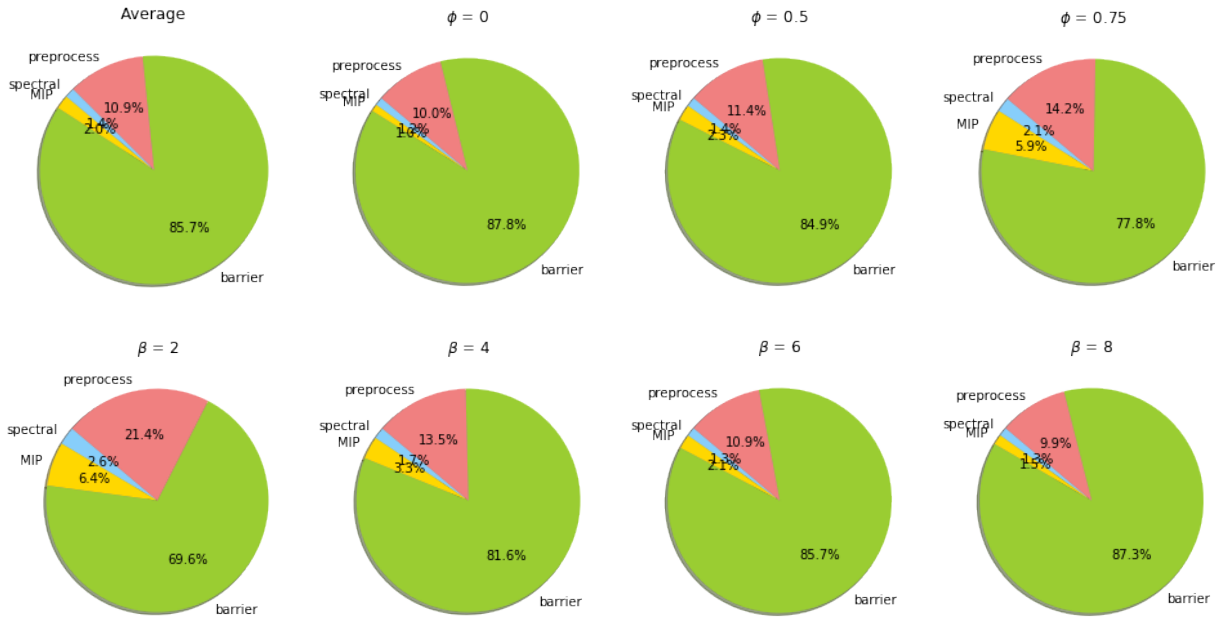

Figure 4: Total running time broken down by stage and pruning

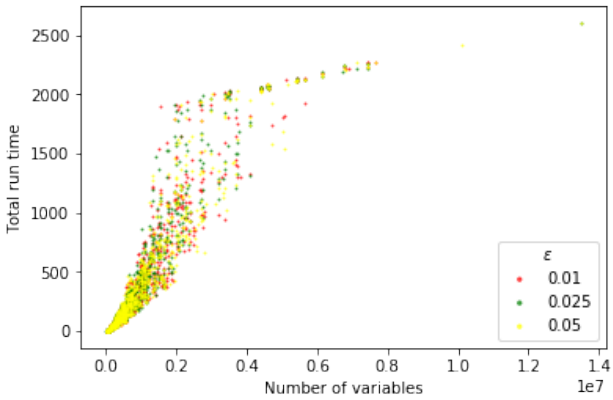

Figure 5: Total runtime (s) vs number of variables by ϵ

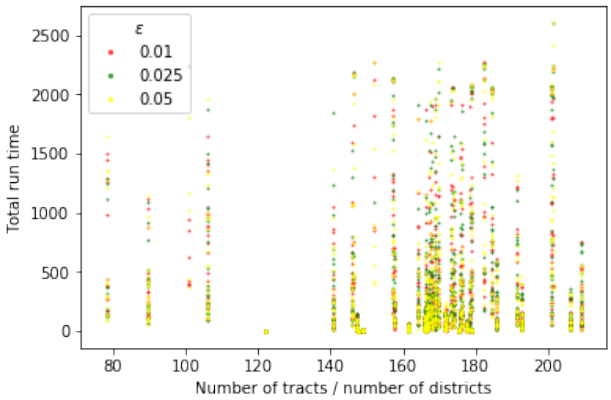

Figure 6: Total runtime (s) vs (n_tracts/k) by ϵ

Figures 5 and 6 depict how the runtime is affected by the total number of variables and the ratios of tracts to districts. Number of variables, as expected, has a very tight correlation with the total runtime. We expected there to be some difference in the ease of solution based on the ratio of y_i to x_{ij} variables but none existed. Problems with less restrictive population tolerances appear slightly easier to solve but only barely. Furthermore there also isn't a strong correlation between the tract to district ratio on runtime.

5 Conclusion and Future Work

In this work we presented a novel method for finding high quality capacitated k-medians solutions motivated by a political districting setting. Our approach was based on efficiently solving the linear relaxation to create an intermediate structure that admitted a natural partition found via spectral partitioning. The spectral partition was used to pick final centers to solve a polynomial time transportation problem, rather than an NP-hard k-median problem.

We investigated different pruning techniques to speed up the linear program without affecting solution quality. We presented a significant empirical study of our algorithm, detailing the behaviours of feasibility, convergence, objective gaps, and execution time. The purpose of our analysis was not to design the fastest possible algorithm with the smallest possible gap but provide an understanding of how these parameters effect performance so that we have the intuition to design such an algorithm.

Future work includes how to scale our approach to the largest states. One promising approach is nested design where we use an outer loop to split a large state into a few smaller pieces, and then optimize over the pieces. Another line of inquiry that could speed up solving the LP is early stopping but we need to study how early stopping affects the spectral selection process. As mentioned earlier, dynamic distance pruning based on k-means results would also be useful to avoid solving expensive relaxations that turn out to be integer infeasible.

Acknowledgements

Thank you to David Shmoys for being my brainstorming partner. Thank you to Peter Frazier for graciously letting us use his cluster for our computational experiments. Thank you to Bobby Klienberg for drawing the hour glass on the board during the second spectral methods lecture which inspired the spectral selection idea.

References

[1] H Peyton Young. Measuring the compactness of legislative districts. *Legislative Studies Quarterly*, 13(1):105–115, 1988.

[2] Jowei Chen, Jonathan Rodden, et al. Unintentional gerrymandering: Political geography and electoral bias in legislatures. *Quarterly Journal of Political Science*, 8(3):239–269, 2013.

[3] Khoa Trinh. A survey of algorithms for capacitated k-median problems.

[4] Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.

[5] Kamal Jain and Vijay V Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001.

[6] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.

[7] Aaron Archer, Ranjithkumar Rajagopalan, and David B Shmoys. Lagrangian relaxation for the k-median problem: new insights and continuity properties. In *European Symposium on Algorithms*, pages 31–42. Springer, 2003.

[8] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

[9] Jarosław Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 722–736. SIAM, 2014.

[10] Shi Li. On uniform capacitated k-median beyond the natural lp relaxation. *ACM Transactions on Algorithms (TALG)*, 13(2):22, 2017.

[11] Juan Carlos Duque, Raúl Ramos, and Jordi Suriñach. Supervised regionalization methods: A survey. *International Regional Science Review*, 30(3):195–220, 2007.

[12] census.gov. Tiger/line shapefiles, 2019.

[13] census.gov. American fact finder, 2019.