Placing Voting Centers in Pittsburgh

Deborah Blank, Ally Meringer, Zara Zetlin

21-393, Fall 2020

1 Introduction

1.1 Project Motivation

Voting is a key aspect of American governance. However, despite its significance and relatively high voter registration rates, voter turnout in the United States is low compared to other countries. This prompts a pressing question: why don't Americans go to the polls on election day?

One possible reason for low voter turnout rates is inaccessible polling locations. In particular, this project focuses on assigning polling locations to geographic regions to ensure that voting is convenient and accessible. Ultimately, the objective of the project is to maximize voter turnout. However, voters decide whether they cast their ballot or not. In effect, this report is written from the perspective of the government, which cannot coerce citizens to vote (although other countries have implemented compulsory voting laws). As a heuristic for measuring voter turnout, this report measures the success of different voting location configurations according to a number of metrics. The subsequent methods discussed in this report optimize over these location configurations.

1.2 Problem Context and Data Intuition

The city of Pittsburgh, PA, home of the Carnegie Mellon University Tartans, is organized into 32 wards, with 402 districts across these wards. Each ward is composed of at least one district, with each district housing one, or sometimes two, voting centers. In the majority of this project, we will assume just one voting center per district. Given this framework, this report explores how the 402 voting centers (districts) ought to be allocated across Pittsburgh's 32 wards. Later, in the last section of the report, we will relax the assumption that there is one voting center per district and we will consider adding more than one voting center per district. Locations where multiple voting centers per district may occur will be informed by our ideal model and the estimated cost of opening a polling location in that area.

Data sets regarding district population, poverty levels, demographics, median household income, and more are available through the United States Census and state of Pennsylvania records. For more information on the source of the data used in this report, see Section 7, References.

These data sets were used to inform the models presented in this project. In particular, this data was critical for determining which regions need voting centers. For instance, a large population may warrant more districts because this would ensure that no single voting center becomes overcrowded. Wards with lower median household income may be assigned more districts because election day is during the business week, and these citizens are less likely to take time off of work to vote. Ultimately, this project sought to leverage large and relevant data sets, and to combine this information with optimization techniques.

1.3 Project Objective

Given that there are a fixed number of wards (32), and a fixed number of districts (402), how should voting centers be assigned to wards? This report approaches the problem in stages, with each stage adding a new layer of analysis. Section 2 outlines how to find a potential optimal district configuration. Section 3 supposes that no new polling stations may be added, and considers how to balance the hassles of change and a suboptimal voting location configuration. Section 4 discusses Pittsburgh's potential to add new voting locations to expand polling accessibility. Such an approach is nontrivial because it involves translating data into a meaningful mathematical representation, formulating multiple optimization subproblems, and leveraging state-of-the-art software (Gurobi).

2 Finding An Optimal District Configuration

2.1 Problem Statement

There are a plethora of different metrics that could be used to assign districts to wards. Several of these metrics are considered in this report, and inform a proposed district configuration. More specifically, given data on population, median household income, and ward demographics, how can an optimal district configuration be found?

Even in modern elections, people of color face voter suppression that makes it more difficult to have their voices heard. In addition, those living below the poverty line face challenges to arrive at polling locations due to cost of transportation and other factors.

We created several models that take the population of a model to be the combination of the population of residents, population of non-white residents, and population of those living below the poverty line.

Our final model, included all these populations and, therefore, non-white residents living below the poverty line were counted 3 times. While, this might seem extreme, the purpose of Section 3 and 4 of this report (post processing) is to use cost to find a middle point between this model and the model used today.

2.2 Mathematical Model

In order to create models that gave heavier weights to areas with marginalized groups, we took the census data and aggregated it to each ward using the geographical location of that census block and the geographical locations of the the wards in Pittsburgh.

After, we entered this information into our dynamic programming algorithm. The size of the memorizer matrix was number of wards $\times$ maximum districts to be distributed per ward $\times$ total districts to be distributed per ward because the tuples entered into the program were (ward, number of districts, number of districts left to be placed in wards). In this model, the size of the memorizer was $32 \times 370 \times 57 = 674,880$.

The purpose of the 3 dimenational matrix, or memorizer, is to store outputs so we can call on values in the matrix and do not have to compute the value of a specific tuple more than once.

This gave us the code attached in the python file.

First, we will define the following variables for this problem,
The number of wards:
$$numWards = 32.$$

The total districts amongst all wards before reordering:

$$totalDistricts = 402.$$

The number of districts we are distributing which is the total minus the number of wards because we are assuming they all have 1:

$$capDistricts = int(totalDistricts - numWards).$$

The max districts a ward can have which helps with the run time:

$$maxDistricts = int(capDistricts * 5/numWards).$$

$f_w(d, c) =$
$max((popData[w] + pocData[w] + povData[w])/(d + 1)) + f_w(d + 1, c - 1), f_{w+1}(0, c))$
$f_w(d, 0) = 0$
$f_w(numDistricts, c) = 0$
$f_{numWards}(d, c) = 0$

We know that this is not trivial because we are not solely weighting by the largest populations. We divide that population by the number of districts that we have already assigned to that ward, so the more you allocate, the more that it lowers the value and we assign it to other wards.

2.3 Solution

The following models have the distribution of districts based on data from the census about
A : Population
B : Population living below the poverty line
C : Population and population living below the poverty line
D : Non-white population

E : Population and non-white population
F : Population, non-white population, and population living below the poverty line

Ward ↓, Model →	A	B	C	D	E	F
1	11	29	15	12	11	14
2	4	4	4	2	3	3
3	4	5	5	9	6	5
4	25	44	28	22	25	27
5	11	17	12	21	13	14
6	6	6	6	3	5	5
7	18	27	20	16	18	19
8	14	11	13	11	13	12
9	9	7	8	3	7	7
10	17	12	16	18	17	16
11	17	12	15	21	18	17
12	10	10	10	25	14	13
13	11	15	11	29	15	15
14	47	20	40	35	44	39
15	17	13	15	14	16	15
16	12	9	11	9	11	11
17	7	58	28	3	6	23
18	10	8	9	13	11	10
19	34	16	29	14	29	26
20	18	12	17	20	19	17
21	3	4	3	6	4	4
22	5	5	5	6	5	5
23	3	3	3	2	3	3
24	4	4	3	3	4	3
25	4	5	4	7	5	5
26	14	14	14	22	16	16
27	17	8	15	20	18	16
28	15	8	13	13	14	13
29	14	7	12	10	13	12
30	5	4	5	9	6	6
31	7	2	6	2	6	5
32	9	3	7	2	7	6

In the post processing phases, we used model F which includes the number of residents, number of non-white residents, and number of residents living below the poverty line.

3 Postprocessing: Managing the Cost of Change

3.1 Postprocessing Problem Statement

Once an optimal district configuration is found, how feasible is it to update district boundaries? There is a cost of maintaining a sub-optimal district configuration, but there is also a cost associated with changing these boundaries. In particular, many voters assume that their polling location has not changed, and go to the most recent place where they cast a ballot. As such, there is a cost incurred for each citizen whose polling location changes.

On one hand, the cost of maintaining sub-optimal district boundaries pushes the overall best configuration towards the proposed configuration discussed in Section 2. However, the cost of changing voting locations pushes the overall best configuration toward the current district boundaries. This subproblem addresses how the cost of both of these issues impact an overall optimal district configuration that takes both of these costs into consideration.

3.2 Postprocessing Mathematical Model

As discussed above, this report takes two costs into consideration:
Let P represent the cost of deviating from the proposed district configuration (Section 2).
Let C represent the cost of deviation from the current district configuration.
Then it is possible to calculate the fractional cost associated with each type of penalty. In particular, define

$$\gamma_P = \frac{P}{P+C},$$

the fractional cost of deviating from the proposed district configuration, and

$$\gamma_C = \frac{C}{P+C},$$

the fractional cost of deviating from the current district configuration. Note that

$$\gamma_P + \gamma_C = 1.$$

Also note that while these costs have an undefined range of potential values, the fractional costs can only take on values between 0 and 1. Because of this, the fractional cost approach was taken in this report to better illustrate the range of results that are possible within the framework of the proposed model.

Next, consider two pertinent models: the current district configuration, and the proposed district configuration from Section 2.
Let m_i represent the current number of districts in ward i, for $i = 1, 2, ..., 32$
Let n_i represent the proposed number of districts in ward i, for $i = 1, 2, ..., 32$
Notice that the m_i values are known from the data, and the n_i values are known from the output of Section 2.

Define x_i as the number of districts in ward i when taking deviation costs into account, for $i = 1, 2, ..., 32$.
Note that the x_i values represent decision variables. Ultimately, this step of the project aims to find an assignment of x_i variables that minimizes the deviation cost.

Next, move on to the constraints of this model.
Assume that there are still exactly 402 districts to be allocated. Written in a more mathematical language,

$$\sum_{i=1}^{32} x_i = 402.$$

Additionally, enforce the constraint that every ward must have at least one district:

$$x_i \geq 1 \quad \forall i = 1, 2, ..., 32.$$

Finally, no fractional districts may be allocated. This means that x_i is an integer quantity for $i = 1, 2, ..., 32$

Moving on to the objective, consider the following function:

$$\min \gamma_P \sum_{i=1}^{32} (n_i - x_i)^2 + \gamma_C \sum_{i=1}^{32} (m_i - x_i)^2.$$

Note that no error terms are negative, i.e.

$$(n_i - x_i)^2 \geq 0 \quad \text{and} \quad (m_i - x_i)^2 \geq 0.$$

Also note that this objective function penalizes deviation in a quadratic manner. This is a reasonable assumption because it penalizes a large deviation more than several smaller discrepancies. In effect, this ensures that no one ward is penalized heavily while all other wards are assigned only light costs.

Importantly, note that this objective is quadratic, not linear. This section of the report leverages state-of-the-art optimization software by using Gurobi Solver, which seamlessly manages such a nonlinear objective. However, note that this objective could alternatively be expressed as a semidefinite program. Because an SDP formulation is not needed to run this program, such a formulation is omitted from the report.

Altogether, the optimization formulation used in this section of the report is:

$$\min \quad \gamma_P \sum_{i=1}^{32} (n_i - x_i)^2 + \gamma_C \sum_{i=1}^{32} (m_i - x_i)^2$$

$$\text{subject to} \quad \sum_{i=1}^{32} x_i = 402$$

$$x_i \geq 1 \quad \forall i \in [32]$$

$$x_i \in \mathbb{Z}.$$

3.3 Postprocessing Solution

This proposed optimization problem was approached using Gurobi Solver. In particular, Gurobi was used in conjunction with the programming language Julia. The resulting code read in the output from the previous section of this report and the district data, then implemented the suggested program.

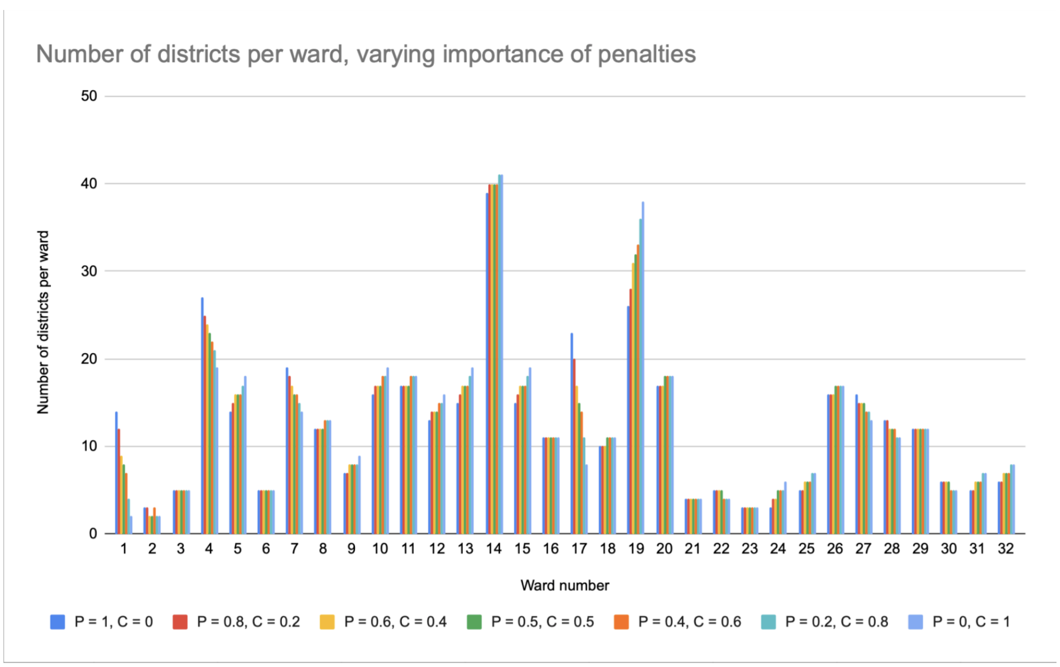
Although the software managed the optimization itself, there are several limitations to the proposed formulation. First of all, finding fractional costs γ_P, γ_C is not straightforward. Polling data primarily focuses on voter registration, turnout, demographics, and party affiliation. As such, data regarding these costs is sparse, and cannot be easily estimated. However, by considering fractional weights rather than absolute costs, the range of potential outputs is limited and easier to interpret.

Additionally, it is unclear how increases in deviation from the current or proposed district schemes scale. Is it a quadratic relationship, as assumed in the objective function? Once again, because data on these latent costs of voting inefficiencies are not well documented, it is difficult to find a more precise measure.

Relaxing these concerns for the time being and solving the proposed optimization problem does yield reasonable results. In particular, when $\gamma_P = 1$ and $\gamma_C = 0$, the program returns the proposed district allocation suggestion. This is intuitive because when only the cost of a suboptimal configuration is considered, the code ought to suggest the proposed optimal output.

Similarly, when $\gamma_C = 1$ and $\gamma_P = 0$, the program returns the current district configuration. This is also intuitive, because when only the cost of change is considered, the code ought to suggest making no changes to district allocation and preserving the current configuration.

In addition to these edge cases, the code was also run with a variety of (P, C) tuples, namely $(0.2, 0.8), (0.4, 0.6), (0.5, 0.5), (0.6, 0.4), (0.8, 0.2)$. The output of these trials is plotted in the figure below:



In particular, note that some wards have a very consistent number of districts across (P, C) tuples. Such wards include 3, 6, 16, 21, 23, and 29. Looking at the data, these wards have similar values across population, poverty, and POC metrics. Therefore, it is unsurprising that the proposed configuration from Section 2 of this report suggested a similar number of districts compared to the current configuration.

In contrast, some wards could have a large range of districts, depending on the (P, C) tuple values. Some such wards are 1, 4, 7, 17, and 19. Looking at the data, these wards have very different values across population, poverty and POC metrics. Consequently, it is unsurprising that the proposed configuration from Section 2 did not suggest a similar number of districts compared to the current configuration, because the model from Section 2 specifically prioritized the poverty and POC metrics.

4 Voting Locations Investigation

4.1 Introduction: Voting Locations

Thus far, in determining our redistricting, we have used the assumption that each of the 402 Pittsburgh districts have one voting location. The only data source listing the voting locations per district in Pittsburgh is from the 2015 elections. Though this data is not the most up-to-date and reliable, we will use it to add valuable insights and depth into our redistricting model to determine how Pittsburgh's voting locations should be allocated within the 32 wards.

The November 2015 Pittsburgh elections had 598 polling locations in total. It costs the city a considerable amount of money to rent out all 598 locations. We aimed to minimize the estimated costs to rent each of the polling centers we recommend.

In general, we wanted to extend the redistricting model to make voting more accessible to marginalized groups. While limiting the total cost to rent these polling locations, we aimed at ensuring there were not too many or too few people at a given polling location. This would help ensure that the lines at a given polling center are not too long, and that polling centers are not too far away from another to make them more accessible to voters.

4.2 Problem Statement: Voting Locations

We set that each voting location has to cover at least 500 citizens, and no more than 2000 citizens. Additionally, that there will not be less than the current 598 voting locations, so that voting would only become more accessible and there would not be as much change for voters.

This problem is nontrivial as it takes into consideration multiple factors to find the optimal distribution of voting centers per ward. The metrics we used are a calculation of a cost estimate for renting a polling facility per ward and the population size in each ward (from the 2010 Census data). We used a linear program to calculate how many polling centers should be in each ward, and we then used a Greedy Algorithm to find integer solutions.

We also found an IP solution to the number of voting centers per district, and we discuss the differences in the two results.

4.3 Mathematial Model: Voting Locations

To calculate the estimated cost of renting a polling center in each ward, we averaged the median household value and the median contract rent per ward as determined in the 2010 Census. We will denote this cost to rent a polling center in each ward as C, where C_i is the cost of renting a polling center in ward i for $i = 1, 2, ..., 32$.

The 2010 U.S. Census data for population per ward will be denoted as P, where P_i is the population of ward i for $i = 1, 2, ..., 32$.

In this representation, we are calculating the number of polling locations per ward which we will label as Y, where Y_i is the number of polling centers we recommend for ward i for $i = 1, 2, ..., 32$.

We set that each voting location has to cover at least 500 citizens, and no more than 2000 citizens. In other words, the population of citizens per ward P_i is split into a number of voting centers Y_i such that there are between 500-2000 people assigned to each polling location.

$$P_i/Y_i \geq 500, \qquad P_i/Y_i \leq 2100, \qquad \text{for} \quad i = 1, 2, ..., 32.$$

$\Longrightarrow$

$$Y_i/P_i \leq 1/500, \qquad Y_i/P_i \geq 1/2000, \qquad \text{for} \quad i = 1, 2, ..., 32.$$

We used the inverse in order to avoid division by zero when calculating Y_i.

In the 2010 election there were 598 voting locations within the 32 wards in Pittsburgh, so we also wanted to ensure that the total number of polling locations we are suggesting is at least as many as before, ie.

$$\sum_{i=1}^{32} Y_i \geq 598.$$

As a whole, our LP is:

$$\begin{aligned} \min \qquad & \sum_{i=1}^{32} C_i * Y_i \\ st \qquad & Y_i/P_i \leq 1/500 \qquad \text{for} \quad i = 1, 2, ..., 32 \\ & Y_i/P_i \geq 1/2000 \qquad \text{for} \quad i = 1, 2, ..., 32 \end{aligned}$$

$$\sum_{i=1}^{32} Y_i \geq 598 \qquad \text{for} \quad i = 1, 2, ..., 32.$$

4.4 Solution: Voting Locations

We solved the above LP to calculate Y, the number of polling locations needed per ward, with Excel's Simplex LP Solver. This gave us the result seen below. The 'New Number of Districts' refers to our solution in our redistricting model in Section 2.

Ward	New Number of Districts	C = Cost to Rent a Polling Location ($)	P = Population	Y = Number of Polling Locations
1	14	557	8392	16.78
2	3	924	2394	4.79
3	5	357	3122	6.24
4	27	497	19477	38.95
5	14	323	8114	16.23
6	5	779	4080	8.16
7	19	1,079	13905	27.81
8	12	701	10235	20.47
9	7	685	6464	12.93
10	16	395	12839	25.68
11	17	731	12757	25.51
12	13	187	7209	14.42
13	15	178	8014	16.03
14	39	1,183	36115	69.37
15	15	416	12660	25.32
16	11	552	8879	17.76
17	23	616	5084	10.17
18	10	210	7083	14.17
19	26	353	25824	51.65
20	17	314	13951	27.90
21	4	490	2208	4.42
22	5	796	3793	7.59
23	3	637	1991	3.98
24	3	230	2394	4.79
25	5	477	2928	5.86
26	16	268	10811	21.62
27	16	343	12909	25.82
28	13	322	11132	22.26
29	12	242	10822	21.64
30	6	156	3709	7.42
31	5	320	4822	9.64
32	6	314	6312	12.62

This result's objective function outputs the cost to rent these voting locations to be

$327,444. This solution has a total of 598 voting centers. This is the minimum number of voting centers, which is the amount used in 2015.

An integer solution for the number of polling centers per district is needed since there can't be a fractional number of polling centers. To do so, we implemented a Greedy Algorithm to find integral number of voting locations for each of our new districts in every ward. This gave us that per district in each ward we should have a mixed number of polling centers:

Ward ▲	New Number of Districts	Y = Number of Polling Locations	Integer Y	Breakdown	
1	14	16.78	16	12 districts with 1 polling center	2 districts with 2 polling centers
2	3	4.79	5	2 districts with 2 polling centers	1 district with 1 polling center
3	5	6.24	6	4 districts with 1 polling center	1 district with 2 polling centers
4	27	38.95	38	16 districts with 1 polling center	11 districts with 2 polling centers
5	14	16.23	16	12 districts with 1 polling center	2 districts with 2 polling centers
6	5	8.16	8	3 districts with 2 polling centers	2 districts with 1 polling center
7	19	27.81	27	11 districts with 1 polling center	8 districts with 2 polling centers
8	12	20.47	20	8 districts with 2 polling centers	4 districts with 1 polling center
9	7	12.93	13	6 districts with 2 polling centers	1 district with 1 polling center
10	16	25.68	26	10 districts with 2 polling centers	6 districts with 1 polling center
11	17	25.51	26	9 districts with 2 polling centers	8 districts with 1 polling center
12	13	14.42	14	12 districts with 1 polling center	1 district with 2 polling centers
13	15	16.03	16	14 districts with 1 polling center	1 district with 2 polling centers
14	39	69.37	71	32 districts with 2 polling centers	7 districts with 1 polling center
15	15	25.32	25	10 districts with 2 polling centers	5 districts with 1 polling center
16	11	17.76	18	7 districts with 2 polling centers	4 districts with 1 polling center
17	23	10.17	23	23 districts with 1 polling center	
18	10	14.17	10	10 districts with 1 polling center	
19	26	51.65	51	25 districts with 2 polling centers	1 district with 1 polling center
20	17	27.90	27	10 districts with 2 polling centers	7 districts with 1 polling center
21	4	4.42	4	4 districts with 1 polling center	
22	5	7.59	7	2 districts with 2 polling centers	3 districts with 1 polling center
23	3	3.98	3	3 districts with 1 polling center	
24	3	4.79	4	1 district with 2 polling centers	2 districts with 1 polling center
25	5	5.86	5	5 districts with 1 polling center	
26	16	21.62	21	11 districts with 1 polling center	5 districts with 2 polling centers
27	16	25.82	26	10 districts with 2 polling centers	6 districts with 1 polling center
28	13	22.26	22	9 districts with 2 polling centers	4 districts with 1 polling center
29	12	21.64	22	10 districts with 2 polling centers	2 districts with 1 polling center
30	6	7.42	6	6 districts with 1 polling center	
31	5	9.64	10	5 districts with 2 polling centers	
32	6	12.62	12	6 districts with 2 polling centers	

In the above table, we kept the integer values of the number of polling locations to sum to 598, the total number in our solution. Looking at a given ward, the section titled 'Breakdown' shows that the sum of the districts is the total 'New Number of Districts'. Additionally, the total number of polling centers per ward within the breakdown (eg. for Ward 1: 12 districts * 1 polling center + 2 districts * 2 polling center) is equal to the 'Integer Y' value.

Alternatively, adding integer constraints to get an IP to work on this problem is feasible if the population per polling location is reduced further to a minimum of 450 voters and a maximum of 1,000 voters at each polling location. This also further optimizes the objective function and hence cost for Pittsburgh's voting centers to be $291,448. This IP also has the number of voting centers at 598, the minimum necessary to maintain the number of polling centers used in the 2015 election. Therefore, as this IP reduces the cost to rent out Pittsburgh's voting centers and as it still maintains the necessary 598 locations, it is the preferred solution.

Ward ▲	New # of Districts	C = Cost to Rent a Polling Location ($)	P = Population	Y = # of Polling Locations
1	14	557	8392	18
2	3	924	2394	4
3	5	357	3122	6
4	27	497	19477	43
5	14	323	8114	18
6	5	779	4080	9
7	19	1,079	13905	14
8	12	701	10235	22
9	7	685	6464	14
10	16	395	12839	28
11	17	731	12757	28
12	13	187	7209	16
13	15	178	8014	17
14	39	1,183	36115	41
15	15	416	12660	28
16	11	552	8879	19
17	23	616	5084	11
18	10	210	7083	15
19	26	353	25824	57
20	17	314	13951	31
21	4	490	2208	4
22	5	796	3793	8
23	3	637	1991	4
24	3	230	2394	5
25	5	477	2928	6
26	16	268	10811	24
27	16	343	12909	28
28	13	322	11132	24
29	12	242	10822	24
30	6	156	3709	8
31	5	320	4822	10
32	6	314	6312	14

5 Conclusion

Finding an optimal district assignment is a challenging problem. Such an assignment depends heavily on available data, including Census data. It is important to note that this project was written in the final quarter of 2020, and that 2020 is a US Census year. However, the data from the 2020 Census had not yet been compiled, meaning that the data used in this project was relatively out-of-date. Perhaps re-optimizing over the 2020 data using the same techniques would give more insight into an optimal district assignment that reflects the present day, rather than an optimal assignment built around data from the past several years.

Of course, any data set, no matter how recent, may not be entirely accurate. Not every resident completes the Census, and people may move across state and international borders immediately after the Census is conducted. In short, although it may be possible to improve the data estimates by waiting for the 2020 Census results, this will not eliminate all uncertainty from the data.

Of course, there are alternate formulations to the methods proposed in this report. Although each section of the report suggests a formulation and argues that it is a reasonable approach, alternate formulations are valid as well. Unfortunately, some voting mechanics are not known with certainty and therefore must be estimated in the project.

However, despite these challenges, the realized results do appear reasonable. The proposed configuration in Section 2 indeed emphasizes the importance of historically-marginalized groups. The cost analysis discussed in Section 4 succeeds in finding a balance between cost and voting accessibility. The postprocessing step illustrates the trade off between change and a suboptimal allocation of voting locations.

Although this model only focuses on Pittsburgh, similar optimization techniques could be used to process data from a larger geographical region. Of course, given the Coronavirus pandemic, absentee ballots and early voting have become increasingly popular alternatives to going to the polls on election day. This project could be improved by taking these recent trends, along with a larger geographical area and updated data, into consideration.

Given the spike in voter turnout during the 2020 election along with increasing political tensions, voting is coming to the forefront of political engagement. Voting accessibility is an extremely relevant topic that affects hundreds of millions of Americans. As such, it is in the best interest of the American people to capitalize on available data and to leverage known optimization techniques.

6 Acknowledgments

We would like to thank Professor Jeff Zhang for sharing his passion for Mathematics and for providing his support and insight throughout this project and the course Operations Research 2 (21-393).

7 References

1. https://www.alleghenycounty.us/uploadedFiles/Allegheny_Home/Dept-Content/Elections/Docs/8th%20Pittsburgh%20Council%20Polling%20Place%20Addresses.pdf

2. https://apps.alleghenycounty.us/website/MuniPgh.asp/

3. https://pittsburghpa.maps.arcgis.com/apps/OnePane/basicviewer/index.html?appid=2a57c4fbe92248e38a57220d20f23ae2

4. https://www.city-data.com/city/Pittsburgh-Pennsylvania.html

5. https://apps.pittsburghpa.gov/redtail/images/10892_2020_Council_Districts___Wards_Revised.pdf

6. https://www.publicsource.org/biden-trump-allegheny-county-pittsburgh-vote-breakdown/

7. https://openac-alcogis.opendata.arcgis.com/datasets/faaf42d7eaa041cb9fa623ac7b42f475_0/data?geometry=-82.481%2C40.067%2C-77.559%2C40.799