Hungry for Equality: Fighting Food Deserts*

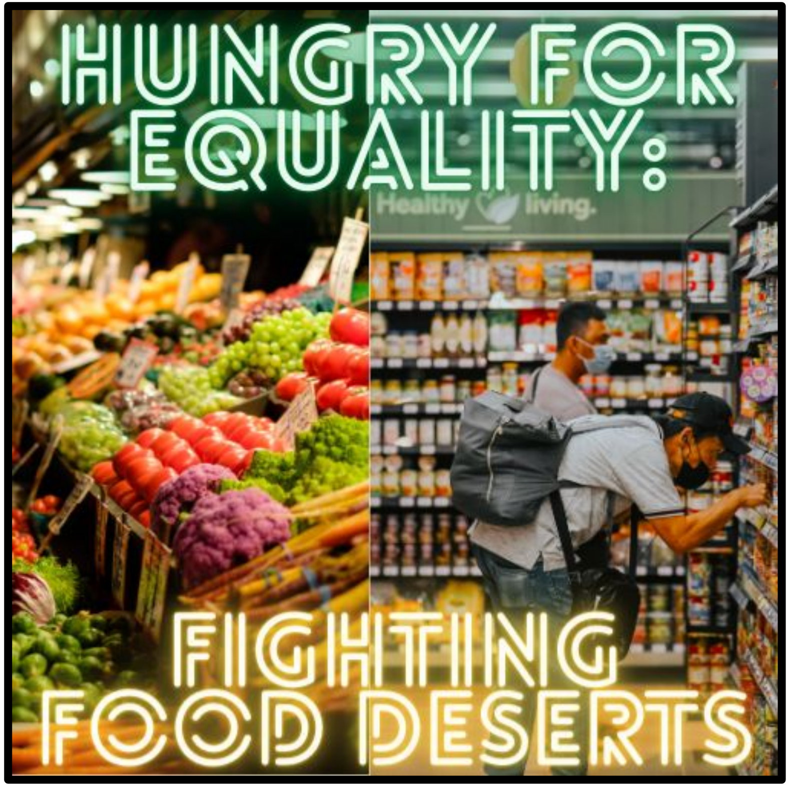
•Drew Horton // Rebecca Robinson•

Math 7594 Final Project

*food desert is a term defined by the USDA that brings focus solely on proximity to food providers, rather than other factors such as racial inequity, poverty, and more.

Project Overview

- A food desert in an urban area is defined to be a region more than 1 mile away from the nearest grocery store
- We want to reduce the number of regions in Denver that are considered to be in a food desert by finding optimal locations to open grocery stores
- Gathered potential grocery store locations from warehouses big enough that are for sale or lease
- We use two different programs to decide which locations would be best to open stores at



Motivation

- The COVID-19 pandemic has had a severe impact on food insecurity
- 32.3% increase in food insecurity since the pandemic began
    - 35.5% of food insecure households considered newly food insecure
- CDC research suggests that higher rates of COVID-19 infections among those in the Latinx and Black populations could be influenced by social determinants of food insecurity and by the lack of nutritional quality of food in food deserts

From a policy perspective:

- We hope to encourage policy makers to consider intervention strategies that prioritize relief in disproportionately affected communities
- This could be by opening grocery stores in an optimal distribution, or by utilizing more food pantries or community fridges.

The First of Two Programs

- Minimizes the total (Manhattan) distance, traveled to the closest grocery store, factoring in population
- Constraints ensure that:
    - Everybody gets assigned a store to go to
    - We don't exceed the number of stores we have
    - Only get assigned a store if its open

$$
\begin{aligned}
\min \quad & \sum_{r \in R} p_r x_r \\
\text{subject to} \quad & x_r = \sum_{s \in S} d_{rs} y_{rs} \quad \forall r \in R \\
& \sum_{s \in S} z_s \leq n \\
& y_{rs} \leq z_s \quad \forall r \in R, s \in S \\
& \sum_{s \in S} y_{rs} = 1 \quad \forall r \in R \\
& z_s, y_{rs} \in \{0,1\} \quad \forall r \in R, s \in S
\end{aligned}
$$

R : Residential areas

S : Stores

p_r : Population of residential area *r*

d_{rs} : Distance from residential area *r* to store *s*

n : Number of stores

y_{rs} : Indicator variable for if residential area *r* is assigned to store *s*

z_s : Indicator variable for if store *s* is used

Kolm-Pollak EDE

Equally Distributed Equivalent (EDE)

- Measure of central location of a distribution, penalized for inequality based on a social welfare function.
- Represents value which would, if everyone had that same value, provide the same level of welfare as the existing distribution.

Kolm-Pollak EDE

- Used in urban planning to rank distributions.
- Assigns a value, a measure of inequality, to a distribution.
- For optimization, we are given a description of a distribution, and we minimize over the measure of inequality represented by the Kolm-Pollak EDE.
- Constructed to evaluate both desirable and undesirable qualities

Kolm-Pollak EDE Formulation

$$
\begin{aligned}
\min \quad & -\frac{1}{\kappa}\ln\left[\frac{1}{|R|}\sum_{r\in R} e^{-\kappa x_r}\right] \\
\text{subject to} \quad & x_r = \sum_{s\in S} d_{rs} y_{rs} \quad \forall r \in R \\
& \sum_{s\in S} z_s \le n \\
& y_{rs} \le z_s \quad \forall r \in R, s \in S \\
& \sum_{s\in S} y_{rs} = 1 \quad \forall r \in R \\
& z_s, y_{rs} \in \{0,1\} \quad \forall r \in R, s \in S \\
& \kappa = \frac{\sum_{r\in R} x_r}{\sum_{r\in R} x_r^2}\beta
\end{aligned}
$$

R : Residential areas

S : Stores

p_r : Population of residential area r

d_{rs} : Distance from residential area r to store s

n : Number of stores

y_{rs} : Indicator variable for if residential area r is assigned to store s

z_s : Indicator variable for if store s is used

β : Aversion to inequality

Piecewise Linear Relaxation of Kolm-Pollak Formulation

$$
\begin{aligned}
\min \quad & w_r \\
\text{subject to} \quad & w_r \geq \alpha p_r(x_r + 1) \quad \forall r \in R \quad (\alpha = -\kappa) \\
& w_r \geq e^{\alpha p_r \frac{b_r}{1000}} (\alpha p_r(x - \frac{b_r}{1000} + 1) \quad \forall r \in R \\
& w_r \geq e^{\alpha p_r \frac{b_r}{100}} (\alpha p_r(x_r - \frac{b_r}{100} + 1) \quad \forall r \in R \\
& w_r \geq e^{\alpha p_r b_r} (\alpha p_r(x_r - b_r + 1) \quad \forall r \in R \\
& x_r = \sum_{s \in S} d_{rs} y_{rs} \quad \forall r \in R \\
& b_r = \max_{s \in S}\{d_{r,s}\} \quad \forall r \in R \\
& \sum_{s \in S} z_s \leq n \\
& y_{rs} \leq z_s \quad \forall r \in R, s \in S \\
& \sum_{s \in S} y_{rs} = 1 \quad \forall r \in R \\
& z_s, y_{rs} \in \{0, 1\} \quad \forall r \in R, s \in S
\end{aligned}
$$

- Used tangent lines at certain points to approximate
- Simplified objective function significantly before attempting to linearize
- Idea: more lines towards the beginning of the function, where there is more curve, less towards the end where the function is steeper
- Not perfect yet, may need less or more constraints in the future, or different ones.
- Computation time much faster, is linear so we can use CPLEX
- Will help us run our program with a lot more data in the future at less of a computational cost

CPLEX

- CPLEX uses branch and cut procedure
- Below is a table demonstrating how different branching strategies in CPLEX affect the runtime of our program
- Branching direction for integer variables is selected by ‘ branch’.
  - branch=1 is branching up, branch=-1 is branching down
- Strategy for selecting the next branching variable during integer branch-and-bound is selected by ‘varsel’
  - varsel=-1 is branch on variable with smallest integer infeasibility, varsel=1 is branch on variable with largest integer infeasibility, and varsel=3 is strong branching

branch	varsel	Root relax.	Root node procc. time	b&c time	Total time	# simplex iteration	# b&b nodes
1	-1	2.55	2.79	0.29	3.08	136	5
1	1	2.63	2.87	0.25	3.12	110	3
1	3	2.6	3.08	0.46	3.54	77	3
-1	-1	2.58	2.82	0.26	3.08	110	3
-1	1	2.71	2.94	0.35	3.19	110	3
-1	3	3.15	3.62	0.45	4.07	79	3

```
Root relaxation solution time = 2.53 sec. (2200.88 ticks)

        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

      0     0      618.6342   962                    618.6342       54
Detecting symmetries...
      0     2      618.6342   962                    618.6342       54
Elapsed time = 3.79 sec. (3392.40 ticks, tree = 0.02 MB)
*     1     1      integral     0      618.6475      618.6345       64    0.00%
Found incumbent of value 618.647524 after 4.91 sec. (3425.87 ticks)
      1     1      618.6475     0      618.6475      618.6345       64    0.00%
      2     3      618.6826   962      618.6475      618.6345       67    0.00%

Root node processing (before b&c):
  Real time             =    3.62 sec. (3256.34 ticks)
Parallel b&c, 4 threads:
  Real time             =    1.32 sec. (3181.65 ticks)
  Sync time (average)   =    0.12 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    4.94 sec. (6437.98 ticks)
Using steepest-edge.
CPLEX 12.10.0.0: optimal integer solution; objective 618.6475236
77 MIP simplex iterations
3 branch-and-bound nodes
```

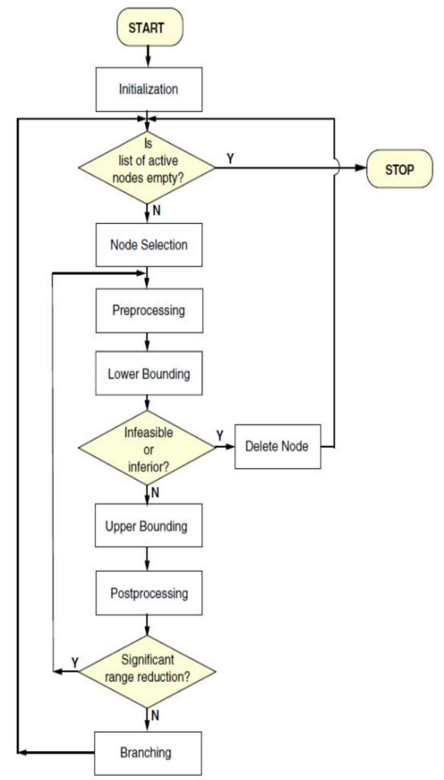
example output from CPLEX with the default branch and varsel options

BARON: Branch-And-Reduce Optimization Navigator

- Proprietary software, so unknown EXACTLY how the program works
- Branch and reduce is a common strategy for non-linear programs.
- Reduce is used to narrow the interval of variables.
- Baron uses the a dynamic strategy for range reduction that is based on:
  - Nonlinear-feasibility-based range reduction option.
  - Marginals-based reduction option
  - Linear-feasibility-based range reduction option.
  - Optimality-based tightening option.
- The branching strategies baron uses are a dynamic combination of the following:
  - **Variable selection strategy:** largest violation and longest edge
  - **Point selection strategy:** ω-branching, bisection-branching, convex combination of ω and bisection
  - **Node selection rule:** best bound, LIFO, minimum infeasibilities

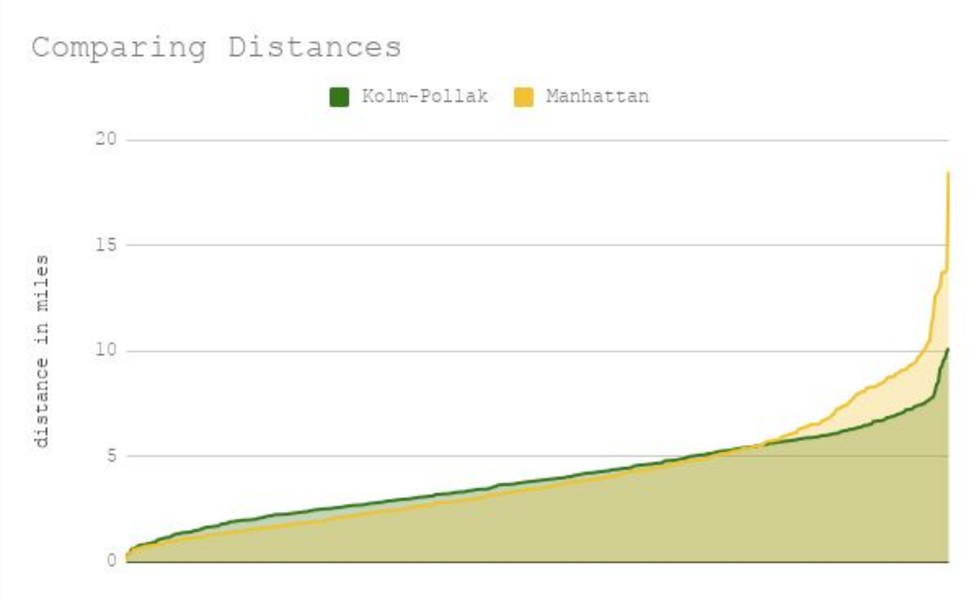
```
Doing local search
Solving bounding LP
Starting multi-start local search
Estimated remaining time for local search is 50 secs
Done with local search
===========================================================================
  Iteration      Open nodes          Time (s)    Lower bound      Upper bound
        1+             1              330.20    -0.191680E-11    0.100000E+52
*       1              1              334.97      0.00000         5163.14
        1+             1              382.45      0.00000         5163.14
*       1              1              383.92      0.00000         100.961
        1+             1              428.07      0.00000         100.961
        1+             1              653.53     100.264          100.961
        1+             1              708.72     100.264          100.961
        1              1              744.37     100.264          100.961
        1+             1              957.78     100.436          100.961
        1+             1             1022.54     100.436          100.961
        1+             1             1060.88     100.436          100.961
        1+             1             1268.83     100.459          100.961
        1              1             1596.42     100.659          100.961
        2+             2             1634.49     100.659          100.961
        3+             2             1666.61     100.659          100.961
        4              3             1696.97     100.826          100.961
*       4              2             1705.64     100.826          100.838
        5              0             1717.32     100.838          100.838

Cleaning up

                        *** Normal completion ***

Wall clock time:                     1753.18
Total CPU time used:                 1727.51

Total no. of BaR iterations:           5
Best solution found at node:           4
Max. no. of nodes in memory:           3
```

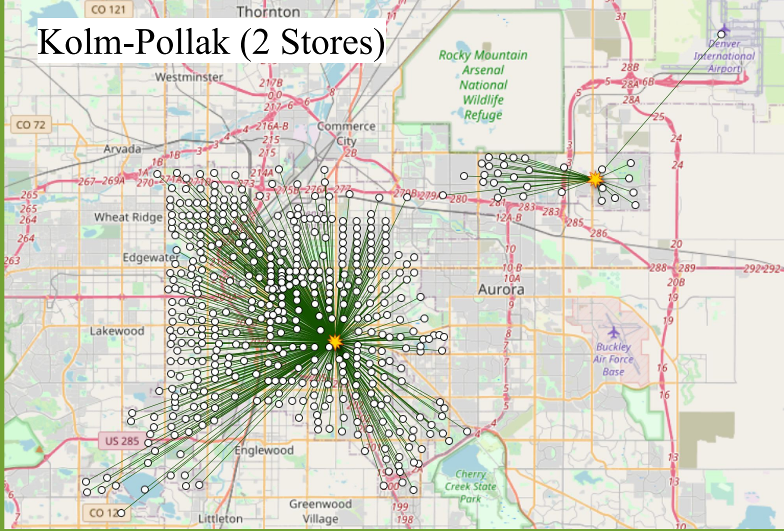
Sample AMPL BARON output

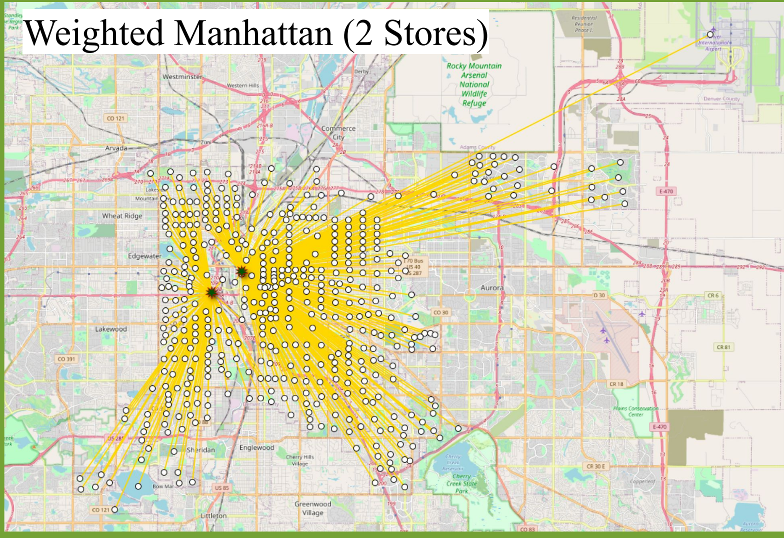


Demonstrating the Difference between Programs

- We ran each program and asked them to pick only two stores for all of Denver
- The chart below is of the “as the crow flies” distances(in miles) of the centroids to the stores each program picked

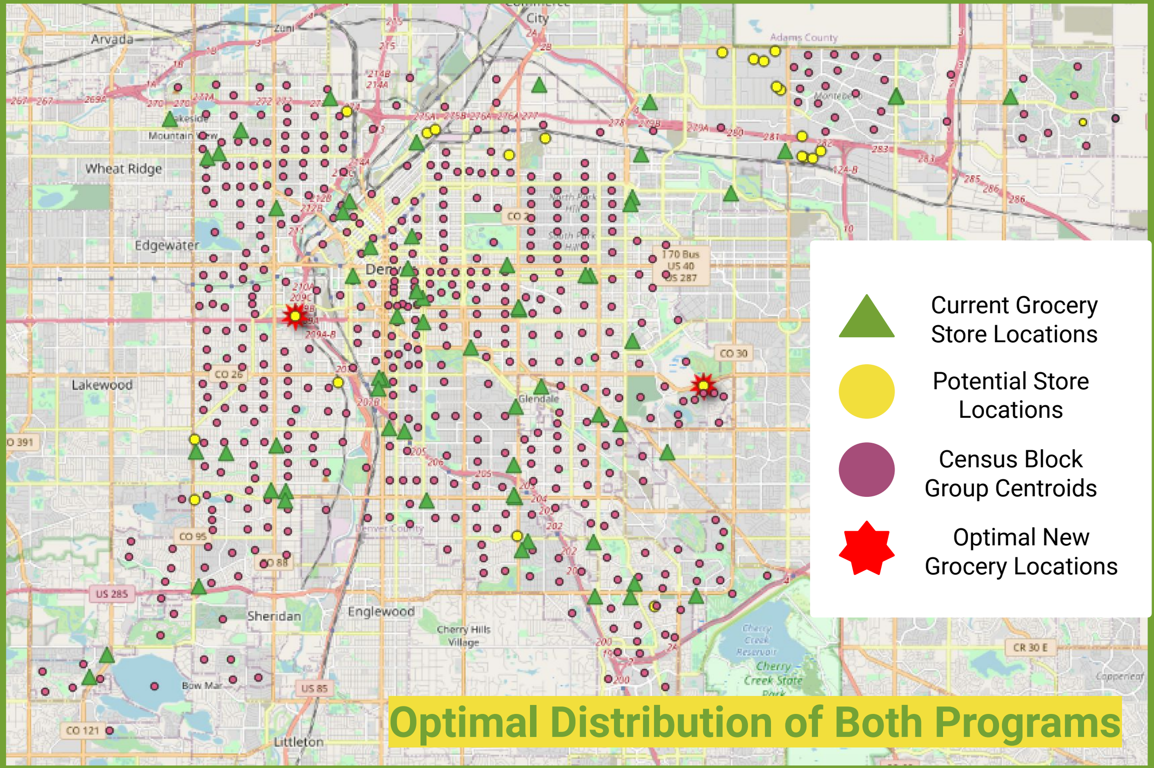






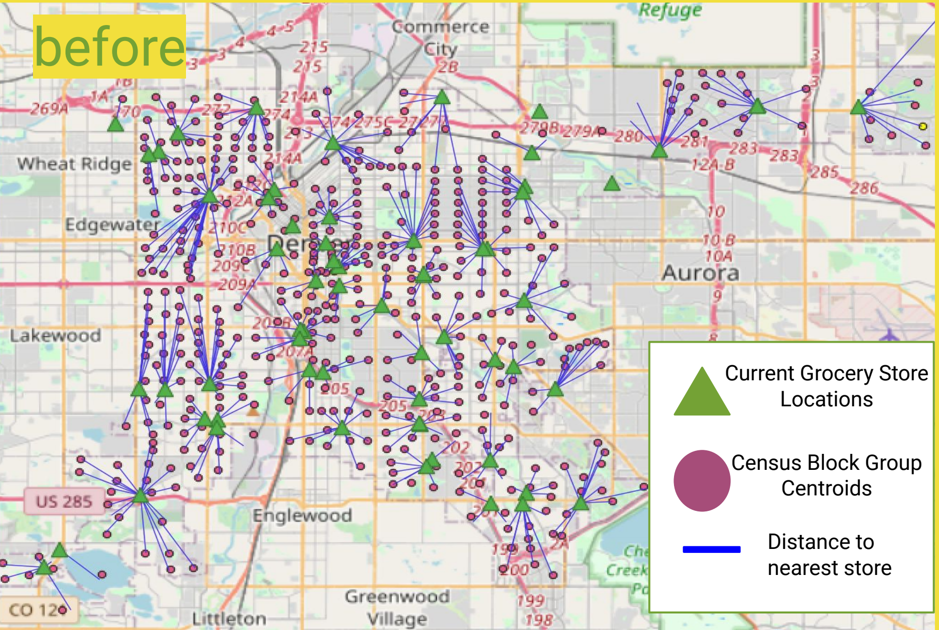
Adding Two New Stores to the Existing Distribution

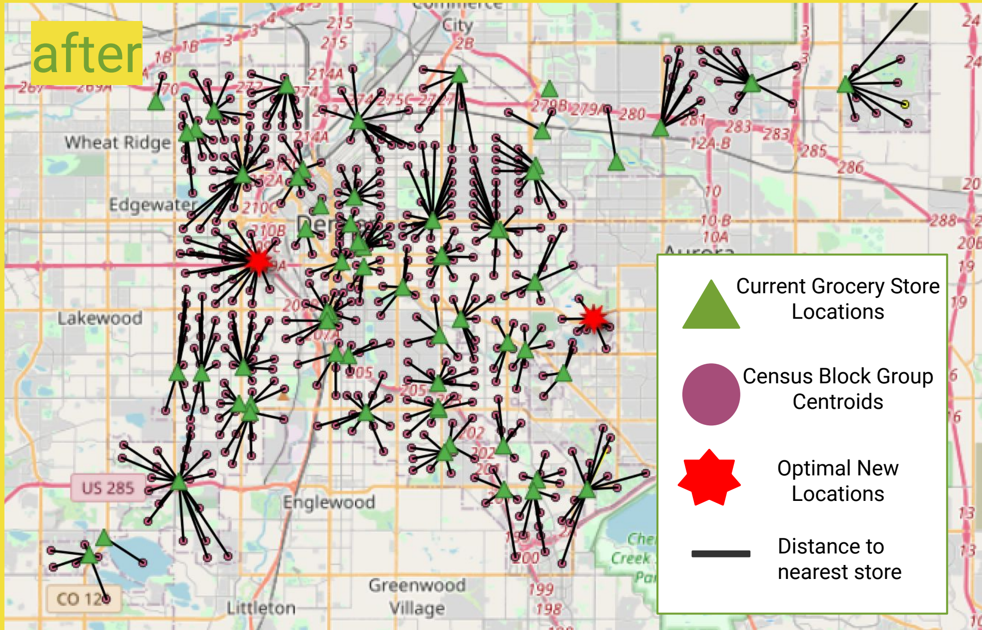
- We asked both programs to produce an optimal distribution with two new stores given the 64 current grocery store locations.
- Both programs gave the same optimal distribution of grocery locations for two stores
- We also ran the program with a larger spread of more potential locations to see if the programs would produce different distributions, but they produced identical distributions.



Interpretation

- With the added stores, only 96/480 of the census block group centroids would be more than 1 mile from a grocery store.
- This is 19 less than the current distribution of grocery stores.





Community Fridges

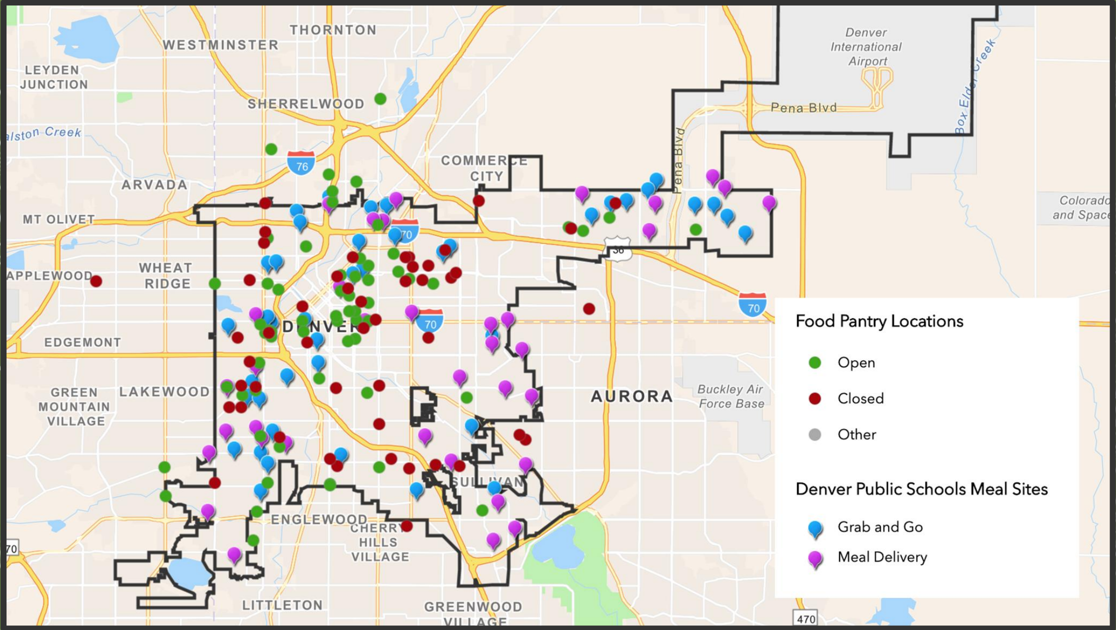
-**Community Fridges** are an expansion of what food pantries can provide, these fridges are stocked with fresh fruits and vegetables and packaged meals. (Any packaged meal donations include a list of all ingredients.)

-**Denver Community Fridges** is a mutual aid program that is working to fight food insecurity in the Denver Metro area, which is located on the stolen land of the Cheyenne, Arapahoe, and Ute. (https://www.denvercommunityfridge.com)

-Our programs can be used to find the optimal locations for additional Denver Community Fridges

The Denver Community Fridges refrigerator and stocked shelves outside Basecoat Nail Salon in Five Points. Photo by Eli Zain

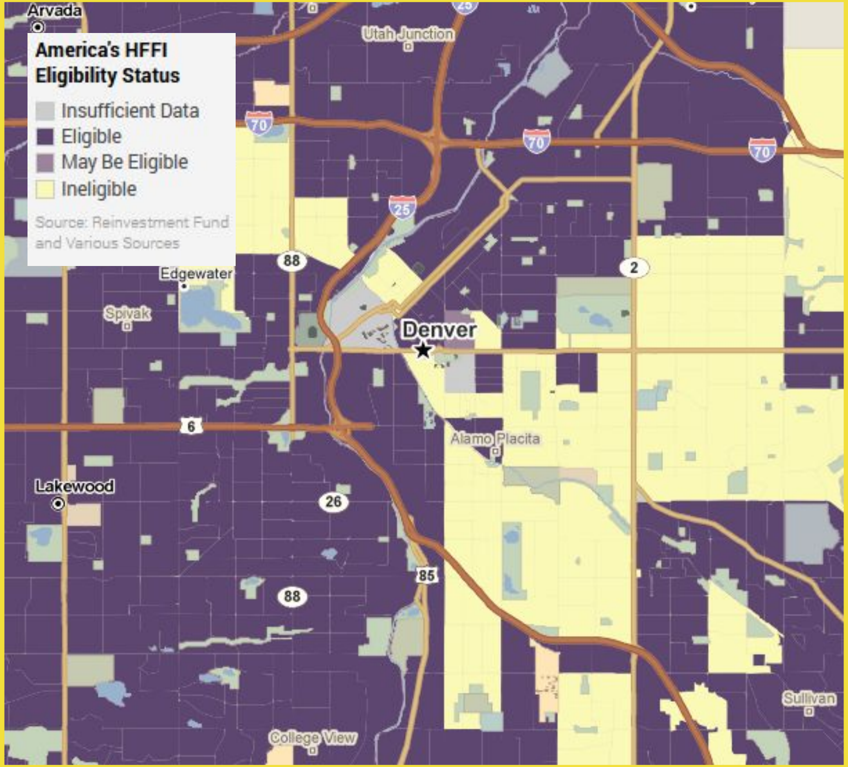
Food Pantries in Denver



- In the beginning months of the COVID-19 pandemic, only 62/106 food pantries in Denver were operational
- Our programs could have been used to determine which of the 44 non-operational food pantries would be optimal to open first
- Now, with all of the food pantries open, our programs could be used to find optimal new locations for food pantries

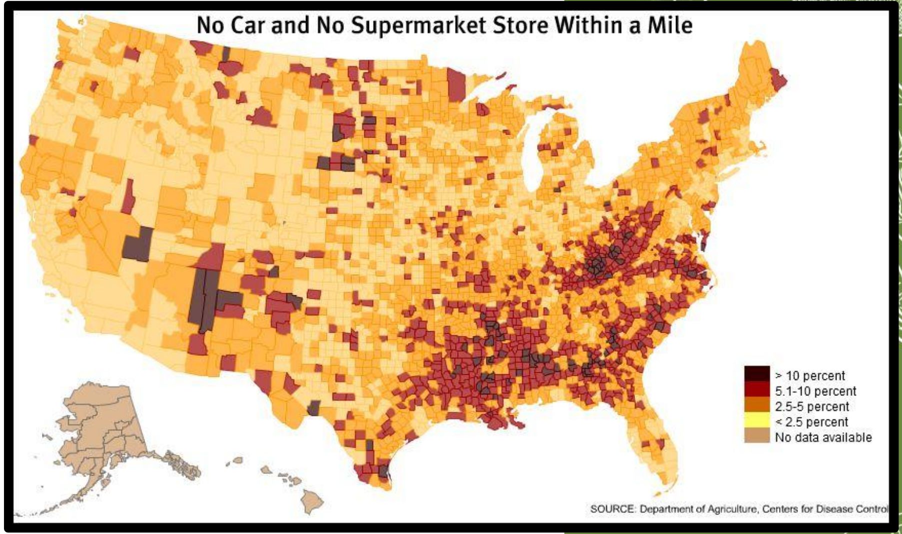
Healthy Food Financing Initiative

- The Healthy Food Financing Initiative (HFFI) is a partnership that provides grants and loans to finance the construction and development of grocery stores and other healthy food retailers in underserved areas
- Between 2011 and 2015, HFFI helped support/develop 1,000 grocery stores nationwide
- Much of the Denver area is eligible



Future Work

- Expand to other cities where food access is worse and disproportionately affects marginalized communities
- Use the programs to look at other amenities such as parks, schools, voting locations, hospitals, bus stops, etc.
- Expand to look at multiple amenities at once
    - For example, bus stops and grocery stores would be a good combination when looking at food insecurity
- Look at Denver in a larger scale (i.e. the whole Denver metro area instead of just Denver County)



Works Cited

Kilinc, M. and N. V. Sahinidis, Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems in BARON, Optimization Methods and Software, 33, 540-562, 2018.

T.M. Logan, M.J. Anderson, T.G. Williams, L. Conrow, Measuring inequalities in urban systems: An approach for evaluating the distribution of amenities and burdens, Computers, Environment and Urban Systems, Volume 86, 2021, 101590, ISSN 0198-9715, https://doi.org/10.1016/j.compenvurbsys.2020.101590.

OpenRefine [Computer software]. (2021). Retrieved from https://openrefine.org/

City of Denver Open Data Catalog. (2021). Food Retail Locations (2020) [Data file]. Retrieved from https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-food-retail-locations

City of Denver Open Data Catalog. (2021). Census Blocks (2010) [Data file]. Retrieved from https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-census-blocks-2010

QGIS Development Team (2021). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org".

Thank you!