A neighborhood search heuristic for the p-median problem with continuous demand*

Shane Auerbach[†] Ⓡ Rebekah Dix[‡]

[†]Lyft, Inc.
[‡]University of Wisconsin-Madison

Abstract

Computing optimal spatial allocations is important for two reasons. First, one may wish to implement them. Second, without a sense of an optimal spatial allocation, one cannot evaluate the efficiency of an observed spatial allocation. Suppose you have p facilities and wish to place them in a city to minimize the average distance between a consumer and her nearest facility. We develop a neighborhood search heuristic for this p-median problem with continuous demand. We discuss challenges to implementing the heuristic, propose solutions, and describe how it can be embedded in hybrid heuristics. We then apply the heuristic to computing optimal spatial allocations of facilities in Chicago, Atlanta, and Los Angeles. In comparing these optimal allocations to the actual ones, we find that allocations of supermarkets do relatively poorly in minimizing transportation costs for consumers.

1 Introduction

There are a variety of important problems relating to facility location.[1] In covering problems, one seeks the minimum number of facilities, e.g. fire stations, and their locations, such that each household is within a given distance of its nearest facility. In center problems, one takes the number of facilities (p) as given and places them so as to minimize the maximum distance from a household to its nearest facility. In median problems, one places them to minimize the sum of consumers' distances rather than the maximum.[2] In this paper,

*We wish to thank Antonio Penta, Daniel Quint, Ken Hendricks, Justin Sydnor, Lones Smith, Marzena Rostek, and Marek Weretka for their guidance and feedback. We have benefited from feedback from seminar participants at the University of Wisconsin, Universidad de los Andes (Santiago), Université de Sherbrooke, Macalester College, and Carleton College as well as at the 2018 INFORMS Annual Meeting. Ⓡ denotes random author order. Data and code from the application are available upon request.

[1]See Love et al. (1988) and Drezner and Hamacher (2001) for a survey and history of the literature, and Daskin (2011) for a textbook treatment on facility location on discrete networks.

[2]Median problems are often called Weber or Fermat-Weber problems as they are generalizations of problems first discussed in Weber (1929) and of the Fermat point, which is the special case of the point that minimizes the total distance to the three vertices of a triangle.

we focus on a variant of the p-median problem in Euclidean space, as opposed to a discrete network, with continuous consumer demand. In this environment, we develop a neighborhood search heuristic that finds local optima and can easily be embedded into hybrid heuristics or metaheuristics to find approximate global optima.

The p-median problem, first proposed in Hakimi (1964, 1965), can be formulated as follows:

Problem 1: The p-median problem

Inputs and decision variable:

$$\begin{aligned}
\mathcal{N} &= \text{a set of nodes in a network} \\
w_m &= \text{demand at node } m \in \mathcal{N} \\
d(m,n) &= \text{distance between demand node } m \in \mathcal{N} \text{ and candidate site } n \in \mathcal{N} \\
p &= \text{number of facilities to locate}
\end{aligned}$$

$$s = (s_1, s_2, \ldots, s_p) = \text{choice of } p \text{ nodes at which facilities are to be located}$$

Problem:

$$\underset{s}{\text{Minimize}}\ z_1 = \sum_{m \in \mathcal{N}} w_m \cdot \min_k d(m, s_k)$$

The goal in the p-median problem is to allocate p facilities across a network to minimize the total (or average) travel distances for consumers. Kariv and Hakimi (1979) shows that finding a p-median is NP-Hard. For even moderately sized problems, the number of possible solutions to evaluate is huge—for example, with 100 nodes and $p = 20$, there are $\binom{100}{20} > 10^{20}$ possible allocations. Because of this, one typically solves p-median problems with heuristics including constructive, local search, mathematical programming and others. Metaheuristics are higher-level heuristics designed to generate specific heuristics for specific problems—these include tabu search, variable neighborhood search, genetic search, simulated annealing, neural networks, and many more. Mladenović et al. (2007) surveys applications of heuristics and metaheuristics to the p-median problem.

In the p-median problem, the space is discrete in that consumers and facilities are located on a network. A similar question can be posed on the Euclidean plane:

Problem 2: The p-median problem with continuous feasible placements

Inputs and decision variable:

$$\begin{aligned}
\mathcal{N} &= \text{a discrete set of consumer locations } (\mathcal{N} \subset \mathbb{R}^2) \\
w_m &= \text{demand at location } m \in \mathcal{N} \\
d(m,n) &= \text{distance between } m \in \mathcal{N} \text{ and candidate site } n \in \mathbb{R}^2 \\
p &= \text{number of facilities to locate}
\end{aligned}$$

$$s = (s_1, s_2, \ldots, s_p) = \text{choice of } p \text{ locations for facilities (each } s_k \in \mathbb{R}^2)$$

Problem:

$$\underset{s}{\text{Minimize}}\ z_2 = \sum_{m \in \mathcal{N}} w_m \cdot \min_k d(m, s_k)$$

In this problem, which is also called a *location-allocation problem* or the *multisource Weber problem*, the set of options is infinite as facilities may be positioned anywhere on the Euclidean plane. Cooper (1963) shows that the objective function is neither concave nor convex and proposes heuristics. Megiddo and Supowit (1984) proves that finding the solution is NP-hard. Brimberg et al. (2000) surveys and compares the many heuristics and metaheuristics for this problem.

Our problem is like *Problem 2* except that we also allow consumer demand to be continuous. That is, instead of having discrete consumers located at specific locations on the Euclidean plane, we have an integrable consumer density function over the plane. The problem can be formulated as follows:

Problem 3: The p-median problem with continuous demand

Inputs and decision variable:

$$\begin{aligned} X \subseteq \mathbb{R}^2 &\text{ is a subset of Euclidean space} \\ f : X \to \mathbb{R}_{\geq 0} &\text{ is an integrable consumer density function} \\ d(m,n) &= \text{distance between } m, n \in X \\ p &= \text{number of facilities to locate} \end{aligned}$$

$$s = (s_1, s_2, \ldots, s_p) = \text{choice of } p \text{ locations for facilities (each } s_k \in X)$$

Problem:

$$\underset{s}{\text{Minimize}}\ z_3 = \int_X f(x) \cdot \min_k d(x, s_k)\ \mathrm{d}x$$

Here we have a continuum of consumers across a subset of the Euclidean plane. Note that f must be integrable but it need not be continuous—we call the demand continuous to distinguish it from the problems above with discrete consumers. Newell (1973) and Geoffrion (1979) are the first to suggest modeling demand with a continuous function. Iri et al. (1984) presents the only heuristic, a gradient-based search, for *Problem 3* with Euclidean distances.[3] Fekete et al. (2005) argues the problem's importance and is the first to develop algorithms for exact solutions with Manhattan (ℓ_1) distance—the authors acknowledge that comparable algorithms for Euclidean distance (ℓ_2) are likely to require intractable integration.

We believe that *Problem 3* is most appropriate in two particular settings. First, in Auerbach and Dix (2018), we discuss the problem of a ride-sharing service that wishes to place its drivers so as to minimize wait-times for passengers. In this setting, the service may have expectations about where, and with what probabilities, the next ride request is likely

[3] Murat et al. (2010) offers a similar heuristic that allows for richer cost structures including facility capacities and transportations costs that vary with demand volumes. See also Okabe et al. (1992) and Okabe and Suzuki (1997) for similar approaches using Voronoi diagrams on other problems.

to emerge. f could be interpreted as a probability density function over the location of the next ride request. Solving *Problem 3* would minimize the expected distance between the to-be-revealed ride request and its nearest driver, thereby minimizing the expected wait-time. In other settings, also, firms or planners may have expectations of consumer density better represented by an integrable function than by discrete demand nodes.

Second, when an analyst is working with spatially aggregated data, she can often place a particular consumer only within a certain region rather than at a specific point. For instance, with census data, we observe the number of consumers in a census tract as well the precise coordinates of the tract's vertices, but we do not observe the specific locations of residents therein. We work with an application involving census data in Section 3.

Problems 1 and *2* receive more attention than *Problem 3*. We suspect this is because the inclusion of integration in place of summation in the objective function is problematic for the purposes of computation. Numerical integration can be computationally costly, so including it in an iterative process can limit its feasibility for larger problems. It is also fairly easily avoided by transforming *Problem 3* into *Problem 2* by discretizing and aggregating the demand into nodes or further into *Problem 1* by also discretizing the space into a network. For instance, one could discretize census data by assuming that all residents within a census tract live at the centroid of that tract. Given that one is looking only for approximate solutions to these problems in any case, the cost of such discretization is simply that the solution is perhaps slightly more approximate.

On the other hand, discretization and aggregation may introduce error that it is hard to quantify (Francis et al., 2009). Two very different polygons may share a centroid—aggregating a census tract's demand to its centroid discards much of the richness of the spatial data. With census tracts, we believe it is more appropriate to assume that consumers are distributed uniformly within each.[4] Where a planner has an integrable function representing expected demand, we believe it best to use it directly. Therefore, we propose a heuristic to solve *Problem 3*. Our heuristic is a neighborhood search heuristic in that it converges to local optima of *Problem 3* through iterative adjustments of each facility's location within its neighborhood.

Maranzana (1963) develops the original neighborhood search heuristic for the p-median problem. In the p-median problem, a particular facility's neighborhood is the set of demand nodes that are served by the facility, i.e. those that are closer to that particular facility than to any other. In Figure 1, assume there is one consumer at each node and the numbers on the edges represent distances between nodes. Given two facilities located at the nodes represented with triangles, the thick, diagonal, dashed line separates the two neighborhoods. Local search seeks to reposition each facility to best serve its neighborhood. In the left neighborhood, the average distance to facility is 2 and this cannot be improved by repositioning the facility to another node. In the right neighborhood, the average distance is 3.75. By moving the facility inward, the neighborhood average would reduce to 2.25. The heuristic alternates between implementing local adjustments and recomputing the neighborhoods—it is sometimes referred to as an alternate heuristic. In this example, the

[4]While we could approach that assumption by generating multiple demand nodes within each tract and distributing the consumers across the nodes, we would then have a discrete model with many nodes, which yields its own computational challenges.

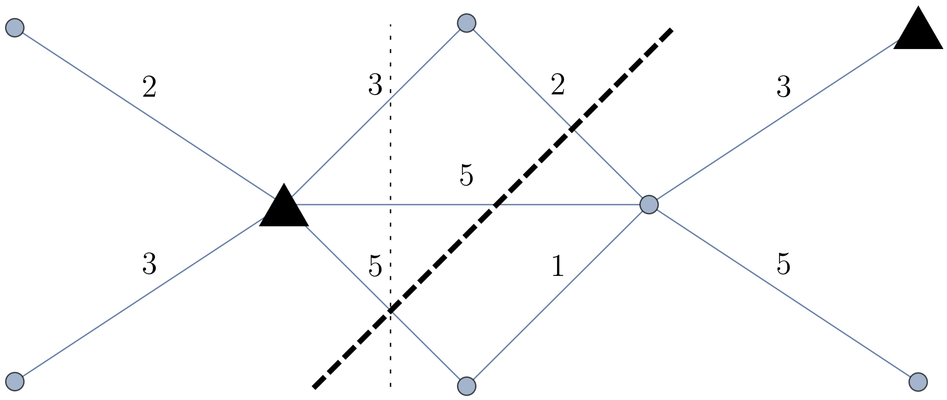


Figure 1: Neighborhood search on a discrete network

first adjustment would result in the top-central node joining the right neighborhood, moving the neighborhood boundary to the thin dotted line. Then it can be shown that no further local adjustment is beneficial, so the heuristic terminates.[5]

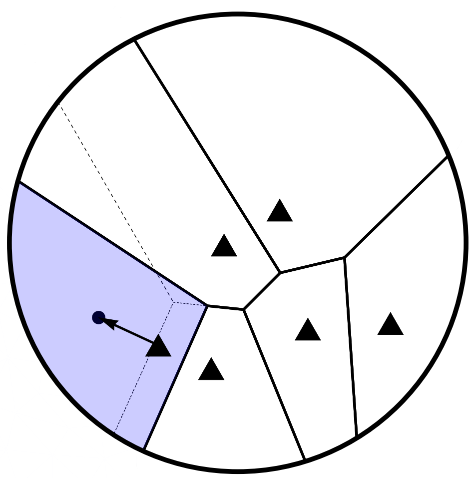


Figure 2: Neighborhood search with continuous demand

With continuous demand, a facility's neighborhood is still the set of consumers that is closer to the facility than to any other, but this now manifests as the facility's cell in a Voronoi diagram of the space. Figure 2 is a Voronoi diagram of six facilities, each represented by a triangle, on a disk. The solid lines divide the Voronoi cells, where each cell is the set of points in the disk closest in Euclidean distance to a particular facility. Neighborhood search in this context involves taking a particular facility and positioning it within its Voronoi cell to minimize the objective function applied locally to that cell. For the left-most facility, whose cell is shaded, this requires moving it up and to the left to the point. The adjustment also results in a new Voronoi diagram, as indicated by the thin dashed lines. We prove that such a movement, which by definition reduces the objective function applied locally, also

[5]In this example, the heuristic terminates at the global optimum for the 2-median problem on the given network, though this need not occur generally.

reduces the objective function applied globally. Iterating this procedure, therefore, must converge to a local minimum in the objective function.[6]

Readers familiar with computer science or electrical engineering may recognize similarities between this procedure and Lloyd's algorithm, which finds evenly spaced sets of points in subsets of Euclidean spaces (Lloyd, 1982). While the logic of how the algorithm works is identical, there are two key differences: First, in Lloyd's algorithm the relevant objective function is the integral of the *squared* distances. Second, Lloyd's algorithm weights all points equally, i.e. f is uniform. In this context, the optimal local adjustment simply moves a facility to the centroid of its Voronoi cell.

To find the optimal local adjustment in our heuristic, we actually need to solve the 1-median problem with continuous demand within the Voronoi cell. While it might seem problematic that our heuristic to solve the p-median problem with continuous demand requires us to repeatedly solve a smaller 1-median problem with continuous demand, it works well in practice because an approximate solution to the smaller problem can be computed efficiently. If f is uniform, this reduces to finding the geometric median of the Voronoi cell. Since finding the geometric median of a polygon is challenging, we discretize the problem at this stage by replacing the polygon with a set of points randomly drawn from it. This also allows us to account for non-uniform f in that we draw points from the polygon in proportion to their density in f. Then, we find the approximate geometric median of this set of points.[7]

Our neighborhood search heuristic for *Problem 3* has two obvious benefits over the gradient-based search of Iri et al. (1984). First, it is not necessary to evaluate the objective function, z_3, at any point in the heuristic.[8] This minimizes the computational burden. Second, it is significantly easier to implement.

The paper proceeds as follows: In Section 2, we define the heuristic, prove its convergence, and discuss extensions. In Section 3, we implement the heuristic to optimize allocations of supermarkets, hospitals, and fire stations in three US cities. We also use these computed optimal allocations to measure the implied inefficiencies of the actual allocations of those facilities in those cities. We conclude in Section 4.

2 Neighborhood search heuristic

In this Section 2.1, we define precisely our neighborhood search heuristic and prove that it converges to a local optimum of *Problem 3*. In Section 2.2 we discuss modifications to the heuristic to speed implementation. In Section 2.3 we propose embedding the heuristic in a

[6] We describe how one might hope to arrive at a global optimum in Section 2.

[7] Algorithms for finding the geometric median of a discrete set of points go back to Weiszfeld (1937). See Bose et al. (2003) and Cohen et al. (2016) for modern alternatives. While we do discretize the polygon into a set of points, we do this with thousands of points on an individual Voronoi cell—if we did this with nodes on the overall space, we would run into computation issues due to the number of nodes. The minimal error that may be introduced from the discretization in a given iteration is unlikely to affect final outcomes given subsequent iterations.

[8] In practice, it may make sense to set a stopping condition when $\Delta z_3 < \gamma$, where Δz_3 is the change in the objective function over some duration or iteration count and γ is a tolerance, but even these stopping conditions can be checked infrequently.

hybrid or metaheuristic.

2.1 Simple neighborhood search

Let $X \subset \mathbb{R}^2$ denote a bounded region on the Euclidean plane. Let $\mathcal{I} = \{1, \ldots, I\}$ denote the set of facilities. A *spatial allocation* s is an I-tuple $(s_1, s_2, \ldots, s_I) \in \prod_{i \in \mathcal{I}} X$ that records the locations of facilities in X. Thus, s is a list of I locations, where the i^{th} element, s_i, is the location of facility i. Let $\mathcal{S}_I = \prod_{i \in \mathcal{I}} X$ be the set of all possible spatial allocations and define $s_{-i} \in \prod_{j \in \mathcal{I} \setminus \{i\}} X$ as the spatial allocation of all facilities other than i's. This notation allows us to consider the movement of facility i holding the positions of her opponents fixed. We will refer to the 2-tuple (x, s_{-i}) as the spatial allocation in which facility i is positioned at location $x \in X$ and the other facilities are positioned according to s_{-i}. We will frequently compute the Voronoi diagram of s, $\mathrm{Vor}(s)$, to set the neighborhoods. $\mathrm{Vor}(s)$ partitions X into Voronoi cells $V_i(s) = \{x \in X \mid d(x, s_i) \leq d(x, s_j), \forall j \neq i\}$ for each facility $i \in \mathcal{I}$.

In its simplest form, our heuristic solution to *Problem 3* can be stated as follows:

Procedure 1 Simple neighborhood search heuristic (SNS)

1: **function** $\mathrm{SNS}(X, f, s, \text{tolerance})$ ▷ s is an arbitrary initial allocation
2: complete $= 0$
3: **while** complete $= 0$ **do**
4: **for** $i = 1$ to I **do**
5: Compute $\mathrm{Vor}(s)$
6: $s_i' = \arg\min_{y \in V_i(s)} \int_{V_i(s)} f(x) \cdot d(x, y)\ \mathrm{d}x$ ▷ Find optimal adjustment
7: $\Delta_i = d(s_i', s_i)$ ▷ Measure adjustment size
8: $s_i = s_i'$ ▷ Implement adjustment
9: **end for**
10: **if** $\max_i \Delta_i <$ tolerance **then**
11: complete $= 1$, ▷ Terminate when adjustment sizes are below tolerance
12: **end if**
13: **end while**
14: **end function**

The heuristic iteratively computes the Voronoi cells and then implements optimal local adjustments. We now show that it converges to an approximate local minimum of *Problem 3*:

Theorem 1. *Procedure 1 converges to an approximate local minimum of Problem 3.*

Proof. We will first argue that each local adjustment computed in Line 6 and implemented in Line 8 weakly reduces the objective function of *Problem 3*. Let $z_i(s) = \int_{V_i(s)} f(x) \cdot d(x, s_i)\ \mathrm{d}x$. Now, note that the the objective function for *Problem 3* can be rewritten as follows:

$$z(s) = \int_X f(x) \cdot \min_k d(x, s_k)\ \mathrm{d}x = \sum_{i \in \mathcal{I}} \left(\int_{V_i(s)} f(x) \cdot d(x, s_i)\ \mathrm{d}x \right) = \sum_{i \in \mathcal{I}} z_i(s)$$

This exploits the fact that $V_i(s)$ represents the set of points closest to s_i and breaks the integral over X into a sum of integrals over the Voronoi cells. Moving facility i from s_i to s_i' must weakly reduce $z_i(s)$ by the definition of s_i' in Line 6. For consumers in $V_j(s)$ for $j \neq i$, there are two possibilities: First, if the consumer's location is in $V_i(s_i', s_{-i})$, then the consumer will move into i's Voronoi region as a result of the adjustment of facility i. If this is so, that consumer's distance must be reduced by the adjustment from s_i to s_i'—otherwise, she would remain in $V_j(s_i', s_{-i})$, with $j \neq i$. Second, it may be that a consumer is located in $V_j(s)$ and $V_j(s_i', s_{-i})$ for $j \neq i$. In this case, the consumer's distance is unchanged. Therefore, it follows that each $z_j(s)$, for $j \neq i$, is weakly reduced by the movement of i from s_i to s_i'.

If each individual adjustment reduces the objective function, then the procedure must eventually terminate for any non-zero tolerance. The resulting allocation is an *approximate* local optimum of *Problem 3* in that, for any given tolerance, the procedure may be terminated before it reaches the precise fixed point. ■

Figure 3 illustrates the proof. The figure shows a spatial allocation of seven facilities. We consider an adjustment for the central facility. The neighborhood of the central facility, its Voronoi cell, is denoted by the solid hexagon. The optimal local adjustment moves it to the small circle, which is the site that minimizes the objective function applied on the Voronoi cell—with uniform demand, this is the geometric median of the solid hexagon. The dashed hexagon represents the neighborhood of the central facility following the adjustment.

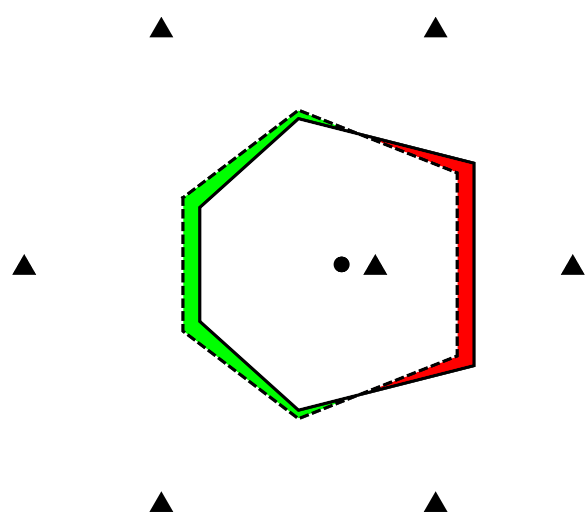


Figure 3: A neighborhood adjustment reduces the objective function

To argue that the adjustment reduces the overall objective function, observe how it affects consumers outside of the solid polygon. It results in consumers in the green shaded region switching to the central facility. If they are switching, it is because they are now closer to the central facility than they were to their closest facility prior to the adjustment, so those consumers are made better off. No other consumers outside of the solid hexagon are affected, hence it follows that the adjustment must reduce the overall objective function.

While it is not necessary for the proof, consider also the red area in Figure 3. These consumers are further from the central facility following the adjustment then they were prior. Their lost welfare was taken into consideration when picking the optimal local adjustment. In fact, however, consumers in the red region are not as worse off as it would appear. Following the adjustment, they are now closer to facilities other than the central facility—fixing their assignment to the central facility overstated their welfare loss. This strengthens the degree to which the local adjustment must reduce the objective function.

In a local neighborhood search heuristic on a discrete network, moving a facility may not affect the neighborhoods. Therefore, one can reach a fixed point quickly but this fixed point may be substantially suboptimal in that it may fail to implement changes that improve the objective function in areas outside of the neighborhood. One of the nice properties of a neighborhood search heuristic with continuous demand is that any adjustment in the location of a facility results in a redefinition of the neighborhoods. The redefinition of neighborhoods, in turn, will always result in a new optimal location in at least one neighborhood, provided $f(x) > 0$ for all $x \in X$. With continuous demand, we never expect to reach a fixed point. Instead, the heuristic asymptotes to a local optimum and we terminate it when we think we are close enough, i.e. within a tolerance.

2.2 Modifications to facilitate implementation

There are two obvious challenges with the implementation of the heuristic. The first is that computing the optimal adjustment, s'_i, is non-trivial. The second is that the included loop requires sequential, not simultaneous, computations and is not conducive to parallelization.

Our solution to the first challenge is a procedure that takes as inputs a Voronoi cell (V_i), a density function (f), and a count for the number of points to be used in the discretization of the Voronoi cell (N). It outputs the optimal adjustment for the discretized Voronoi cell:

Procedure 2 Compute optimal adjustment (s'_i)

function COMPUTEOPTIMALADJUSTMENT(V_i, f, N)
  g = rescaled f, such that $\int_{V_i} g(x)\mathrm{d}x = 1$
  sample = N points drawn randomly from V_i with weighting from g
  s'_i = geometric median of sample
end function

Procedure 2 approximates the Voronoi cell and f by a discrete set of points and then finds the geometric median of that set.[9]

Our solution to the second issue, regarding parallelization, is simply to swap lines 4 and 5 in Procedure 1, bringing the computation of the Voronoi diagram outside of the *for* loop. Technically, with this configuration, it is no longer necessarily true that each adjustment will weakly reduce the objective function. If several adjustments have already been made, an optimal adjustment on an out-of-date Voronoi cell could increase the actual objective function. In practice, however, this is unlikely to occur and inconsequential in any case. If

[9] Again, see Weiszfeld (1937), or Bose et al. (2003) and Cohen et al. (2016) for more modern solutions. This functionality is also efficiently built-in with some software, including Mathematica.

such an adjustment is made, it will be undone in subsequent iterations in which the cells have been updated. Also, as the algorithm converges, the sizes of the adjustments are reduced and therefore so is the degree to which the neighborhoods change as a result of the adjustments. Early bad adjustments based on out-of-date Voronoi cells will be undone by subsequent adjustment, and bad adjustments are less and less likely to occur as the algorithm converges and the neighborhoods become more static.[10]

The benefits of this configuration are obvious: with the Voronoi cells held constant, one can compute the optimal adjustments for all I Voronoi cells simultaneously in a parallelized setting. The computation can be distributed across as many as I cores.

2.3 Embedding neighborhood search in a hybrid heuristic

The neighborhood search heuristic described in Procedure 1 can be run from an arbitrary starting spatial allocation. On the other hand, it will converge much more quickly, and possibly to a better local optimum, if it is given a good allocation to start from. To this end, one could compute a large set of random spatial allocations, evaluate the objective function on each, and run the heuristic on the best among them.[11]

Another alternative would be to construct a good starting allocation using a myopic (greedy) heuristic à la Kuehn and Hamburger (1963): Draw a large set of random points from X. Move a single facility across all of the points, evaluating the objective function in each case, and finally placing the facility at the point that minimizes the objective function. Then, taking the location of the first facility as given, try placing a second facility at each of the points, evaluating the objective function with two facilities in each case and placing the second facility to minimize it. Repeat until all facilities are placed.

Beyond reducing computation times by providing a good starting allocation, one may also hope to find the global optimum, or at least a *good* local optimum. There are several approaches for this. First, one could run the local search heuristic from many starting allocations and select the best local optimum. Second, one could implement a shaking procedure. A shaking procedure takes a local optimum and disrupts it. In our context, this could involve taking one or more facilities and moving them to new locations in X. Then the local search heuristic could be resumed to find a new local optimum.

It could be particularly useful to look for global adjustments that may not be realized with neighborhood search. For instance, one could look for local maxima of $h(s,x) = f(x) \cdot \min_k d(s_k, x)$ as areas with significant density and distance from facility in allocation s—these are good candidate sites for a facility. To find candidate sites for facility removal, one could look at sites with low market share, i.e. $\int_{V_i(s)} f(x)\mathrm{d}x$. Evaluate all origin-destination pairs and implement the adjustment that reduces the objective function the most. Resume local neighborhood search to find a new local optimum.

Ultimately, we see local neighborhood search as being most effective as a subroutine within a hybrid heuristic that combines constructive approaches, local neighborhood search, and global searches. The exact nature of that hybrid heuristic would depend on the specific problem to which it was being applied.

[10] It is also possible that this affects which local minimum the procedure converges to.

[11] To minimize computation time, we would suggest evaluating the objective function using Monte Carlo integration or something similar.

3 Application: facilities in US cities

In this section, we apply the local search heuristic on facility location problems in the US cities of Chicago, Atlanta, and Los Angeles. While it may be of some interest to see what optimal spatial allocations look like in these environments, we can also use the computed optimal allocations as a benchmark to measure the inefficiency of the actual spatial allocations of the facilities in those cities.

In Section 3.1, we define our measure of spatial inefficiency. In Section 3.2, we note the modeling assumptions implied by the use of our measure. We discuss the sources of our data on actual spatial allocations in Section 3.3. Section 3.4 details the specific implementation of the heuristic in this environment. We present the results in Section 3.5.

3.1 Defining spatial inefficiency

Our measure of the inefficiency of a spatial allocation is based on the degree to which it yields higher transportation costs than an optimized spatial allocation with the same number of facilities. Therefore, we first look at the average distance to nearest facility for an observed allocation s.[12] $\bar{d}(s)$ is that average:

$$\bar{d}(s) = \frac{\int_X f(x) \cdot \min_i \left(d(x, s_i)\right) \mathrm{d}x}{\int_X f(x)\ \mathrm{d}x} \tag{1}$$

To compare across different applications, we look not just at the average distance but also the difference between the average distance and what the average distance would be in an optimized allocation, holding I fixed. (2) defines an I-optimal spatial allocation, s^{*I}, as one which minimizes the average distance.

$$s^{*I} \in \arg\min_{w \in \mathcal{S}_I} \bar{d}(w) \tag{2}$$

Finally, (3) defines a measure of spatial inefficiency, $\xi(s)$, as the difference between the average distance in s and that in s^{*I}, dividing by the latter to get a percentage difference that abstracts from units.

$$\xi(s) = \frac{\bar{d}(s) - \bar{d}(s^{*I})}{\bar{d}(s^{*I})} \tag{3}$$

Note that our measure of spatial inefficiency does not account for prices. This is by design as it allows us to apply it to a variety of spatial applications, only some of which will involve price competition. However, it also means that a social planner may prefer a spatially inefficient allocation over one that is spatially efficient if the former yields advantageous consequences in terms of prices.

3.2 Modeling assumptions

Most facility location problems are in two-dimensions with non-uniform customer distributions. To consider efficiency in these settings, we need data on the actual allocations

[12] Our methods are easily extended to quadratic transport costs and average squared distances.

of facilities and the customer distribution. We also need to be able to compute optimal allocations to form the reference points for our calculations of spatial inefficiency.

In this section, we evaluate allocations of supermarkets, hospitals, and fire stations in three major cities. We refer to these generally as facilities, rather than firms, to accommodate the inclusion of fire stations as well as non-profit hospitals, which may or may not behave as firms. We chose these three classes of facilities because we suspect that the mechanisms generating their allocations span the spectrum from regulated market competition (supermarkets) to central planning (fire stations), with hospitals somewhere in between.[13] By comparing $\xi(s^{\text{act}})$, ξ applied to actual allocations, across these classes of facilities, we can get a rough empirical comparison of spatial allocations resulting from competitive forces with others resulting from central planning. That is, we look to identify the spatial inefficiency resulting from the competitive mechanism by looking at a difference in differences—our evidence is not that $\bar{d}(s^{\text{act}})$ is greater than $\bar{d}(s^*)$ for supermarkets, though this is true and the difference is large, but rather that the percentage difference $\xi(s^{\text{act}})$ is much larger for supermarkets than for hospitals and fire stations.

Analysis with (1)-(3) in this context requires strong assumptions. First, we are viewing our consumers as static components of the environment, and identical but for location. Consumers, in their choices of residence, are perhaps as mobile as the firms.[14] Second, we are assuming that the firms are identical and can serve arbitrarily many customers, akin to Bertrand (1883). Third, we are treating the consumer as uniquely located, i.e. ignoring the possibility that a customer could frequent a distant supermarket with little inconvenience due to it being located on a commute.[15] In using Euclidean distances, we ignore transportation networks.[16] We also ignore important empirical infeasibilities, both physical and regulatory. These include infeasibilities of traveling in a straight line between pairs of locations as well as infeasibilities in facility placement due to lakes, zoning, etc.

We make these strong assumptions because they allow us to compute the optimal spatial allocations that we need to evaluate $\xi(s^{\text{act}})$. With all of these assumptions, even if an empirically observed spatial allocation were actually optimal for our measure given real-world constraints, we would still likely calculate a non-negligible $\xi(s^{\text{act}})$ for it given that our optimal s^{*I} is computed without these real-world constraints. However, to argue that the spatial allocations of supermarkets are inefficient, we show not just that their $\xi(s^{\text{act}})$ are large, but more importantly that they are much larger than those for allocations of hospitals and fire stations. Unless the assumptions are substantially more problematic for supermarkets than they are for hospitals and fire stations, differences in $\xi(s^{\text{act}})$ may still be

[13]While this vague conjecture is as deep as we go into the mechanisms generating the allocations in this section, a goal of the broader research agenda is to design or improve mechanisms to yield spatially efficient outcomes.

[14]While people may move frequently, housing stocks, and therefore population distributions, change slowly.

[15]For an interesting analysis that models commuting paths directly, see Houde (2012).

[16]Alternatively, one could perform this analysis as a version of *Problem 1* by aggregating demand and modeling the transportation network. In practice, a big challenge here is computing a distance matrix for the network. Google Maps, among others, has an API for computing these, but it limits the size to 25x25 (up to 316x316 daily for paid users). This is only sufficient for small or highly aggregated regions. Allen et al. (2015) takes another approach in computing the distance matrix manually by generating a heat map of the transportation network and running shortest path algorithms between origin and destination pairs on that heat map.

attributable to the difference in mechanisms generating the allocations.

3.3 Data

Our data on supermarket locations comes from OSM.[17] OSM's definition of a supermarket is a large store for groceries and other goods. It includes only full-service grocery stores, meaning most specialty and ethnic grocers are not included. Data for hospitals comes from two sources. Our primary source is the *Hospital General Information* dataset in Medicare's *Hospital Compare* data (Medicare, 2017). From this, we select all acute care and critical access hospitals that have emergency services. Because the geocoding in that database is incomplete, we cross-reference each hospital with its entry in the *Hospitals* database of the Department of Homeland Security's Homeland Infrastructure Foundation-Level Data (HIFLD) to get its location. Our fire station data also comes from HIFLD.[18]

We conduct our analyses on three cities: Atlanta, Chicago, and Los Angeles. We selected these cities both for certain desirable characteristics as well as technical reasons. For characteristics, we wanted cities of different sizes and in different regions to argue the external validity of our results. As for technical reasons, we wanted cities with statistical areas that were surrounded by areas of low population density—this allows us to analyze allocations on the city's statistical area and ignore users of the facilities outside of that area without introducing much error. Additionally, neighborhood search may struggle on cities that have sizable interior areas with zero customers, so we avoided cities with significant interior areas of water.[19] For each city, we conduct our analyses on both that city's metropolitan statistical area (MSA) and on a much smaller area roughly corresponding to the city lines.[20]

Finally, we take our population data from two sources. We account for population and population density at the level of census tracts with TIGER data from the 2010 US Census (U.S. Census Bureau, 2016). We then augment that with updated 2015 population estimates from Esri et al. (2017). TIGER data includes coordinates for each vertex of each census tract. We cannot use latitude and longitude coordinates directly for Euclidean distances because the distance of moving a degree North/South does not equate with that of moving a degree East/West. Instead, for each area, we convert the census tract vertices and facility locations to a metric stereographic projection centered in that area so that Euclidean distances between the points can be interpreted as distances in meters with only minimal error resulting from

[17] See OpenStreetMap contributors (2017).

[18] See Oak Ridge National Laboratory (2017) for HIFLD hospital data and TechniGraphics, Inc. (2010) for HIFLD fire station data.

[19] Having water form a boundary, as is the case with Chicago and Los Angeles, is not an issue. Small rivers and lakes are also fine. But cities like Boston, New York, and San Francisco would yield additional challenges.

[20] The boundaries of American cities are complicated—many cities contain non-city enclaves and include disconnected exclaves. We take convexifications of the actual cities. Chicago City includes all census tracts that intersect city boundaries with the exclusion of those around O'Hare airport (a city exclave) and with the inclusion of tracts in Norridge (a non-city enclave). Atlanta City includes all tracts in Fulton and DeKalb counties that intersect city boundaries, except one (FIPS: 13089020802) that juts out and includes a non-city exclave. Los Angeles City includes all tracts that intersect city boundaries, adding in several non-city enclaves (e.g. Beverly Hills and Santa Monica), and truncating the corridor leading down to the Port of Los Angeles by excluding all tracts south of Interstate 105.

the earth's curvature.[21]

3.4 Implementation of neighborhood search heuristic

One of the particularities of this application is that we have a demand function f that is uniform within each census tract and then discontinuous across census tract boundaries. For each neighborhood adjustment, we need to find all of the census tracts that intersect with the facility's Voronoi cell. Because census tracts are often quite complicated polygons with hundreds of vertices, computing the intersections between a Voronoi cell and each of the more than two thousand census tracts (in Chicago MSA, for example) is one of the more computationally taxing components of the procedure.[22] Since most of those intersections are empty, one could save significant time by searching for intersections for only a subset of the census tracts. We found it sufficient to filter the full list of census tracts by first looking for intersections between bounding boxes of the tracts and bounding boxes of the Voronoi cells—this is much faster. For tracts for which these bounding-box intersections are non-empty, we compute the actual intersections.

Figure 4 shows the Voronoi cell of a supermarket in the Chicago MSA. The figure's interior lines represent boundaries of census tracts—the cell intersects roughly 30 tracts. The census tracts are shaded red with darker reds representing higher population densities. The black region to the right of the figure is Lake Michigan. The black triangle is the current location of the supermarket. To calculate the geometric median, the black circle, we draw points randomly from each tract-cell intersection, with the number drawn from each determined by the area of the tract-cell intersection multiplied by the population density of the tract, all divided by the sum of these products across all tract-cell intersections. This gives us a percentage of our sample size to draw from each tract-cell intersection. We then construct the sample of discrete points and find the geometric median of the sample.

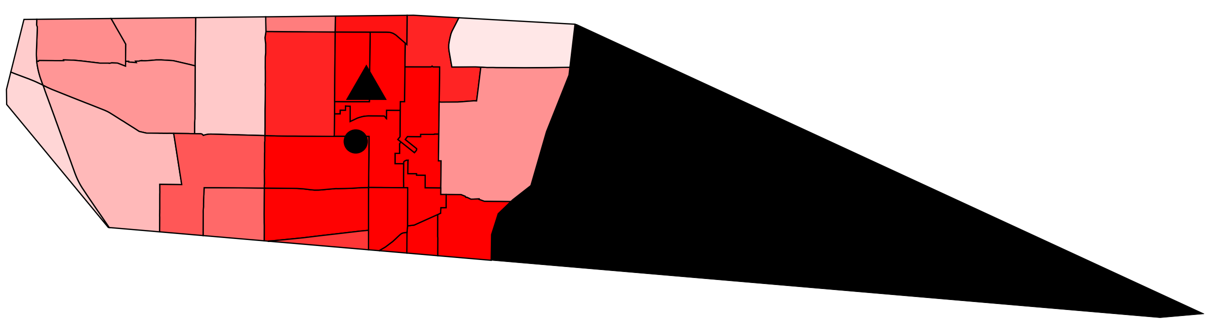


Figure 4: Voronoi cell of Super Fresh Market in Waukegan, IL

The multiplicity of local optima is a significant issue in this environment—when we run the heuristic here from different initializations, we converge to many different fixed points.

[21] With our projection on Chicago, for instance, the calculated distance between two points 200 kilometers apart will have an error of about 76 meters.

[22] Region intersection functionality is built in to Mathematica but can be slow for complicated polygons.

The reason for this lies in the non-uniformity of the customer distribution and the fact that our algorithm involves only local adjustments. The algorithm essentially pulls each facility towards its nearest population mass, as if the model were gravitational, and then optimizes the allocation of facilities over each population mass. What it does not do as well is distribute facilities efficiently across the population masses. As an example, in the Chicago MSA, there may be one fixed point with 4 facilities serving Aurora and 2 serving Joliet, another with the numbers reversed, and another with 3 serving each. Because the population density is low between the two suburbs, facilities are not necessarily pulled across from one suburb to another even if it would be optimal to do so.

Our approach to dealing with fixed point multiplicity is to run the algorithm from fifty randomly generated initial allocations as well as the actual one for each optimization, selecting the best. For about half of our optimizations, the actual allocation proved a better initialization than any of the fifty randomly generated initializations. This is not surprising—we show below, in Figure 5, that actual facility allocations match population densities fairly well. Insofar as they have the right number of facilities in each community, the actual allocations serve as good initializations for the algorithm. We could also have taken other approaches, as we describe in Section 2.2.

While the heuristic does not technically require evaluation of the objective function, we need to be able to evaluate it for two purposes. First, it is an obvious metric over which to define a tolerance and determine convergence of the heuristic—we proposed an alternative, the distances of the adjustments, in Section 2. Second, we need to evaluate average distance for both the actual and the optimal allocations to measure $\xi(s^{\text{act}})$. To this end, we compute the average distance over the entire city or MSA by computing the average distance within each census tract through numerical integration and then taking a weighted average across the census tracts based on their populations. Importantly, the approximation of a region by drawing points randomly that we describe above and use in our implementation of the algorithm is not relevant to the final evaluation of average distances for both the observed and optimal allocations. So while it is true that the allocation we call optimal is only approximately so, the precision of our evaluations of the average distances is only limited by the minor potential imprecision of numerical integration.

3.5 Results

In Table 1, we present summary data on the city and MSA of each of our three cities. We include their areas in square miles, their populations, and the number of supermarkets, hospitals, and fire stations contained in each.

Table 2 shows the average mileages to facilities for each facility, each region, and each of three allocations. $\bar{d}(s^{\text{ran}})$ is the average mileage across a thousand randomly generated allocations with facility numbers equal to those in the corresponding actual/observed allocations. We then show the average distances in the actual allocations, $\bar{d}(s^{\text{act}})$, and those in the optimal allocations, $\bar{d}(s^{*})$, which were generated by our algorithm. In Table 3, we report the implied spatial inefficiencies for each facility in each city.

For hospitals and fire stations, the actual allocations are better than the average of the randomly generated allocations for all six regions. For supermarkets, the actual allocations are worse than the randomly generated allocations for the cities but better for the MSA's,

Table 1: Summary statistics on regions and facilities therein

Region	**Sqmi**	**Pop. (mil.)**	**Tracts**	**S**	**H**	**F**
(1) Atlanta City (AC)	163	0.47	130	19	4	38
(2) Atlanta MSA (AM)	8835	5.53	951	174	37	502
(3) Chicago City (CC)	227	2.78	803	76	25	96
(4) Chicago MSA (CM)	6304	9.56	2210	259	80	759
(5) LA City (LC)	536	4.44	1109	100	29	137
(6) LA MSA (LM)	4754	13.14	2923	277	86	535

Table 2: Average mileages to facilities for random, actual, and optimal allocations

	Supermarkets			**Hospitals**			**Fire Stations**		
Region	$\bar{d}(s^{\text{ran}})$	$\bar{d}(s^{\text{act}})$	$\bar{d}(s^{*})$	$\bar{d}(s^{\text{ran}})$	$\bar{d}(s^{\text{act}})$	$\bar{d}(s^{*})$	$\bar{d}(s^{\text{ran}})$	$\bar{d}(s^{\text{act}})$	$\bar{d}(s^{*})$
(1) AC	1.58	1.90	1.01	3.58	3.47	2.24	1.10	0.86	0.71
(2) AM	3.58	3.59	2.01	7.83	5.13	4.24	2.11	1.47	1.18
(3) CC	0.93	1.09	0.60	1.67	1.49	1.06	0.82	0.61	0.54
(4) CM	2.74	2.40	1.39	5.01	3.06	2.34	1.58	0.90	0.75
(5) LC	1.20	1.43	0.73	2.33	2.06	1.36	1.02	0.75	0.63
(6) LM	2.12	1.77	1.00	3.83	2.30	1.77	1.52	0.82	0.68

Table 3: Spatial inefficiency of actual facility allocations

	$\boldsymbol{\xi(s^{\text{act}})\%}$		
Region	**S**	**H**	**F**
(1) Atlanta City	87	55	22
(2) Atlanta MSA	79	21	25
(3) Chicago City	82	41	14
(4) Chicago MSA	73	30	21
(5) LA City	96	51	19
(6) LA MSA	78	30	20

with the exception of Atlanta MSA, for which the two are very close. This is a first indication of our general point that supermarkets are inefficiently allocated, at least for the purposes of consumer transportation costs.

The inefficiency of the supermarket allocations is yet more stark in Table 3. In each of the six regions, the spatial inefficiency associated with the supermarket allocations is significantly greater than that of hospital and fire station allocations.[23] They still represent average distances between 56% and 96% longer than the optimal configurations. The difference between being 1.09 miles and 0.6 miles from a supermarket in the city of Chicago, for instance, is fairly significant for consumers without a car.[24]

One might question whether our measure of inefficiency would correlate with the number of facilities, and thus a comparison between two allocations of different I would not be meaningful. In this case, however, there are more supermarkets in each region than there are hospitals and less supermarkets than fire stations, and both hospital and fire station allocations are more efficient. Another contention would be that our assumption that supermarkets can serve arbitrarily many customers is influencing the results. But this is also true of hospitals and fire stations,[25] for which we find less inefficiency.

We include figures to directly compare the actual allocation of supermarkets with the optimal allocation. Figures 5a and 5b show the population densities of census tracts in Chicago MSA as well as the actual spatial allocation of supermarkets. The latter shows that supermarkets do tend to locate in high-density areas, which is efficient, but they also tend to cluster, which is not efficient by our measure. Figures 6c and 6d show distances of all points in Chicago MSA for the actual and optimal allocations respectively. It is immediately apparent that most of the MSA is much closer to their nearest supermarket with the optimal allocation. Of course, what matters is not how close points in the region are to their nearest supermarket, but rather how close people are. Therefore, we also provide *scarcity plots* that reveal where there is significant population that is distant from a supermarket. To this end, in Figures 5e and 5f we map over the region a density plot in which the shade of coloring at a point is determined by the product of the population density of the containing tract and the distance from that point to its nearest supermarket.[26]

Figures 6c-5f also show that a consequential difference between the actual and optimal allocations is that a significant number of supermarkets have been pushed out from the city center to Chicago's suburbs and exurbs in the optimal allocation. This is, indeed, optimal in terms of minimizing average distances.[27] Figure 5a is a little misleading in its portrayal of population density as it seems to suggest that there is very little population

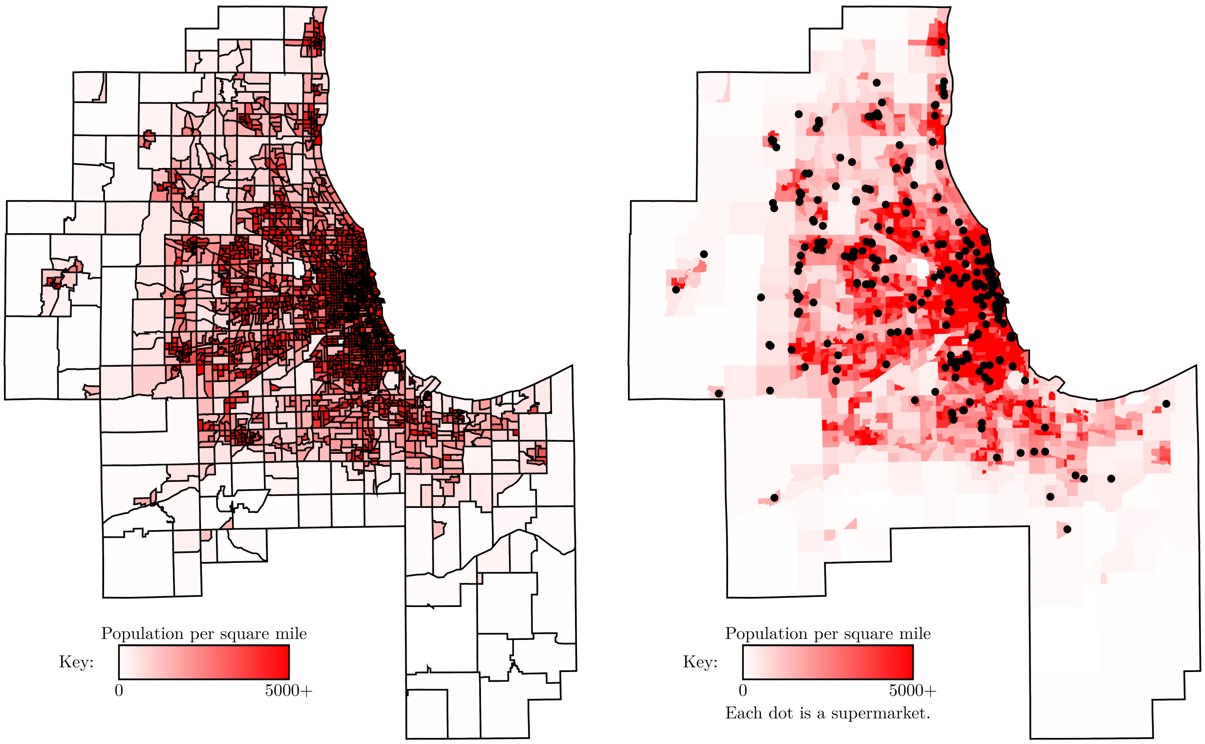
[23]The inefficiency in the supermarket allocations appears to come from their tendency to cluster near areas of higher population density. In this sense, they are at least clustered where the people are.

[24]While we are using supermarkets as an example to illustrate a broader point about spatial allocations resulting from competition, our analysis overlaps here with literature on food deserts and access. Distances affect food choice, not just transportation costs. For a review of this literature, see Walker et al. (2010). Also see ERS-USDA (2017) for a related spatial atlas.

[25]Fire stations' duties may scale more with area than population.

[26]We subtract one mile from the distance in the figure coloring as otherwise high-density areas suggest scarcity even quite close to a supermarket.

[27]Allen et al. (2015) comes to a similar conclusion through a very different analysis—they find that the welfare of Chicago residents would increase if more area was allocated to residential usage in the central business district and more area was allocated to businesses in the outlying neighborhoods.



(a) Census tracts and pop. density (b) Supermarkets and pop. density

Figure 5: Chicago MSA

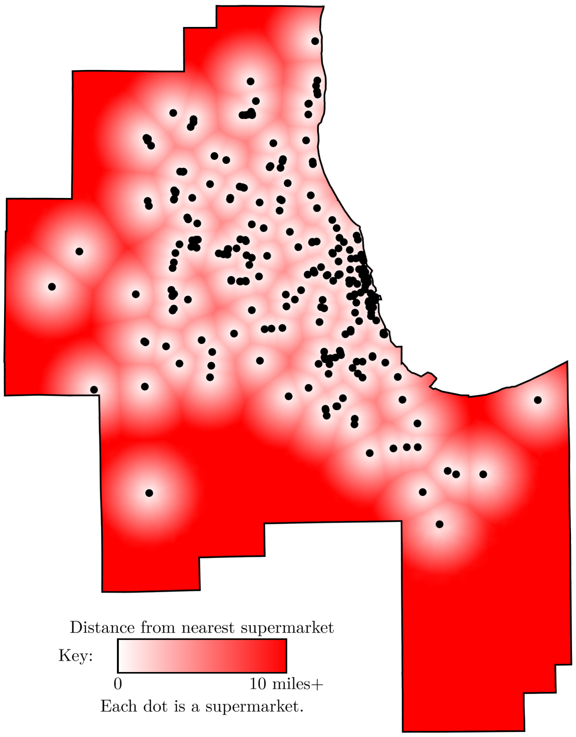
at the periphery of the MSA. But the almost-white census tracts on the periphery actually have roughly the same population as census tracts closer into the city—census tracts are defined to have similar populations. There are even some significant towns of concentrated population density in some of these peripheral census tracts. But the population densities for these tracts, which determines the shading, is still almost negligible because of their size. In any case, it should come as little surprise that the optimal allocation attempts to serve these communities.[28] The fact that the actual allocation does not hints at the sort of principle of minimum differentiation clustering that was posited in Hotelling (1929).[29] All of this notwithstanding, we did the analyses on the city boundaries also to show that not all of our efficiency gains in the optimal allocations come from movements of facilities from the inner city to suburbs and exurbs.[30] Figures for other facilities in Chicago MSA and for supermarkets in Chicago city are provided in Appendix A.

We can also estimate the cost of the spatial inefficiency in supermarket allocations in monetary terms. Our optimal allocation reduces the average distance to nearest supermarket from 2.4 miles to 1.4 miles. Since we have no evidence as to the feasibility of the optimal allocation, let us compare the actual with an allocation of supermarkets that has the same

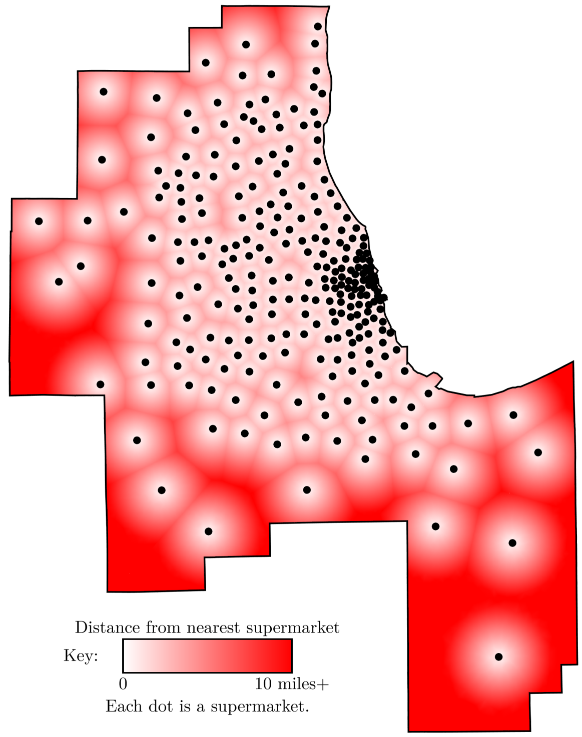
[28]We do not consider different rates of car ownership between urban and sub/exurban tracts—if our goal was to generate truly optimal allocations, considering this would be necessary.

[29]Remember that our optimal allocation is for the I from the actual allocation. We do not suggest that this I is optimal, nor that policymakers should seek to implement s^*. Our focus is the inefficiency of s^{act}.

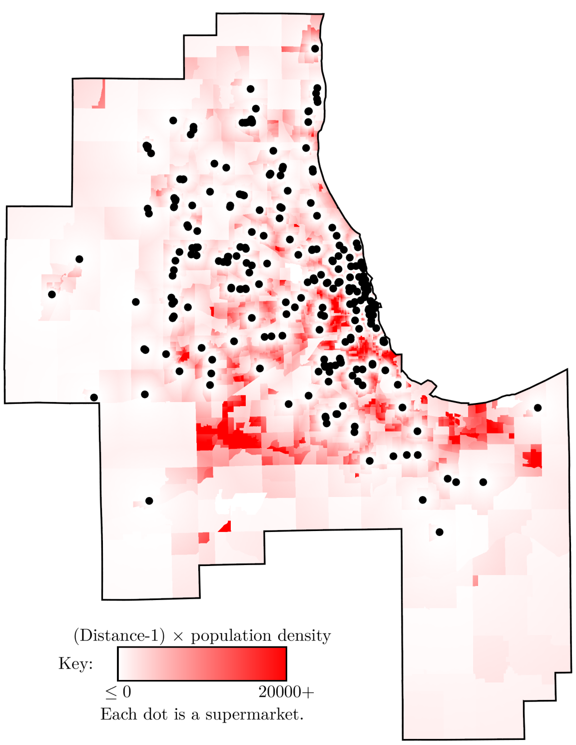
[30]Additional gains come largely from declustering.



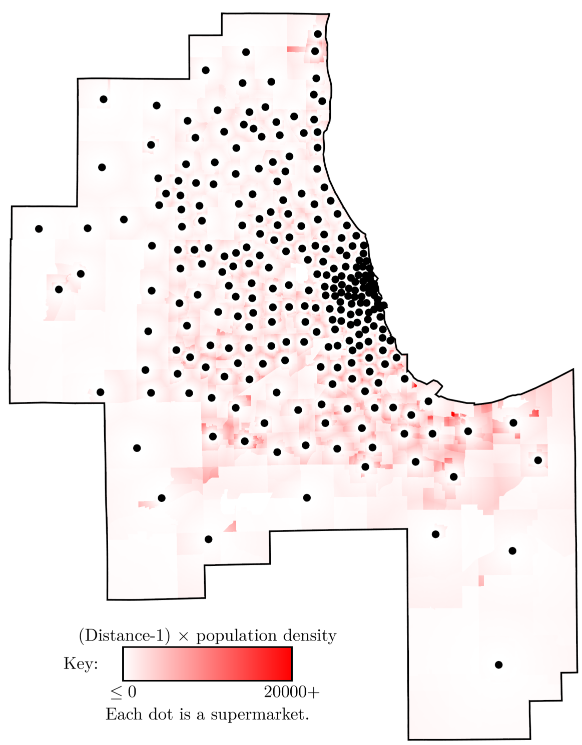
(c) Distance to supermarket in s^{act}



(d) Distance to supermarket in s^{opt}



(e) Supermarket scarcity in s^{act}



(f) Supermarket scarcity in s^{opt}

Figure 5: Chicago MSA (continued)

spatial inefficiency of 30% as Chicago hospitals, the less efficient of the other two facility allocations. Matching that 30% inefficiency means bringing the distance from 2.4 miles down to about 1.8 miles. For a single round-trip, this represents an average savings of 1.2 miles. If we multiply this by the roughly 3.4 million households in the Chicago MSA and suppose that a member of each household visits a supermarket weekly, we get a reduction of about 214 million miles traveled annually. If we multiply this by the IRS standard mileage rates used to calculate the deductible costs of operating an automobile for business, 53.5 cents per mile, we would value these miles at about 114 million dollars.[31] This estimate includes only direct transportation costs, not time costs. Even for the transportation costs, we think it is highly conservative—reducing the Euclidean distance to supermarkets by 0.6 miles usually reduces the travel distance by more than 0.6 miles, and we suspect per-mile travel costs to be greater that 53.5 cents per mile for those without cars.[32]

Further research is required to determine the degree to which inefficiencies we find for supermarket allocations can be generalized to other allocations resulting from competition. We also do not isolate the degree to which current regulation may either limit or exacerbate these inefficiencies. But we believe that the tendency of similar firms to cluster is fairly general. In retail and dining with limited consumer information, this may be efficient—consumers enjoy shopping at multiple retail stores at one location and diners can choose a location and then compare dining options. But when firms are homogeneous, as we suggest is largely the case for supermarkets in our data set, we view clustering as an inefficient phenomenon.

Insofar as the clustering of homogeneous firms is inefficient, we are interested in policies that could limit clustering. Our empirical analysis has no direct policy conclusions. We do not propose optimizing the spatial allocations of supermarkets or gas stations because this could yield additional deadweight loss from price competition—we suspect that the clustering of homogeneous firms has advantageous effects on prices. Our optimized allocations also take the number of firms as given, and we have not considered the different sizes of our facilities and the value of land.[33] To consider policy, we would need a predictive model of spatial competition. Auerbach and Dix (2018) discusses this challenge.

4 Conclusion

In this paper, we have proposed a neighborhood search heuristic for the p-median problem with continuous demand. We have discussed issues relating to its implementation and the embedding of neighborhood search within a hybrid heuristic. We then applied the heuristic to facility location problems in US cities to find that actual spatial allocations of supermarkets in cities do relatively poorly in minimizing transportation costs for consumers.

[31] Analogous calculations for Atlanta MSA and Los Angeles MSA yield 127 million dollars and 114 million dollars, respectively.

[32] We grant that some households get groceries from vendors other than full-service supermarkets. While this may reduce their transportation costs, it has other costs in terms of food choice and public health.

[33] Our optimized allocation has supermarkets further from the city center, on average, than the actual allocation, which likely means that the optimized allocation would occupy less valuable land than the actual.

References

Allen, T., Arkolakis, C., and Li, X. (2015). Optimal city structure. Technical report.

Auerbach, S. and Dix, R. (2018). Myopia in dynamic spatial games. Technical report.

Bertrand, J. (1883). Review of theorie mathematique de la richesse sociale and of recherches sur les principles mathematiques de la theorie des richesses. *Journal des savants*, 67:499–508.

Bose, P., Maheshwari, A., and Morin, P. (2003). Fast approximations for sums of distances, clustering and the fermat–weber problem. *Computational Geometry*, 24(3):135–146.

Brimberg, J., Hansen, P., Mladenović, N., and Taillard, E. D. (2000). Improvements and comparison of heuristics for solving the uncapacitated multisource weber problem. *Operations Research*, 48(3):444–460.

Cohen, M. B., Lee, Y. T., Miller, G., Pachocki, J., and Sidford, A. (2016). Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 9–21. ACM.

Cooper, L. (1963). Location-allocation problems. *Operations research*, 11(3):331–343.

Daskin, M. S. (2011). *Network and discrete location: models, algorithms, and applications.* John Wiley & Sons.

Drezner, Z. and Hamacher, H. W. (2001). *Facility location: applications and theory.* Springer Science & Business Media.

ERS-USDA (2017). Food access research atlas. https://www.ers.usda.gov/data-products/food-access-research-atlas/.

Esri, TomTom North America, Inc., and US Census Bureau (2017). USA census tract boundaries (shapefile). Retrieved from `https://www.arcgis.com/home/item.html?id=ca1316dba1b442d99cb76bc2436b9fdb`.

Fekete, S. P., Mitchell, J. S., and Beurer, K. (2005). On the continuous fermat-weber problem. *Operations Research*, 53(1):61–76.

Francis, R. L., Lowe, T. J., Rayco, M. B., and Tamir, A. (2009). Aggregation error for location models: survey and analysis. *Annals of Operations Research*, 167(1):171–208.

Geoffrion, A. M. (1979). Making better use of optimization capability in distribution system planning. *AIIE Transactions*, 11(2):96–108.

Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459.

Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475.

Hotelling, H. (1929). Stability in competition. *The Economic Journal*, 39(153):41–57.

Houde, J.-F. (2012). Spatial differentiation and vertical mergers in retail markets for gasoline. *American Economic Review*, 102(5):2147–82.

Iri, M., Murota, K., and Ohya, T. (1984). A fast voronoi-diagram algorithm with applications to geographical optimization problems. In *System modelling and optimization*, pages 273–288. Springer.

Kariv, O. and Hakimi, S. L. (1979). An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560.

Kuehn, A. A. and Hamburger, M. J. (1963). A heuristic program for locating warehouses. *Management science*, 9(4):643–666.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Love, R. F., Morris, J. G., and Wesolowsky, G. O. (1988). *Facilities location: models & methods*. North-Holland.

Maranzana, F. E. (1963). On the location of supply points to minimize transportation costs. *IBM Systems Journal*, 2(2):129–135.

Medicare (2017). Hospital compare: Hospital general information. Retrieved from `https://data.medicare.gov/Hospital-Compare/Hospital-General-Information/xubh-q36u`.

Megiddo, N. and Supowit, K. J. (1984). On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196.

Mladenović, N., Brimberg, J., Hansen, P., and Moreno-Pérez, J. A. (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939.

Murat, A., Verter, V., and Laporte, G. (2010). A continuous analysis framework for the solution of location–allocation problems with dense demand. *Computers & Operations Research*, 37(1):123–136.

Newell, G. F. (1973). Scheduling, location, transportation, and continuum mechanics: some simple approximations to optimization problems. *SIAM Journal on Applied Mathematics*, 25(3):346–360.

Oak Ridge National Laboratory (2017). Hospitals. Data retrieved from Homeland Infrastructure Foundation-Level Data (HIFLD), `https://hifld-geoplatform.opendata.arcgis.com/datasets/hospitals`.

Okabe, A., Boots, B., and Sugihara, K. (1992). *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, Inc.

Okabe, A. and Suzuki, A. (1997). Locational optimization problems solved through voronoi diagrams. *European Journal of Operational Research*, 98(3):445–456.

OpenStreetMap contributors (2017). `https://www.openstreetmap.org`. Feature shapefiles retrieved from `http://osm2shp.ru`.

TechniGraphics, Inc. (2010). Firestations. Data retrieved from Homeland Infrastructure Foundation-Level Data (HIFLD), `https://hifld-geoplatform.opendata.arcgis.com/datasets/fire-stations`.

U.S. Census Bureau (2016). 2016 tiger/line shapefiles (machine-readable data files). Retrieved from `https://www.census.gov/geo/maps-data/data/tiger-line.html`.

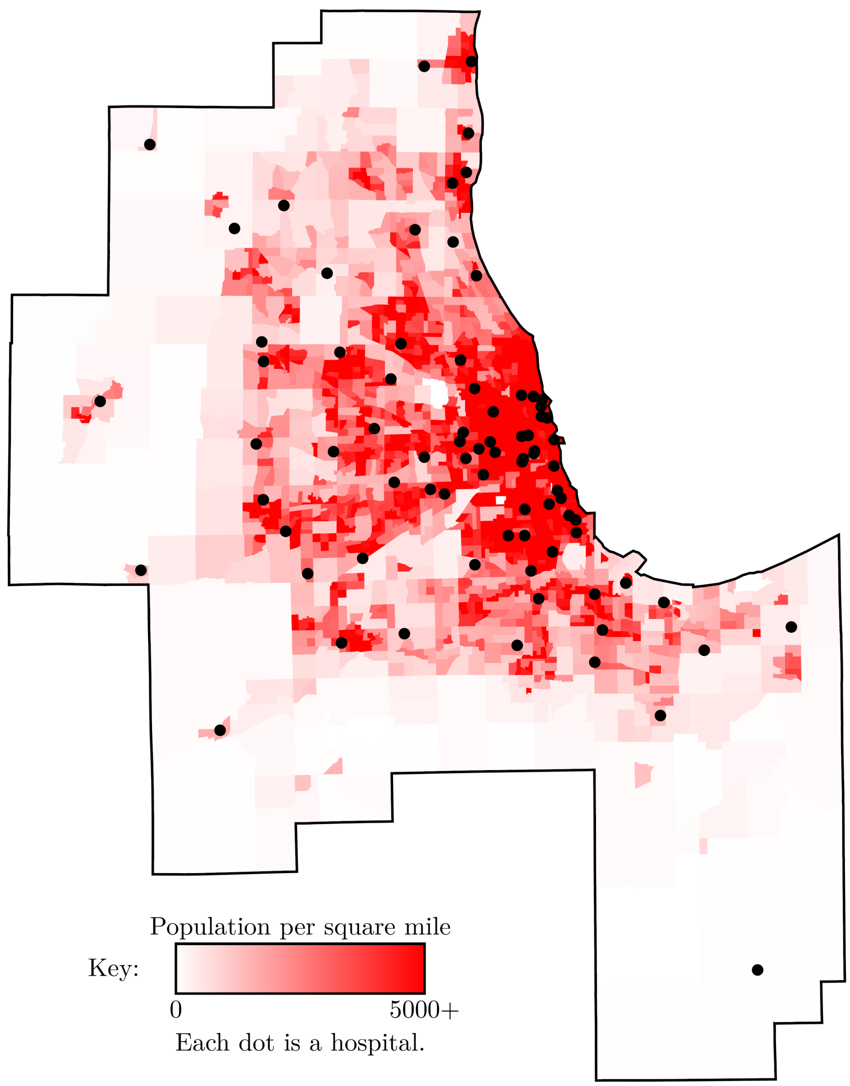
Walker, R. E., Keane, C. R., and Burke, J. G. (2010). Disparities and access to healthy food in the united states: A review of food deserts literature. *Health & place*, 16(5):876–884.

Weber, A. (1929). *Theory of the Location of Industries*. University of Chicago Press.

Weiszfeld, E. (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386.

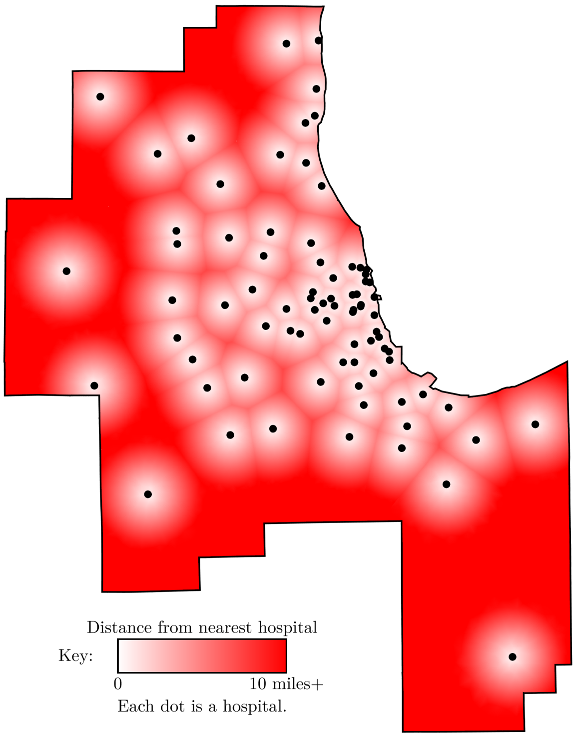
A Additional figures from Section 3.5

In this appendix, we provide a few extra figures for comparison with those presented in the body of the paper. First, we continue Figure 5 to show allocations of hospitals and fire stations in Chicago MSA. Then, in Figure 6, we show the supermarket allocations in Chicago city as an additional comparison to those in Chicago MSA. Figures for other regions and features are available upon request.

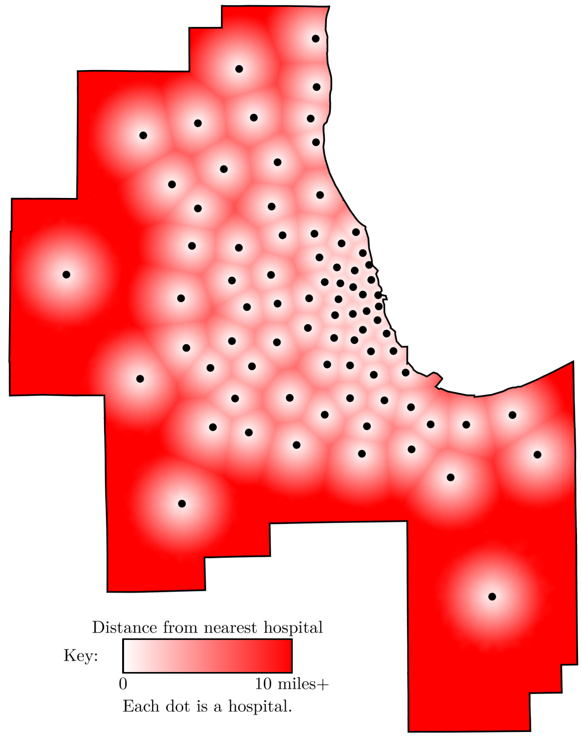


(g) Hospitals and pop. density

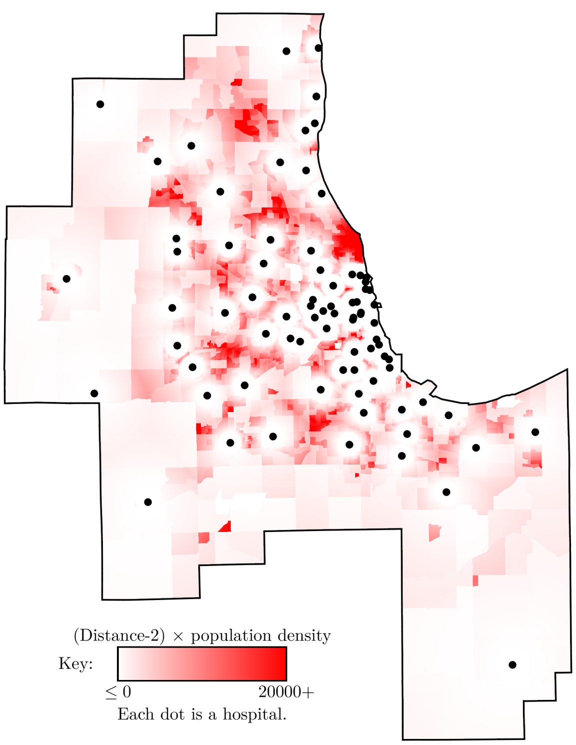
Figure 5: Chicago MSA (continued)



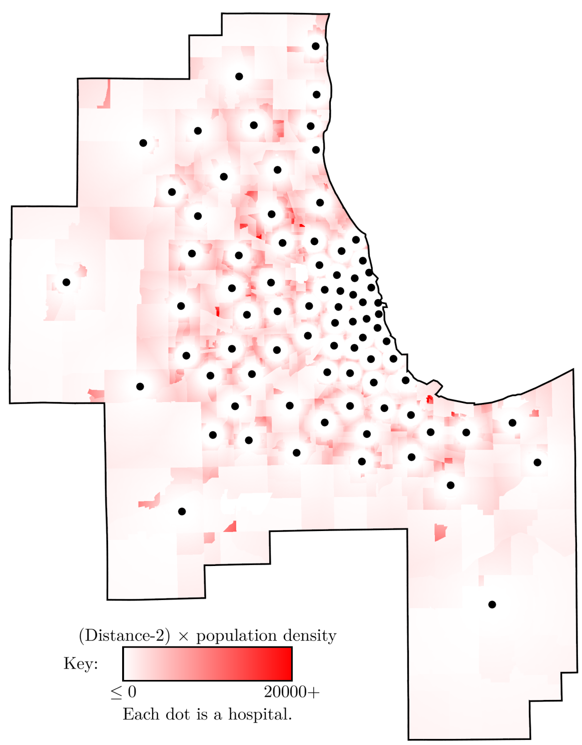
(h) Distance to hospital in s^{act}



(i) Distance to hospital in s^{opt}

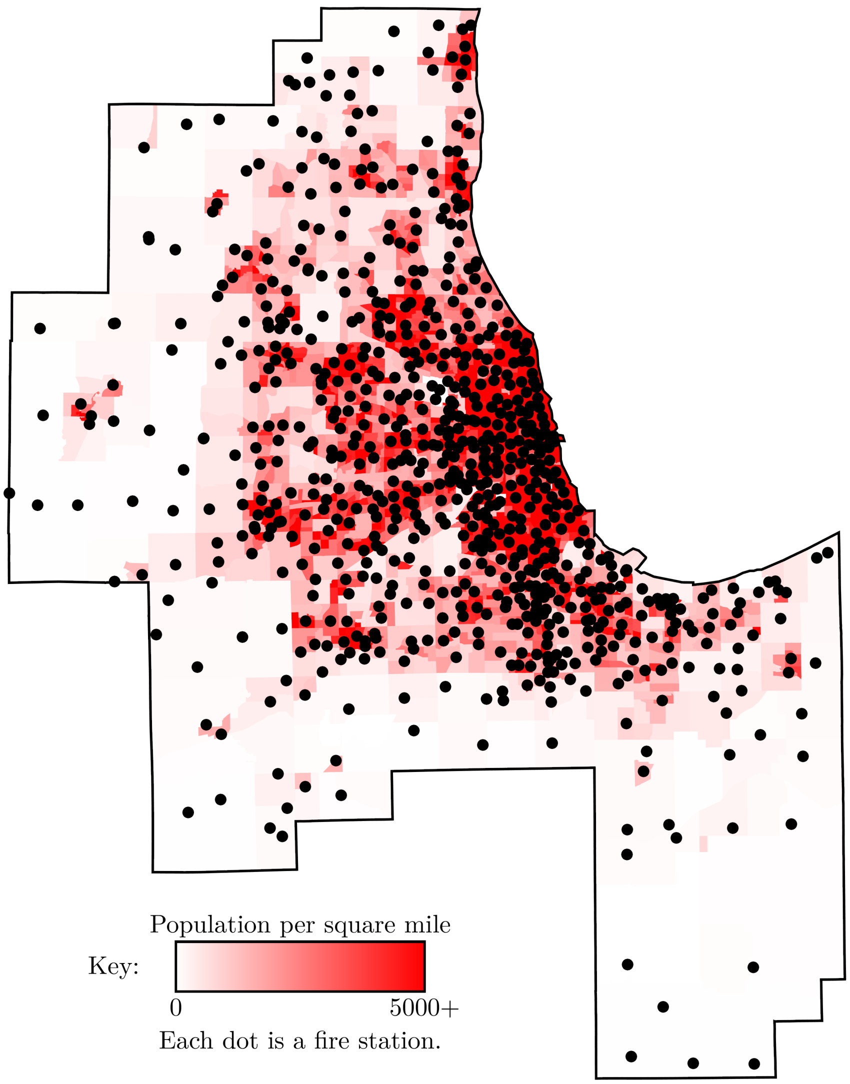


(j) Hospital scarcity in s^{act}



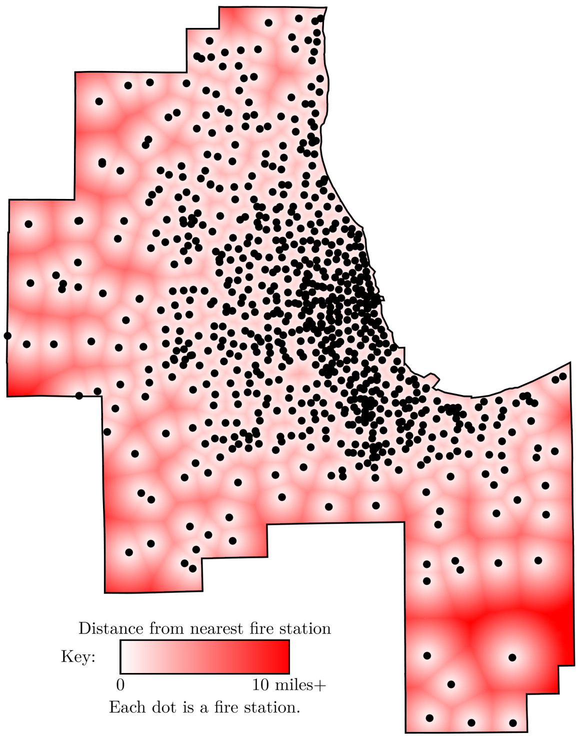
(k) Hospital scarcity in s^{opt}

Figure 5: Chicago MSA (continued)

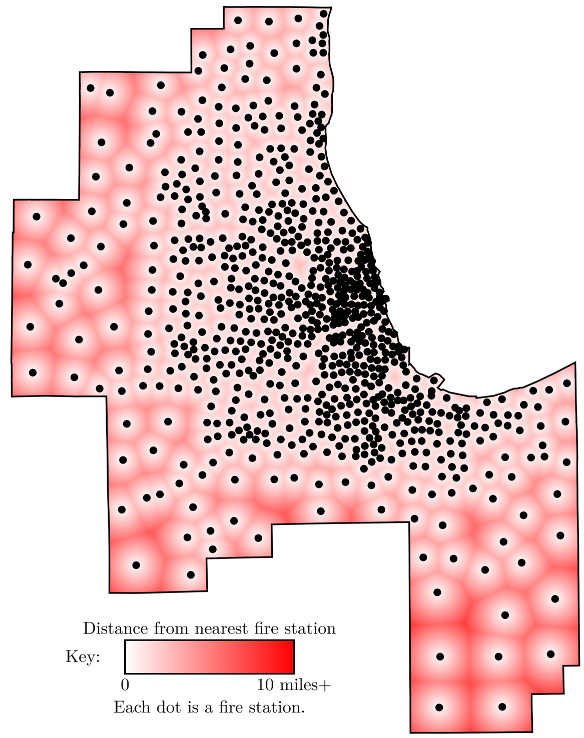


(l) Fire stations and pop. density

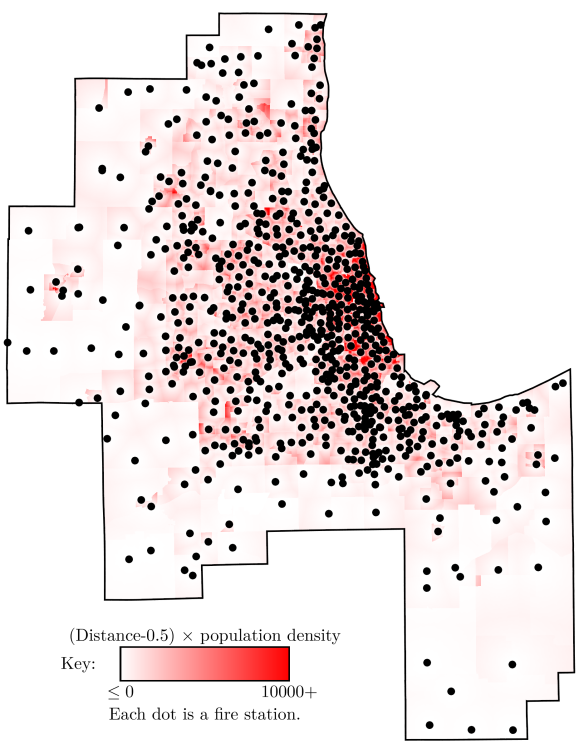
Figure 5: Chicago MSA (continued)



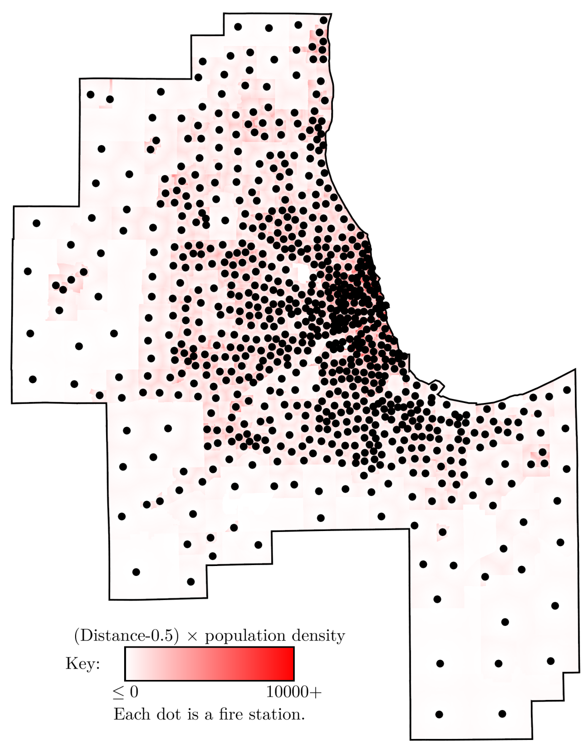
(m) Distance to fire station in s^{act}



(n) Distance to fire station in s^{opt}

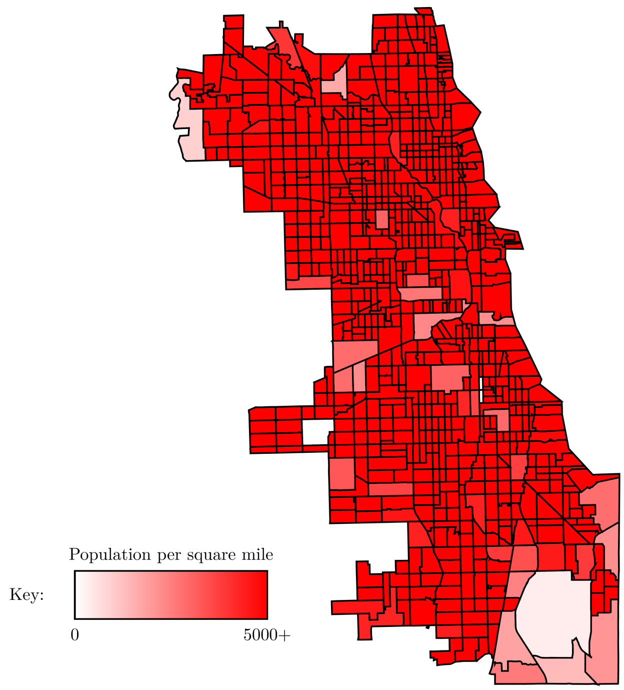


(o) Fire station scarcity in s^{act}

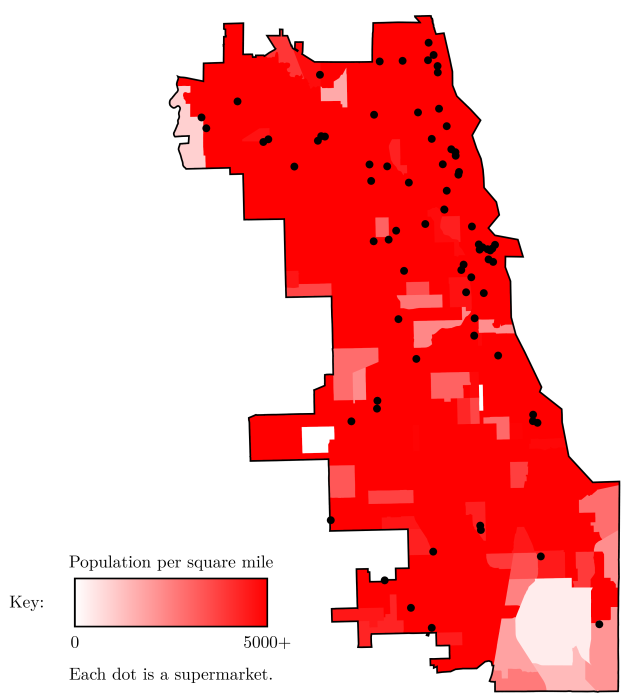


(p) Fire station scarcity in s^{opt}

Figure 5: Chicago MSA (continued)

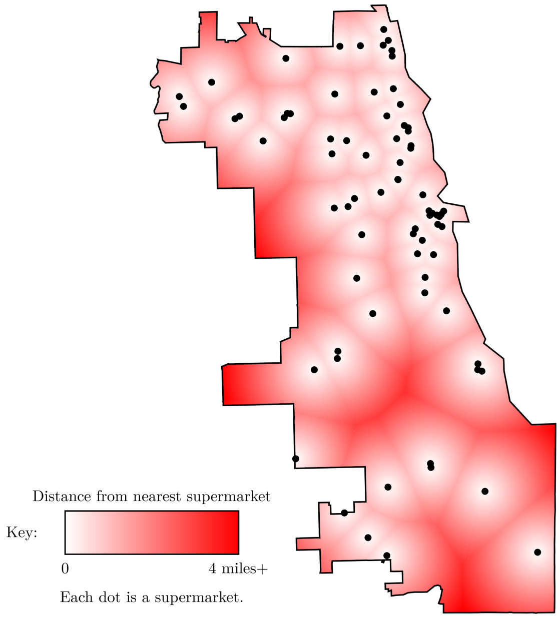


(a) Census tracts and pop. density

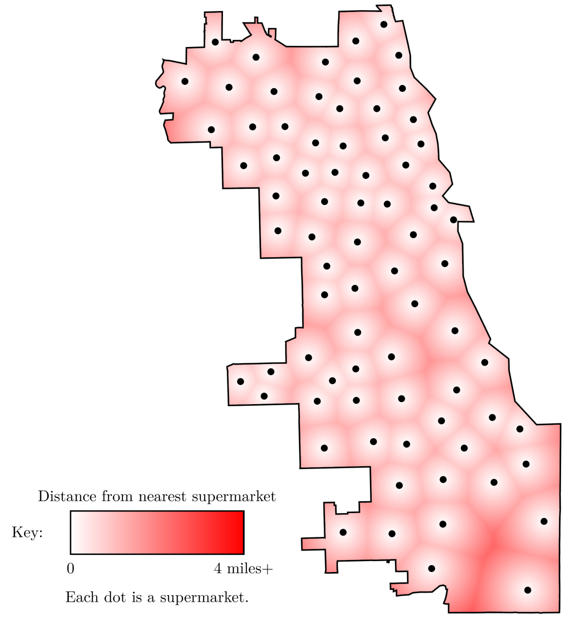


(b) Supermarkets and pop. density

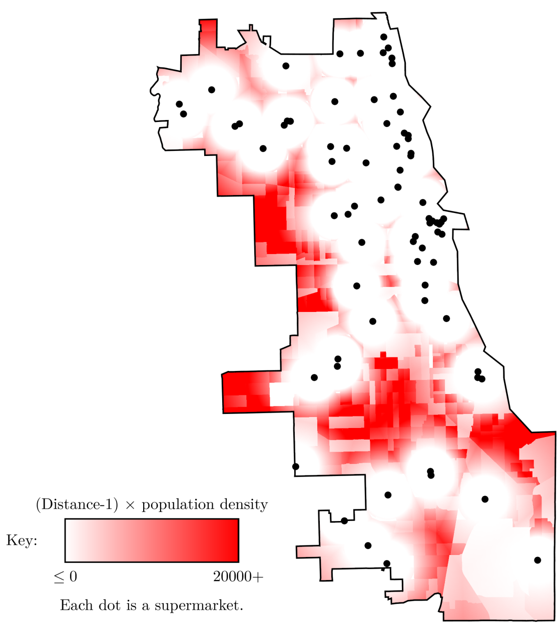
Figure 6: Chicago city



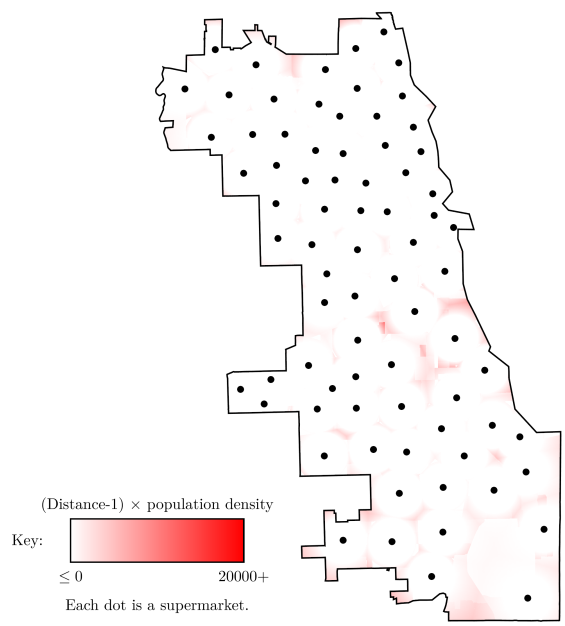
(c) Distance to supermarket in s^{act}



(d) Distance to supermarket in s^{opt}



(e) Supermarket scarcity in s^{act}



(f) Supermarket scarcity in s^{opt}

Figure 6: Chicago city (continued)