Chapin Hall at the University of Chicago

Policy research that benefits children, families, and their communities

MEMORANDUM ON IMPROVING FUNDING ALLOCATION PROCESSES WITH COMPUTER-ASSISTED METHODS

INTRODUCTION OF THE CHICAGO ALLOCATION TO NEIGHBORHOOD-ORIENTED PROGRAMS FOR YOUTH (CANOPY) ALGORITHM

INTRODUCTION

These notes describe a method of allocating Head Start and Preschool For All (PFA) funding using, in part, a computer optimization algorithm--for the sake of reference termed the Chicago Allocation to Neighborhood Oriented Programs for Youth algorithm or "CANOPY"--that Chapin Hall has piloted, which can be adapted to assist with the allocation process. We explain the rationale for this pilot below, show the results of an example of its implementation, and suggest how an instance of CANOPY can be adapted to the Head Start and PFA allocation task.

This pilot is in response to expectations that the allocation process for Head Start and PFA funding may be highly time consuming and/or difficult to execute. The difficulties of this task stem from the ambitious charge of the allocation process to:

1. Make a large number of decisions, allocating funding for tens of thousands of seats across hundreds of providers;
2. Simultaneously balance service to the highest needs community as well as provide allocation to the highest-quality providers;
3. Meet a complex overall goal, taking into account at least two separate objectives for the allocation (i.e. the Needs and Language Indices identified by the City Needs Assessment Taskforce), and possibly other less tangible objectives such as multiple dimensions of provider quality, provider service records, and information about the neighborhoods;
4. Execute these steps in a timely, transparent manner, that uses consistent reasoning from start to finish.

CANOPY was designed with the specific intention of meeting all of the challenges listed above. It uses a type of trial and error algorithm for improving allocations (described in more detail below), representing a natural human approach to solving a complex problem. But, compared to a human process, it can work millions of times faster and do so while balancing a large number of factors involved in the goal.

Its benefits include:

- **Intuitive Logic** - The Chicago Allocation to Neighborhood Oriented Programs for Youth (CANOPY) algorithm is based on natural human problem solving logic, where the involvement of computers makes that human logic work faster and more systematically, especially in the case of complex problems. (The algorithm is described in more detail below.)

- **Transparency** - To work, CANOPY necessarily requires a clearly stated, transparent position about what factors are considered in the allocation and what weight they receive. We emphasize that CANOPY can only provide computer *assistance* to the allocation process, and cannot substitute for careful, thoughtful planning.

- **Consistency** - A human decision making process may have inconsistencies along the way if the process is very time consuming, involves consideration of a large number of factors, and if there is any individual discretion involved in making a decision. CANOPY can test and compare millions of alternative allocations in a matter of minutes, and uses the exact same decision-making process from start to finish.

- **Can Meet Rich, Multi-factor Goals** - In general, stakeholders desire a nuanced consideration of many factors when making allocation decisions. A human decision process may either have to simplify those considerations into a simpler "rule-of-thumb" for the sake of having a transparent and tractable procedure for determining allocations, or may have to use human discretion to balance many factors simultaneously, which comes at the expense of reduced transparency and consistency. CANOPY's use of computer power gives it the ability to consider more factors than is possible by a human brain, at the same time as following human instructions and human logic to apply those considerations.

- **More Efficient Time Use for All Staff** - Once CANOPY is set up, it can run in a matter of minutes. This immediately avoids the need for long and tedious involvement of many staff across many days. Additionally, solving the labor-intensive part of the process increases time for analytical and creative time in the process. Stakeholders can instead spend time carefully crafting--and testing--rich schemes for how to consider many factors in the allocation.

An essential note is that CANOPY does not (and *cannot*) substitute for a high quality process of discussions and planning. CANOPY is simply a tool for extending this human reasoning and sensibilities to a clearly-stated problem. Indeed, its most critical input is a clear statement of explicit goals for how to decide between alternative allocations, which is an essential step in any process requiring transparency. In addition to helping solve complex problems without resorting to simplifications of goals, the fact that it works so quickly means that committee discussions can get rapid feedback. This makes it possible to continually develop and improve the planning process, whereas a highly time-consuming allocation process might only be able to be run once or twice, with little possibility for reconsideration.[1]

Below, this memo introduces the intuition behind CANOPY, provides an example of how it works based on a simple pilot application, and provides a brief discussion of how it might be fit into the broader allocation process.

INTUITION BEHIND CANOPY

CANOPY formally draws on computer-based simulated annealing methods, which represent an intuitive process to find optimal allocations in the case of complex problems. In cases where no straightforward process exists for making an optimal set of allocation decisions, one slow but familiar approach that a human might undertake is to start with an initial candidate allocation, use intuition about how it can be made better, experiment by making small changes to the allocation, and judge whether those changes made things better or worse. This type of approach is never undertaken in practice because it is typically too unsystematic, too time consuming to do by hand, and too hard to keep track of what's already been tried.

CANOPY uses this same intuitive process of experimentation, but addresses those limitations by using computing power. CANOPY starts with a candidate allocation, randomly selects one seat allocated to a given providers, and considers reassigning that seat to another randomly chosen provider. Based on instructions that it's been given about how to judge how good a given allocation is, it can decide whether or not to pursue the change that it tried. By contrast to the equivalent human process, CANOPY can explore millions of alternative allocations in just a few minutes, uses a consistent sensibility for how to compare those alternatives, and can remember all of the details of what it has tried and what the best solutions were.

CANOPY's speed and systematicity also allow it to do things that it is hard for a human to do. At the beginning of CANOPY's run, the algorithm will be more willing to experiment in the sense that it may, with some probability, choose to pursue reallocations that do not improve the objective. That is, the algorithm is openminded to see where each of a wide range of alternative paths may eventually lead, since a very different configuration may be much better. As the algorithm continues to run, it becomes gradually less willing to pursue unproductive avenues, and focuses on fine tuning the best allocations that it has already found.

PILOT APPLICATION FOR CANOPY

[1] Once CANOPY is set up for the problem, it would be fast enough to be run multiple times in a given meeting, each time receiving different trial specifications of the goal, in order to integrate a discussion of how different sets of goals produce different optimal allocations.

As a demonstration of concept, Chapin Hall has applied CANOPY to a simple city allocation problem. While simple, this example shows the algorithm's intuition, speed and effectiveness, and also the type of progress reporting that can be generated as it runs.

The pilot application involves a problem of allocating 20,000 seats across Chicago community areas. Although CANOPY is capable of handling much richer problems, the goal for this exercise was kept simple for interpretability and ease of judging that the algorithm worked. In this case, CANOPY was instructed to find the allocation that maximized an "toy" objective where:

- Seats allocated to even-numbered community areas score no points.
- Seats allocated to odd-numbered community areas do score points, and the value of allocating a seat to a given community is highest when there are few seats allocated there (i.e. the community is underserved), and is lower if a large number of seats have already been allocated there (i.e. the community is better served).[2]
- The total score for a given allocation equals the total number of points across all community areas.

This setup was chosen to capture notions that some allocations are higher priority than others (in the differential treatment of even and odd community areas), and that general conditions of supply should be taken into account (in that the algorithm can be asked to prioritize underserved areas). Given this setup, the optimal solution is to have the exact same number of seats allocated to odd-numbered community areas, and none to even-numbered community areas.

The following figures show the progress of CANOPY as it works to identify the optimal allocation. In the pilot, CANOPY was instructed to take 300,000 total steps, which guaranteed that the algorithm find what we knew (by our setup) to be the optimal allocation. As CANOPY ran, it generated a series of maps to display the best allocation that it had found after every 30,000 steps. These maps are attached at the end of this document.

These maps offer a view of the logic underlying CANOPY's operation. The algorithm runs rapidly, testing close to 10,000 alternative allocations every 10 seconds. Although its operation continually made improvements with respect to its objective, its path to the optimal allocation was not linear. In some cases, the algorithm overshot the number of seats that would be there in the optimal setting. This is especially clear with community areas 3, 35 and 59, as those are some of the most centrally located areas, and thus served as conduits for the algorithm to move seats around from one end of the city to another as it attempts betters solutions. As the algorithm reached the end of the 300,000 steps we told it to take, CANOPY focused much less on exploring new paths and much more on fine-tuning its best solution. The result is that by the 300,000th step and about 34 seconds of run time, it had found what we know is the uniquely optimal allocation.

---

[2] Formally, the value of having ***N*** seats allocated to a community area equals the square root of ***N***.

More complex problems may require more steps, but the rapid operation makes it possible to easily solve expanded problems. Alternate test runs of the pilot had attempted 5,000,000 steps, which ran in less than 10 minutes on a standard Dell laptop. If called for, this algorithm can be significantly sped up using advanced computing resources available to Chapin Hall.

The next section discusses how CANOPY can be applied to the richer allocation process for Head Start and Preschool For All funding.

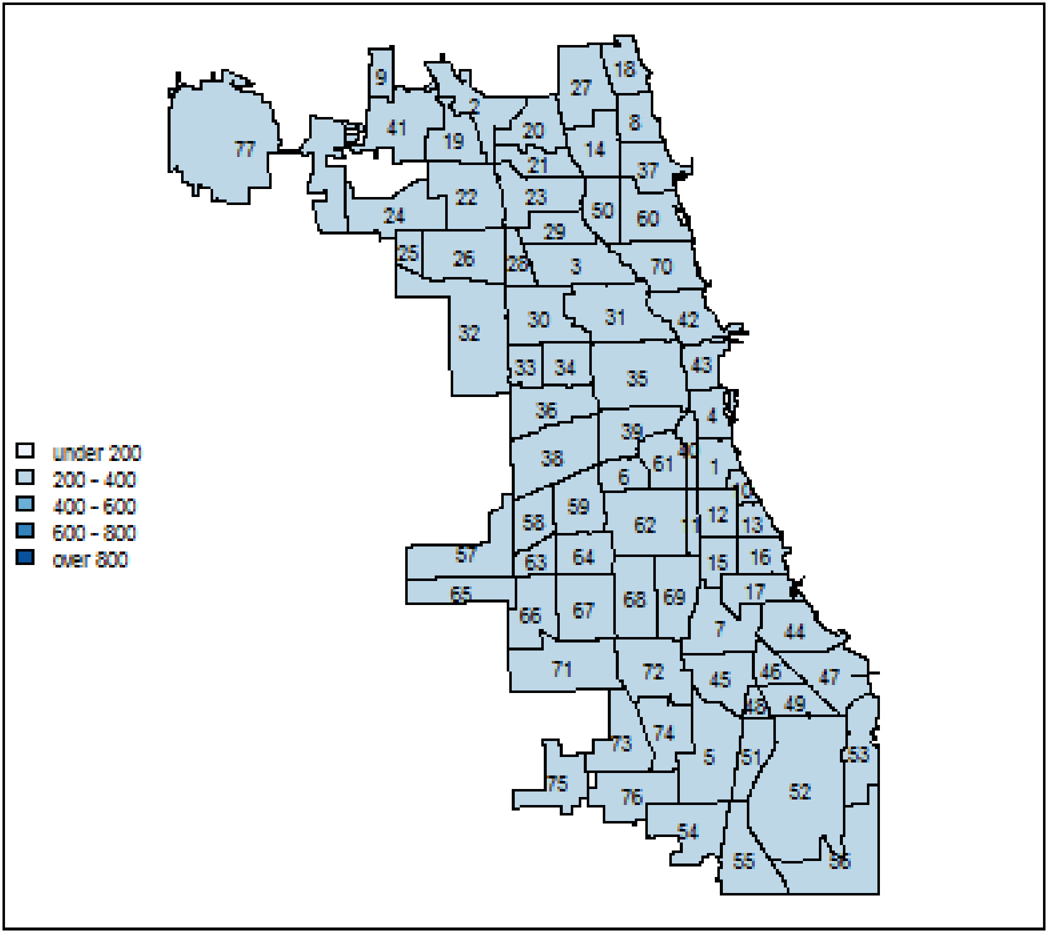
USE OF CANOPY IN THE BROADER ALLOCATION PROCESS

As suggested above, CANOPY must necessarily be embedded in a high quality process of discussions and planning among stakeholders. CANOPY can replace the time-consuming, labor intensive part of the allocation process, but cannot substitute for careful planning.

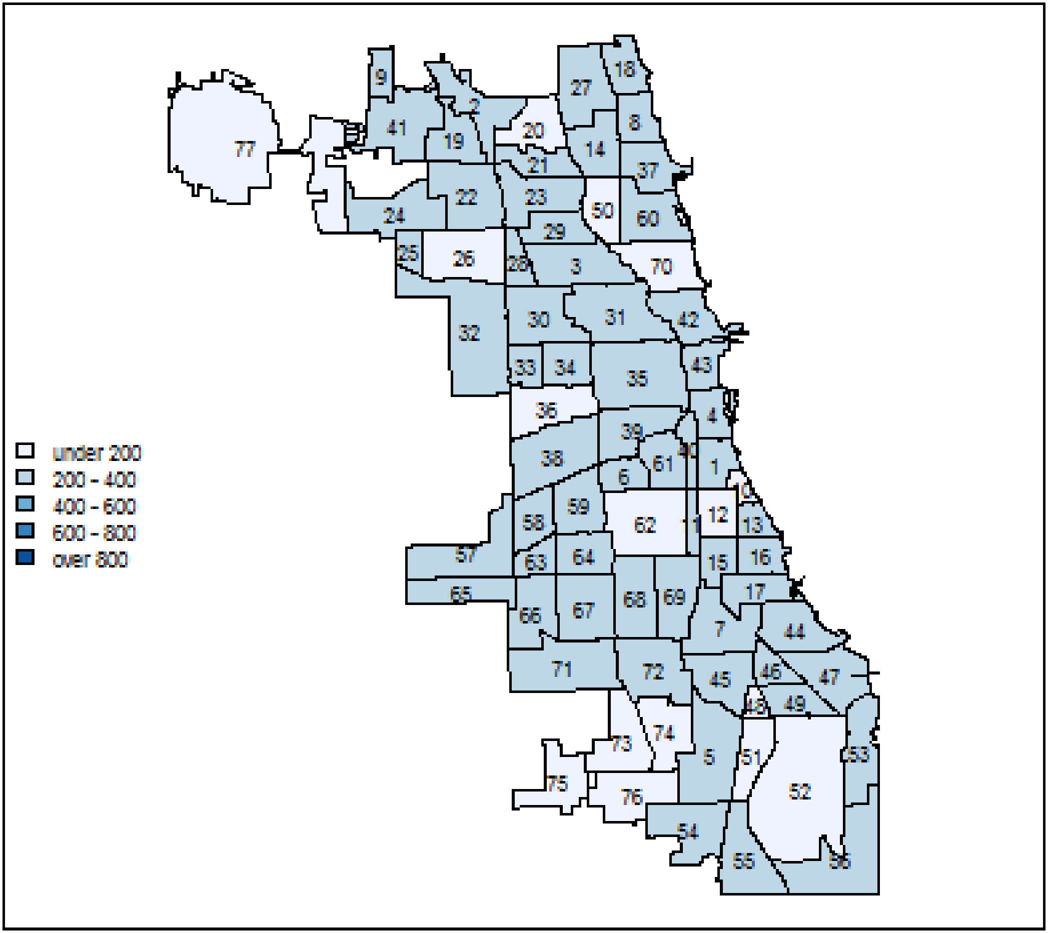
Because of the computational power on its side, CANOPY can simultaneously and flexibly take into account a large and nuanced set of considerations that stakeholders may indicate as desirable to the allocation process. Examples of factors that can be considered by CANOPY:

1. **Child and family needs**. This can be measured either by a single index, or by many dimensions.
2. **The overall quality of a provider**. This can be a single summary measure of provider quality, or can be multidimensional.
3. **Provider characteristics besides quality that are relevant**. For example, if desired, some additional consideration can be given to providers that have a long track record of service, in order to lean towards continuity of service when other factors are equal.
4. **The interaction of provider services with child and family characteristics or circumstances**. If desired, the allocation of a seat to a given provider can be given extra consideration if the provider has characteristics that benefit the population of families it expects to serve. For example, it may get additional points:
    a. if it has bilingual staff and is in an area with many households with limited English capability,
    b. if it has staff with counseling training and serves a population with high exposure to violent crime; or
    c. if the provider is highly accessible by transportation in areas where households have limited housing options.
5. **Neighborhood characteristics**. It may be desirable to give added consideration to neighborhoods that have been traditionally underserved, or are in a particular location in the city such as providers that are near large employment centers.

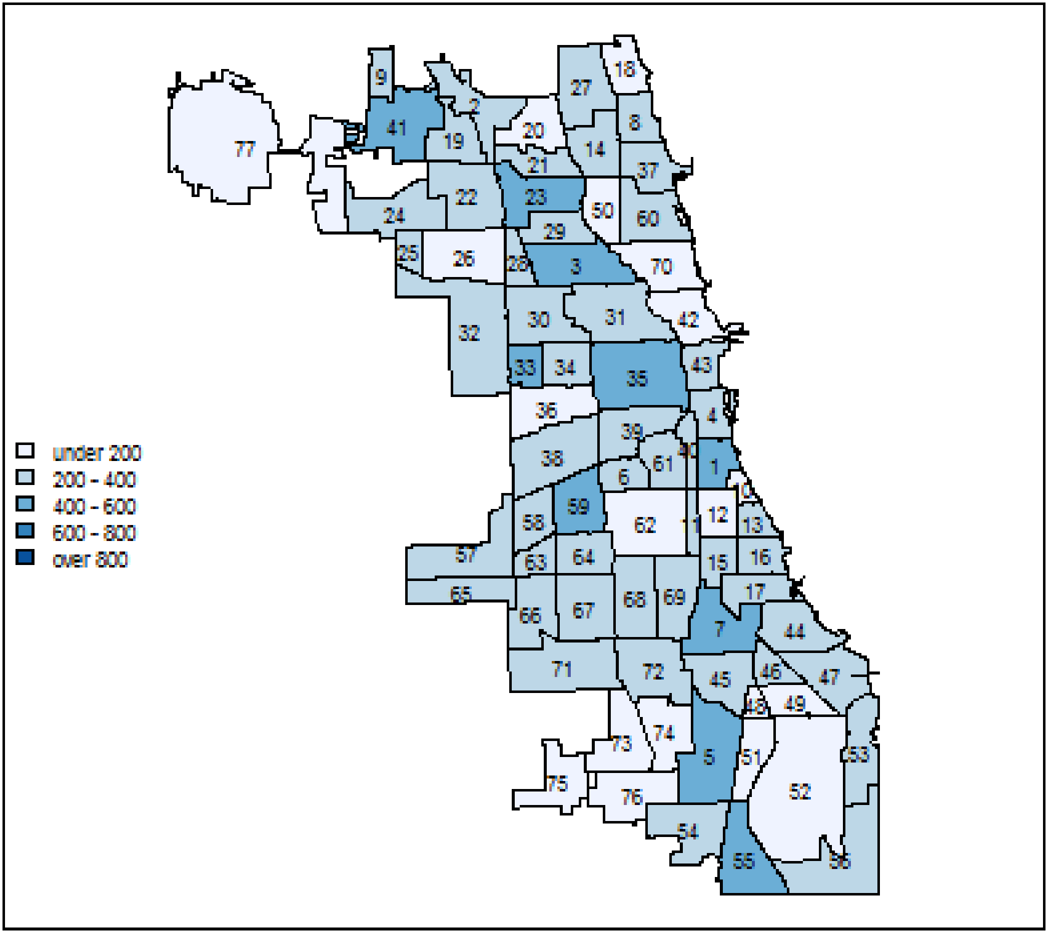
Best Allocation Found After 0 Seconds and 0 Steps



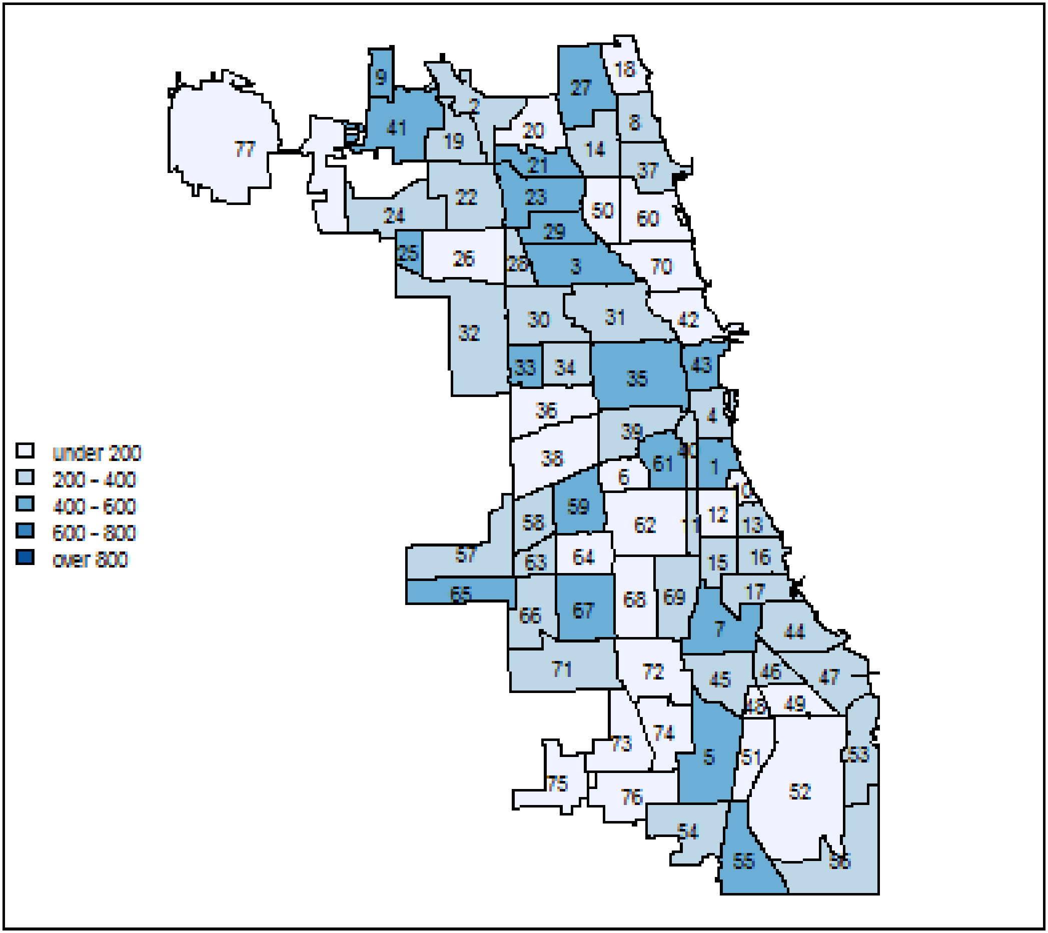
Best Allocation Found After 3 Seconds and 30,000 Steps



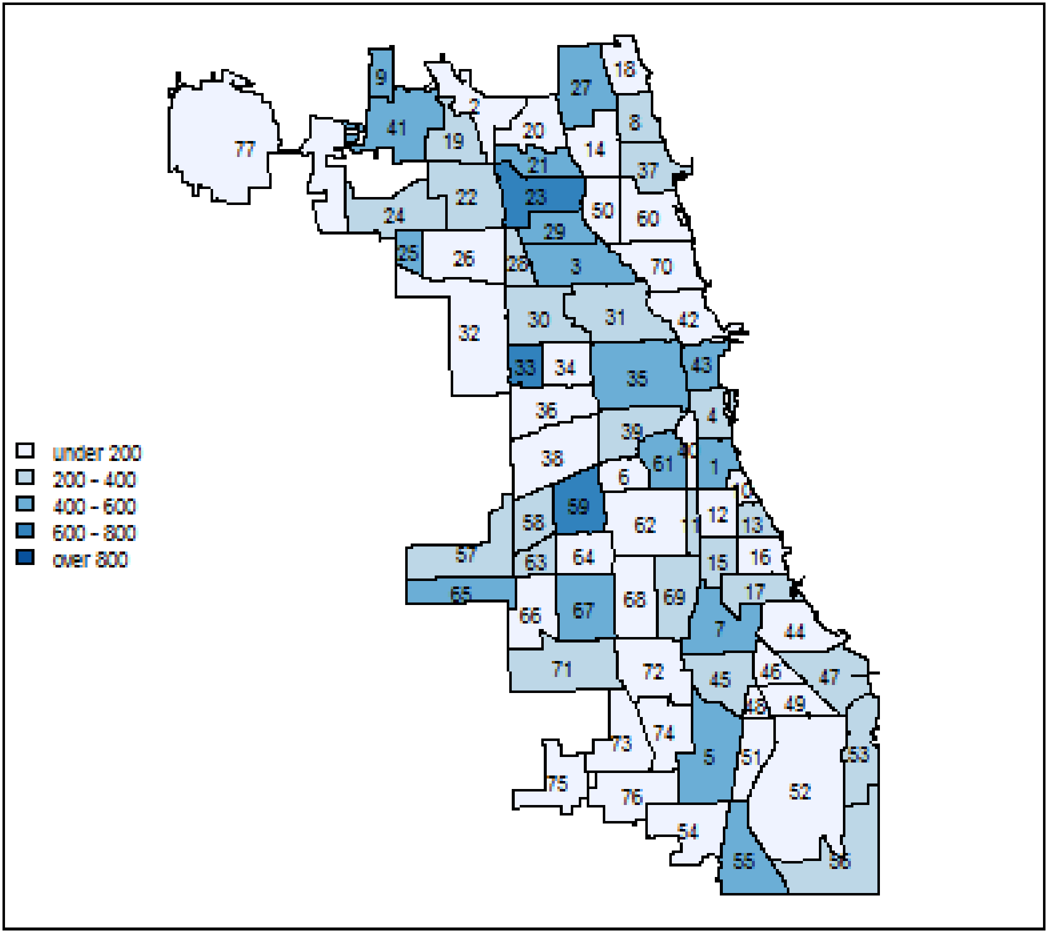
Best Allocation Found After 7 Seconds and 60,000 Steps



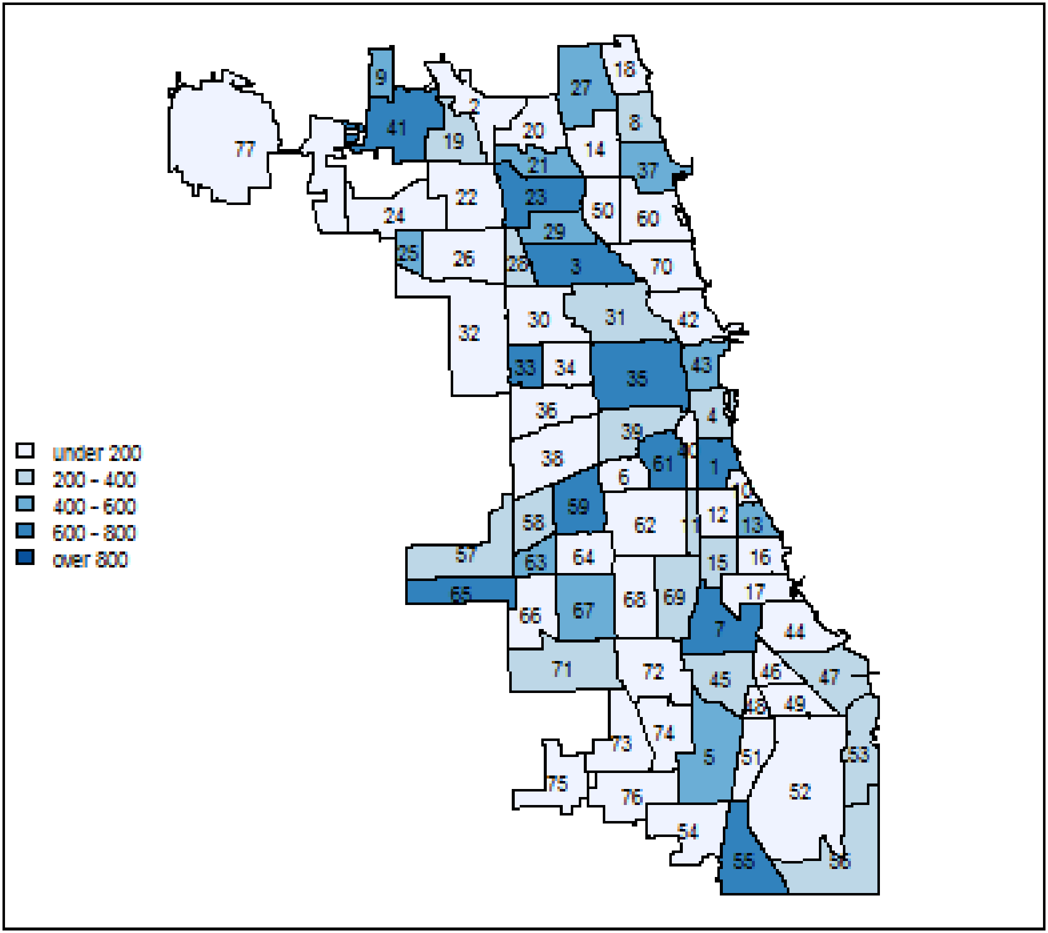
Best Allocation Found After 10 Seconds and 90,000 Steps



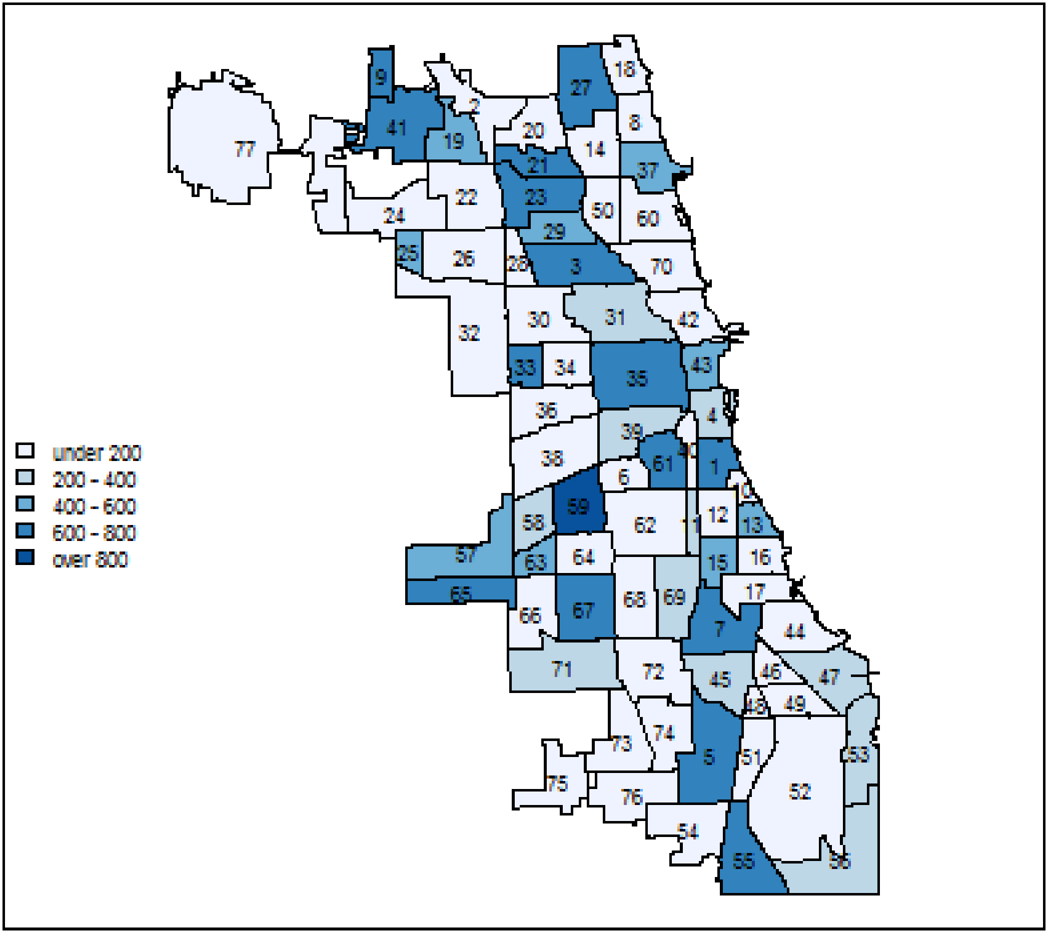
Best Allocation Found After 13 Seconds and 120,000 Steps



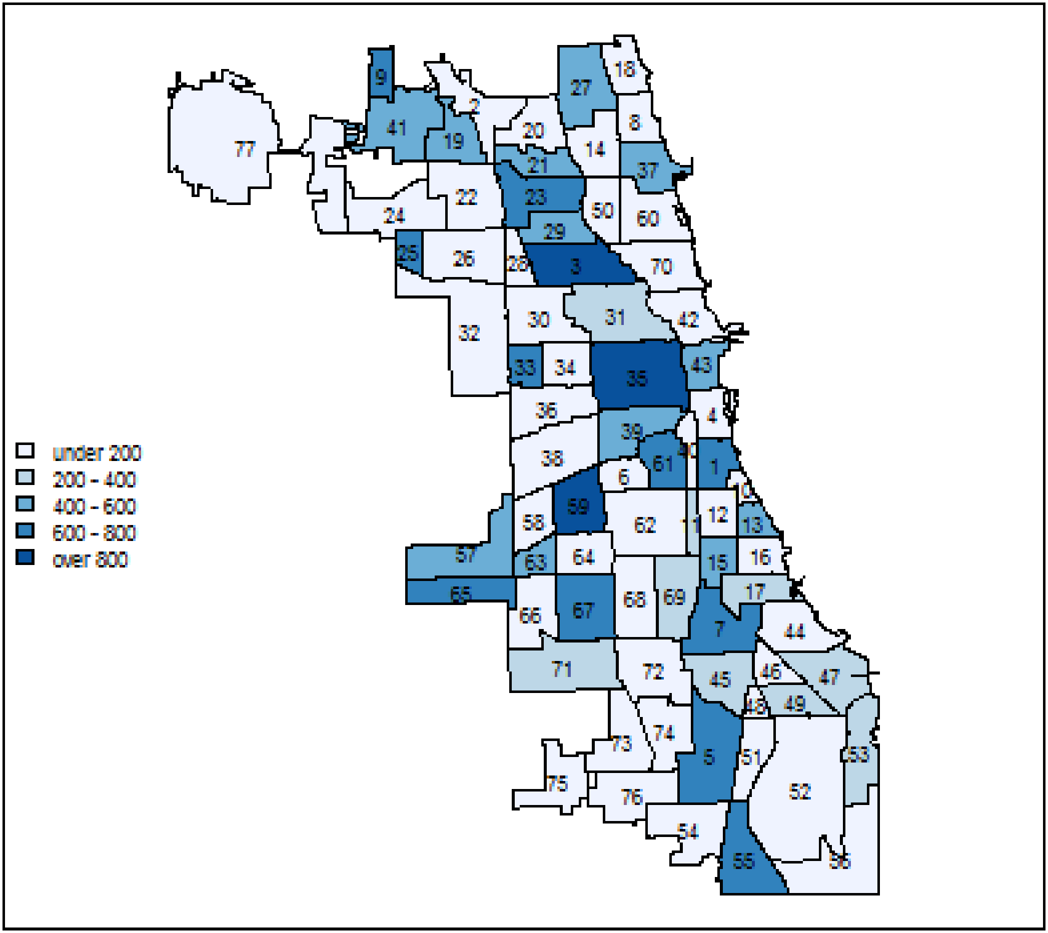
Best Allocation Found After 17 Seconds and 150,000 Steps



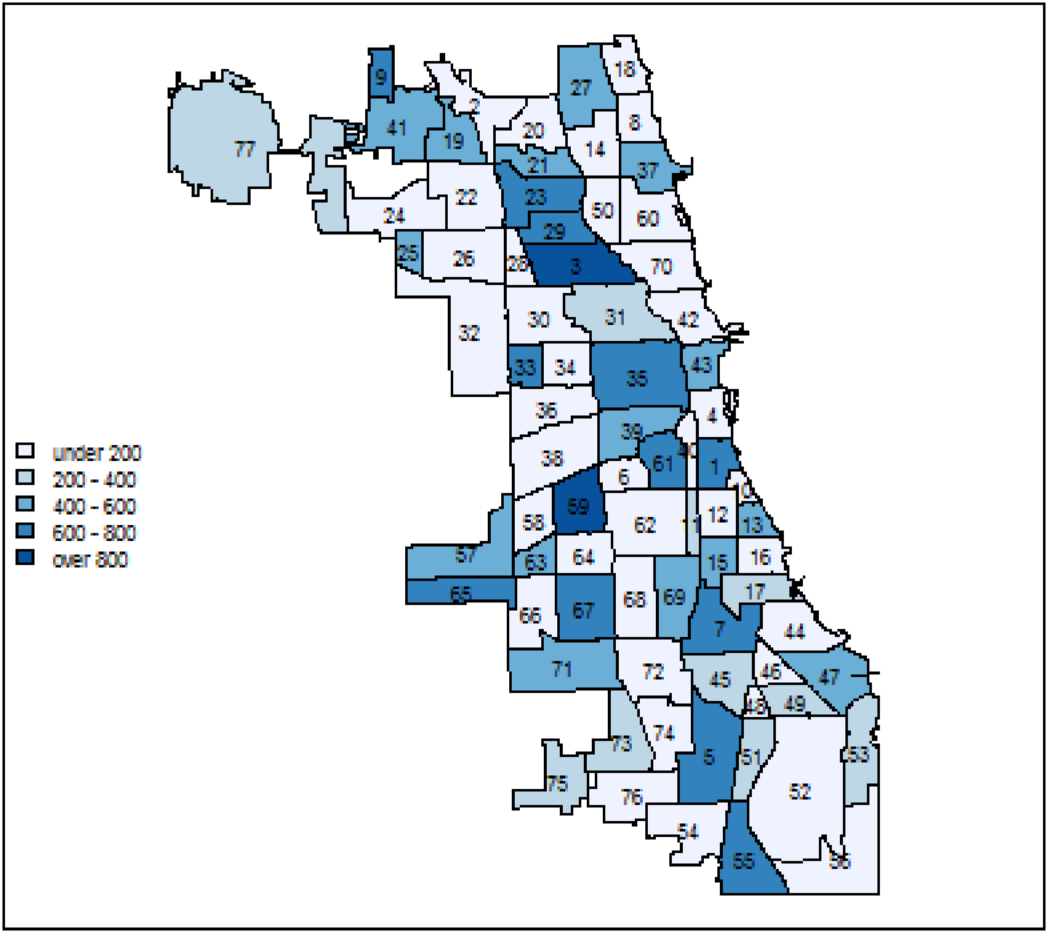
Best Allocation Found After 20 Seconds and 180,000 Steps



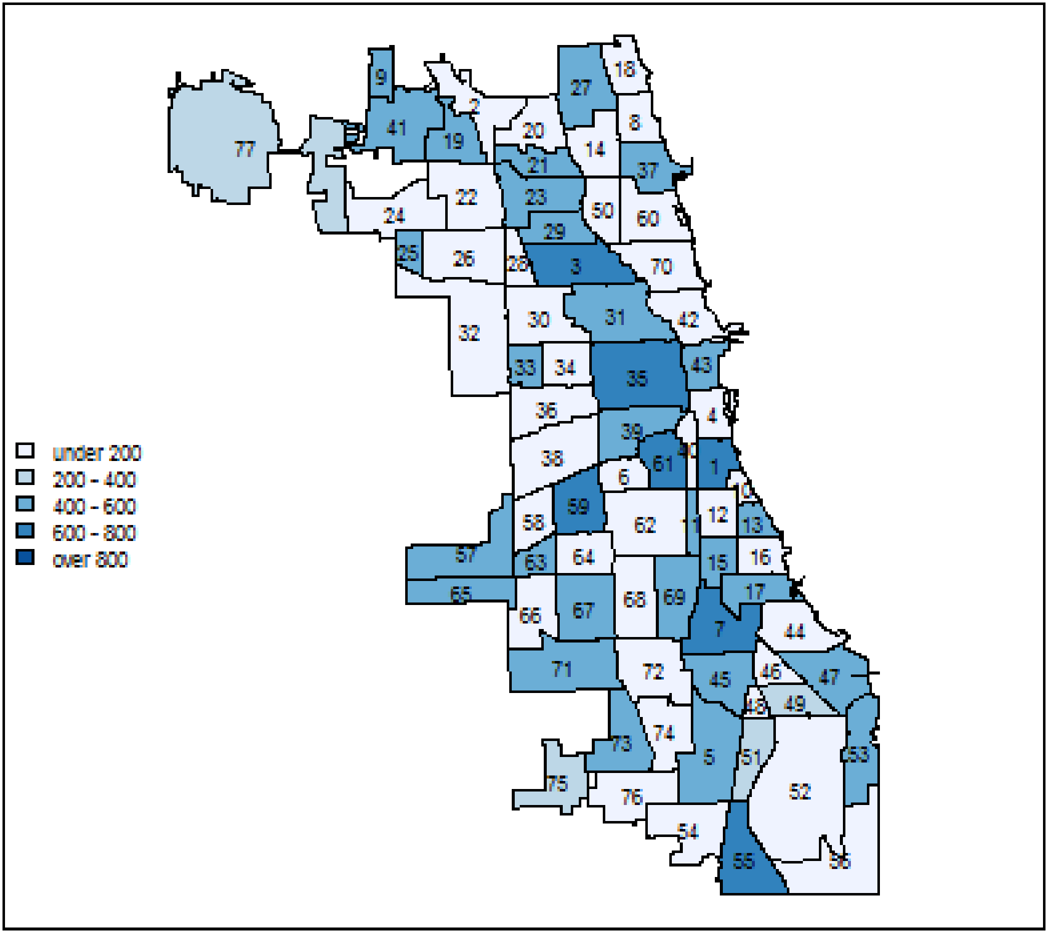
Best Allocation Found After 23 Seconds and 210,000 Steps



Best Allocation Found After 27 Seconds and 240,000 Steps



Best Allocation Found After 30 Seconds and 270,000 Steps



Best Allocation Found After 33 Seconds and 300,000 Steps