The Public Mapping Project
Altman, Micah, McDonald, Michael P.

Published by Cornell University Press

Altman, Micah and Michael P. McDonald.
The Public Mapping Project: How Public Participation Can Revolutionize Redistricting.

格力 For additional information about this book
https://muse.jhu.edu/book/62760

格力 For content related to this chapter
https://muse.jhu.edu/related_content?type=book&id=2793431
The DistrictBuilder application is a cornerstone of the Public Mapping Project, intended to foster greater public participation and transparency in redistricting. The software we developed to achieve this goal is emblematic of current redistricting applications, so it is instructive to review what DistrictBuilder does to understand how mapping applications assist humans in drawing districts and the role they play in public mapping. Our guiding principle was to create an easy-to-use internet redistricting application that allows people to quickly get up to speed and start drawing districts, rather than spend frustrating hours installing software, configuring it, and learning how to use it. High school students, retirees, and many people in between have used our software to create perfectly legal
districts, something naysayers said was impossible when we embarked upon the project.

Our aspiration to create DistrictBuilder began during a 2007 American Mathematical Society meeting in Washington, DC, where we delivered a presentation of our research on automated redistricting. We concluded automation was not a viable solution to gerrymandering, as explained more fully below. After our presentation, we were approached by Daniel Goroff, professor emeritus of mathematics and economics at Claremont’s Harvey Mudd College and a vice president of the Sloan Foundation, a charitable organization. He challenged us: What would you do? We responded that instead of relying on machines, we would empower humans. After all, the original meaning of the word *computer* is “one who computes” as it was applied to human computers who did tedious, repetitive calculations before the advent of machine computers. There is a good reason why we sought to engage humans: our complex brains have ways of seeing solutions to problems that might elude a machine. This is particularly true with redistricting for two reasons. First, humans often perform better than computers in processing visual information like geographic units awaiting assignment to districts. For example, a computer has a difficult time seeing how to tie geographically separated communities together, whereas humans can quickly see the problem and form a solution. Second, redistricting plans are proposals for political representation—although made in a technical form. Since there is
no universally agreed-upon measure of representation, it is important for the public to be able to express proposals that reflect their conception of what representation means. As we told Daniel Goroff, we wanted to involve more people in the redistricting process by providing them with mapping tools and data, delivered through the internet. We believed that having more eyes on the problem would help expose policymakers, courts, media, and the general public to a wider range of possibilities beyond the gerrymandered districts offered by political parties.

Daniel was intrigued by our answer and assisted our vision by providing support from the Sloan Foundation.33 To create DistrictBuilder, we engaged Azavea Corporation, a Philadelphia company that applies geospatial technology for positive civic, social, and environmental impact while also conducting research.34

**Accessing DistrictBuilder**

DistrictBuilder is among the new generation of redistricting applications that are web-based, so people can immediately begin mapping without having to worry about software installation and data management. What distinguishes DistrictBuilder from other redistricting applications is that it is open-source software, which means anyone can obtain the software at no cost; in our case, from a popular internet archive known as GitHub.35 It is distributed
under an open-source license, meaning that it is available to everyone for inspection and reuse.

Indeed, Bill Morris—a Burlington, Vermont, city administrator—used DistrictBuilder for his city’s redistricting in 2011. We only learned about his work when he contacted us with questions. He even did some programming to meet his specific needs, later adding his programming to the DistrictBuilder code base. This illustrates another benefit of open-source software: it is in the public domain, so the public can take ownership of it to make it better. In contrast, the public cannot inspect closed-source, proprietary systems; these programs can contain errors, usually a result of unintentional software bugs, but we cannot rule out the possibility of intentional manipulation hidden from the public. For this reason, we believe strongly that election systems—be they voting systems or redistricting systems—belong to the people and should reside in the public domain.

What follows next is an orientation of DistrictBuilder’s basic features. We will likely continue to modify the software to improve users’ experiences, but these core functions will remain. The description offers glimpses into redistricting complexities, how we overcome them, and how we executed the vision of crowd-sourced mapping for domestic and international use. Persons wishing to draw redistricting plans or orchestrate public mapping advocacy will find this section instructive, but if the following orientation demystifies the process in somewhat too much detail for the casual reader, feel free to skip ahead to the next section, “Why Not Automated Redistricting?”
Using DistrictBuilder

An intrepid mapper needs advanced computer skills to install DistrictBuilder directly onto his or her own computer from its source code. Therefore, our user-friendly deployment model is for a knowledgeable administrator to configure the software for others to access via web browser.

Assuming the presence of an administrator familiar with cloud computing infrastructure, we made the installation process easier by placing prebuilt instances on cloud computing sites like Amazon, which has a gentler learning curve than a bare-metal installation from source code. While the software is free, a person or organization using DistrictBuilder on the computing cloud will incur charges for server time. Monthly charges run only a few dollars for personal use. Another advantage of a cloud server is that in the case of a catastrophic success—for instance, a popular rush to map new districts—increasing server capacity to accommodate demand is easy.

Here we should mention that hosting DistrictBuilder for many simultaneous users requires a hefty server that will incur higher monthly costs, perhaps into the thousands of dollars. An organization hosting a redistricting competition or general public mapping may wish to install the software on their own servers to better manage the computing load, as there are backend tricks that can optimize performance.
Login Experience and Creating Accounts

By default, DistrictBuilder allows users to create a new account from the login page. Administrators can opt out of this setting, however, and distribute login credentials themselves. This feature accommodates organizations that wish to set up internal mapping applications.

While restricting access may seem the antithesis of our public mapping philosophy, anyone can tweak the software code to do this anyway, and there are certain use cases we wish to support, such as enabling internal sharing between state and local governments during redistricting to improve the integrity of election data. For instance, a state government might wish to draw new districts or precinct boundaries, and in this case, an inefficient option would be to purchase a separate commercial software license for each state and local redistricting effort. With DistrictBuilder, however, the state can distribute credentials and work in tandem with their localities in a shared, private setting, using the same mapping system and data.

Note that a large number of users can overwhelm a server. Therefore, DistrictBuilder puts a throttle on the number of accounts that can run simultaneously. When the limit set by an administrator is reached, users will receive a message to try back later.
Basic Features

A brief tour of DistrictBuilder provides a feel for what the current generation of redistricting software looks like and does. A user's typical mapping session begins by logging into their account, selecting a new or existing plan to work on, and then mapping. With a click of a button, the software can be displayed in English, Spanish, French, or Japanese. (We chose to implement Japanese during software development because the language reads right to left.) The internationalization features work; we have had Spanish-speaking university students in Mexico successfully use the software to draw districts.

Once logged in, users are presented with a file directory containing existing maps to edit or the option to create a new map. The administrator may provide suggested starting maps. Users can select any other map that other people have shared, which will copy the map to their account for editing. Users can also copy their plans for exploratory mapping.

To become familiar with the mapping interface, it will be helpful to walk through a real-world example. In figure 4.1, the background base map looks somewhat like what one might see in a common phone map app. Cities, roads, water, and other landmarks assist in identifying the location where users are drawing districts. We use open-source base maps in this example, but there are options to use proprietary base maps, too, even including satellite imagery. In this case, these
Figure 4.1. DistrictBuilder Philadelphia deployment
visual clues help us identify that we are drawing council districts for the City of Philadelphia.

Atop the Philadelphia map we overlay information needed for redistricting. The variously shaded, grayscale blocks represent Pennsylvania’s wards, with lighter shading signifying a ward with lower population and darker colors a higher population. The shading scheme also tells us about each district’s total population (we will explain the colors in a moment). It is possible to change what the shading signifies, however, to instead provide information about an area’s racial, ethnic, or partisan composition.

Not apparent in the static figure 4.1 is a tool that allows users to zoom in and out. Users don’t need to select different geographies to work with; simply zooming in and out presents logical choices. When a user zooms in, smaller pieces of geography become visible, allowing that user to work with voting precincts or even individual census blocks. Besides simplifying the user experience in an intuitive manner, this feature is incredibly important from a technical standpoint. A typical state has over a hundred thousand census blocks; displaying all of them while viewing a statewide map would place terrible strain on the server, require pushing a lot of data through the internet, and diminish the software’s performance. Trimming the data to match the zoom level helps manage this performance challenge.
Meeting Legal Requirements

Redistricting plans must comply with federal and state requirements. Online resources like the Brennan Center’s *A Citizen’s Guide to Redistricting* and the National Conference of State Legislatures’ *Redistricting Law* series are good primers on the legal intricacies. In a nutshell, federal law requires that districts have equal population, and that the task of drawing districts should ensure minorities an opportunity to elect a candidate of their choice in compliance with the Voting Rights Act and the 14th Amendment. Requirements vary by state, and sometimes even by congressional and state legislative district within the same state. Common requirements include contiguity; compactness; respecting existing political boundaries such as counties and townships; respecting communities of interest; respecting geographic features; continuity of the territory in the old and new districts, to the extent that it is possible; and even compatibility with political goals such as political fairness and competitive districts.

DistrictBuilder has features that assist map drawers to achieve legal goals like these. In our example, an intrepid mapper’s Philadelphia’s city council districts overlay the color-coded wards. These districts are represented by gold boundaries and gold-, blue-, and clear-filled areas on our map, helping users achieve the first important federal requirement: equal population districts. The ideal target population for a
district is calculated simply by dividing the population of a state or locality by the number of its districts. Blue-colored districts fall short of the ideal target population for population equality, gold-colored districts go over the target, and clear districts meet it.

The side panel on the right can display any measurable statistic about districts. Some merely add up numbers, such as districts’ total or voting-age population broken down by race, ethnicity, and various election results to measure partisan leaning. Some statistics relate to districts’ geography, such as whether a district is contiguous (all parts connect), district compactness (measured in different ways), how many local political boundaries are split by districts, and even the travel time across a district, which is a consideration for Mexico’s federal districts.

In figure 4.1, the side panel displays districts’ total population, with colored context clues for over- and under-populated districts; a checkmark indicates whether a district is contiguous (green) or not (red), and one of many available compactness measures. A pull-down menu allows users to display other preset statistics configured by an administrator, as well as a user-defined custom set of statistics.

**Drawing Maps**

To do mapping, a user first selects geography to assign to a district. The visual cues like color-coding and shading
will suggest good pieces of geography to add to a district. A person can point and click to select a single piece of geography or use a lasso tool to select several. Once the geography is chosen, it can be grabbed, dragged, and dropped into the desired district, which assigns it to that district. This workflow functions best when editing adjoining districts to balance population. Users can also choose to immediately assign any selected geography to a specific district without dragging and dropping, which is useful when building a new district. Of course, there is an “undo” button to walk back mistakes.

An important innovation of DistrictBuilder is it enables crowd-sourced redistricting. Individuals and organizations no longer need to work independently and in isolation on their own machines. A central web server can simultaneously support several users. People can work on their districts in private, and when they are ready, share their maps publicly with others. They can copy and paste districts from any shared map into their working plan. In this way, people can work together to improve upon others’ ideas.

**Sharing Maps**

Most important, users can share their work. Features allow the import and export of redistricting plans in commonly used data formats. Importing is useful when
a government or organization releases a plan for public consumption, and exporting is useful to submit plans to governments.

Web links are generated for specific maps, which can be incorporated into news stories or social media. These links are accessed through an anonymous login that allows people to see but not edit the map. This serves two important purposes. First, people do not need to create an account to view a map. Second, stripping down the mapping interface lessens the server load, which improves users’ experience.

Once created, all districts and entire plans can be scored by how well they meet various criteria. DistrictBuilder can rank shared plans on leaderboards, which list the plans that have the highest score on criteria such as district compactness, splits of local community boundaries, political fairness, and many other district and plan metrics. These individual scores can be combined into an overall score that is a weighted composite of different measures—something Mexico does formally as part of their redistricting process, and what Ohio reformers did as part of a recent redistricting competition.

Data

Redistricting software cannot work without data. In the information age it is easy to assume that data are readily available, but this has not been entirely true for redistricting.
The Census Bureau publicly releases population and geospatial data, the most basic necessities for drawing districts. Most state governments have not made the election data necessary, even if politicians use these data to assess the consequences of moving district boundaries to add or subtract a partisan-leaning community from their district. Keeping such information private prevents the public from doing the same evaluation, so the public cannot know to what degree a proposed redistricting plan is a gerrymander.

Augmenting census data with election data is costly, as it requires collecting both election boundaries and election results and marrying these data to census data. A few states, such as California, Hawaii, Louisiana, Michigan, Minnesota, North Carolina, Oklahoma, Texas, Utah, and Wisconsin, maintain websites where they provide election boundaries and election results for every election on a statewide basis. Sometimes states will release redistricting databases of merged election and census data in the midst of redistricting, such as in Ohio’s case. In other states, these data must be collected from localities, which can be a time-consuming process; many localities do not post their data online in accessible formats, some local election officials do not respond, or they charge for their data.

Creating redistricting databases is also possible using an alternative source of data, if a given state does not release election boundary data. In advance of the decennial census, in a year ending in a 7 or 8, the Census Bureau requests
political boundary lines from states and localities in what is known as Phase 2 of the Redistricting Data Program. These boundaries include what the Census Bureau calls Voting Tabulation Districts, or VTDs, which is their generic name for precinct, ward, and election district boundaries. Nearly all states participate fully in Phase 2, although in the past a few did not participate at all or provided only partial data. The availability of these data make it easier to augment census data with election data circa a year ending in 8. A catch is that some localities—primarily larger ones—continuously change election boundary lines so there is no absolute guarantee that the Census Bureau’s VTD boundaries correspond with the actual boundaries used in an election. Accounting for these changes can still be tedious work.

Without diving too far into the weeds, this data work is fraught with further complexities. Briefly, one issue is how to account for election results that some states and localities do not report by precinct, such as early voting results reported only for entire counties. Another is how to merge together census and precinct data when the boundaries do not perfectly coincide with one another. We describe methodologies on how to resolve these issues elsewhere. We, and other organizations, are working to collect and disseminate publicly these election data in advance of the 2020 round of redistricting.

Some state governments do this data work themselves and release a merged census and election database during redistricting. In addition to the states that continually release
election boundary and results data, Arizona, Colorado, Florida, and Ohio released these databases in prior rounds of redistricting. If a state releases such a database, we recommend using it rather than duplicating a significant amount of work. The adage of “trust, but verify” applies. In the midst of the Ohio redistricting competition, the DistrictBuilder software experienced what looked to be a software bug. After frantic days of troubleshooting, Azavea discovered that the Ohio database had geographic errors that caused the software to seize up. After we notified the state, they issued a new database that still had errors. The third time was the charm, but we were delayed for weeks—at considerable expense—by an error that people wrongly associated with DistrictBuilder when the state’s data were to blame.

Why Not Automated Redistricting?

Our DistrictBuilder software description raises a natural question: Why not just let computers do redistricting all and take the grubby self-dealing humans out of the equation? You’re not alone in asking this question. As then-governor Ronald Reagan stated, “There is only one way to do reapportionment—feed into the computer all the factors except political registration.” Many others have subsequently supported this viewpoint. So, why not just let machines do the job?
The short answer why we shouldn’t just hand over redistricting to computers is that humans program computers. A programmer is faced with many choices when creating an algorithm that automatically draws districts. These choices might unintentionally or intentionally bias a computer to select a particular type of redistricting plan, thereby substituting human gerrymandering for machine gerrymandering. Furthermore, these programming choices often embed interpretations of representational values, such as fairness and protection of communities of interest, that should emerge as the result of the redistricting process—not be defined by it.

The long answer is that programming computers to do redistricting is surprisingly really hard. James Weaver and Sidney Hess, the Delaware advocates who created the first automated redistricting application in the early 1960s, understood this limitation because they approached the problem from an operations sciences background. In this field, businesses have high demand for mathematical solutions to similar problems that cost companies substantial amounts of money. They did not completely dismiss a role for computers; after all, they wrote the first automated redistricting algorithm. Stuart Nagel, who developed an algorithm soon after, observed that an automated redistricting algorithm is useful in “testing some policy proposals.” By virtue of being able to generate a large number of plans quickly, computers help inform us about the available choices. Curiously, when it comes to redistricting, the wheel is constantly being
reinvented by contemporary scholars who rediscover some of the lessons of these early efforts.\textsuperscript{41}

Redistricting is a mathematically hard problem because there is a staggeringly large number of redistricting plans. A modest-sized state may have a hundred thousand census blocks, which might be partitioned into five or six congressional districts or fifty or more state legislative districts. To get an idea of why there are so many redistricting plans, imagine flipping a coin and noting whether it comes up heads or tails. Now imagine flipping the coin one hundred thousand times, with each sequence of heads and tails representing a unique redistricting plan. The number of possible combinations of heads and tails boggles the mind. There are more possible redistricting plans than quarks in the universe, and if every computer on Earth were set to the task of redistricting, the sun would engulf the Earth before the computers could draw all the feasible plans.

A proponent of automated redistricting might counter that a computer need not find all the redistricting plans. To work well it needs only to find the best redistricting plan, or else randomly sample redistricting plans. A problem with this approach is that a computer cannot be simply programmed to choose the best plan or an unbiased random selection of plans. Thinking back to our coin-flipping exercise, it would be tempting to think that a computer algorithm could just virtually flip a coin a hundred thousand times to produce a legal redistricting plan. Computers complete
repetitive tasks very quickly. It turns out this approach is terribly inefficient in finding legal redistricting plans. Our sequence of coin flips would technically be a redistricting plan, but it might not be a legal redistricting plan because it gives too much or too little population to a district, creates noncontiguous districts, violates the Voting Rights Act or other state constitutional or statutory requirements, and so on. The problem is that we don’t know which sequence of coin flips will result in a legal plan until we start flipping the coin. Like the proverbial monkeys banging on typewriters in search of Shakespeare’s complete works, an incredibly large number of unsuccessful coin flips is needed to draw a single legal redistricting plan.

Proponents of automated redistricting get around this complexity problem in two ways. Their first approach simplifies the problem. Instead of drawing districts out of census blocks, they draw them out of larger voting precincts. This reduces the number of geographic units to assign from the hundreds of thousands to a few thousand. Simplification makes the problem more computationally manageable, but this simplification produces districts that are not legal, most obviously because combinations of voting precincts rarely produce districts with equal population.

Even by reducing the number of units to assign to districts, redistricting is still a staggeringly difficult problem such that a computer still cannot efficiently search for the best plan or randomly sample with as few as forty geographic
units to assign to districts. A second simplification uses a heuristic to draw redistricting plans. Heuristics are rules to solve problems, such as when in a maze, always turning right when hitting a wall before proceeding forward again will eventually get you to the exit. The problem with applying heuristics to very complex problems is they are not guaranteed to find good solutions. Trapped in a really large maze, one might die before exiting if constrained to only making right turns. Automated redistricting algorithms typically use variants of the following heuristic, first implemented in the 1960s:

Step 1. Select a random census block (or precinct).

Step 2. Randomly assign adjacent census blocks until a legal district is created. Repeat Step 1 and Step 2 as needed to create an entire redistricting plan.

Step 3. Make random trades of census blocks between adjacent districts until a legal (or optimal plan) is created.

Surprisingly, doing random things does not guarantee random results. A frequent heuristic is the random assignment of adjacent census blocks to build districts (Step 2). This tends to, but does not always, favor the creation of districts that are centrally clustered over those that are dispersed. Some may view this as a feature, not a bug, but it means that this heuristic will not actually search the space of all legal redistricting plans,
potentially missing some that perform better on the goals one may care about. For example, such automated algorithms have trouble creating voting rights districts, which sometimes require tying together nonadjacent communities of color.

In a third step, an automated algorithm trades geography in pursuit of creating an optimal plan. There are subtle but important complications to this approach. Computers can only be programmed with measurable goals. A particularly problematic goal is respecting communities of interest, which twenty-four states require for state legislative redistricting and thirteen for congressional redistricting.42

There is no agreed-upon definition for communities of interest, so this criterion typically devolves into vague impressions of what constitutes a community by those conducting redistricting. Even when a goal can be measured, there must be agreement on how to measure it. One group of scholars finds there are over one hundred different ways to measure district compactness.43 These various measures might examine a district’s perimeter, area, convexity, and the location of its population, or take into account geographic barriers like water. Suppose that a decision can’t be made on a single measure, and instead, more than one will be used. How does one combine them into a single overall measure? A simple average might make sense, but the measures might not be calculated on the same scale. This is certainly a problem if we incorporate goals other than compactness into a redistricting authority’s decision-making
process, such as population equality, the number of times districts split local political boundaries, and so on.

Moreover, selecting the “optimal” redistricting plan inevitably requires making decisions about how different representational values should be balanced. This is true in practice, and even in theory. That is, it is impossible to simultaneously maximize competitiveness, partisan unbiasedness, communities of interest, and other desirable criteria—even when we agree on how to quantitatively measure these. Purely automated systems preempt human judgment about how to balance legitimate goals.

Perhaps one day someone will create, in the limited time available during a redistricting period, an automated redistricting algorithm that works well, draws legal districts respecting all required criteria, and can draw districts in an unbiased manner that reflects with fidelity whatever representational values the public wishes to consider. That day is not today, and given what we know about the mathematical complexity of the problems and limits of computers, such a program may not be designed in our lifetimes. Suppose it was, however—what then? Automated redistricting algorithms create samples of redistricting plans that vary across the goal one cares about. An important public policy matter like redistricting should not be a crapshoot. Human intervention is needed to make a choice among all proposed maps, be they by machine or human.
Man versus Machine: Why Not Both?

Our critique of automated redistricting should not be interpreted as a wholesale rejection of the approach. We believe automated algorithms can serve an important function to quickly develop policy alternatives for consideration, along with maps created by humans.

Mexico’s use of automated redistricting illustrates both the promise and potential pitfalls in automation. Mexico’s national election management board has required automation as a step in past redistricting processes for the country’s single-member, lower-chamber districts. Alejandro Trelles, a public mapping collaborator at Brandeis University, was formerly a staff member of Mexico’s redistricting commission, what is now known as Instituto Nacional Electoral, or INE (formerly known as Instituto Federal Electoral, or IFE). Alejandro connected us to his former colleagues who have considered using DistrictBuilder as a part of the INE’s official public outreach. Our communications led to access to the internal INE plans and data produced during a prior redistricting.

Mexico’s redistricting process starts with an INE committee defining a set of measurable criteria to apply to districts. These criteria include population equality, compactness, respecting municipal boundaries, and minimizing travel distance across districts. (DistrictBuilder can produce statistics for all these criteria.) The committee then assigns weights
to these components to develop an overall score for each redistricting plan. INE then employs an optimization algorithm to draw federal districts for each of Mexico’s thirty-two states. After the computer produces districts, the committee members, who are delegates of the political parties, can offer alternative plans. In the majority of states, committee members’ counterproposals scored higher than those produced by a computer. Indeed, in 2013, humans offered counterproposals that scored much better than the computer-generated plans in the states of Hidalgo, Puebla, and San Luis Potosí.

Automation plays an important role in Mexico’s process insofar as it constrains the choices available to the political parties. They have to play a game of beat-the-machine by producing plans that score better than the baseline generated by the automated algorithm. This means a proposed plan must fare better on some combination of population equality, compactness, respecting municipal boundaries, and minimizing travel distance across districts. If a political party can do so, then their counterproposal may be considered for adoption by the redistricting committee. A drawback of this approach is the proverbial dog that did not bark. There is no incentive for a political party to offer a plan that makes them worse off. This may be mitigated by the involvement of other political parties, but not all parties have the same capacity to draw maps. We noticed certain larger parties making counterproposals more frequently.

Another innovative facet of Mexico’s redistricting is computer-assisted design. The political parties’ counter
proposals accepted by the INE committee are fed into the computer and the optimization algorithm is run again to see if the computer can devise yet a better solution. In some states the computer algorithm did so, and humans were yet again able to beat the computer in some of their final counterproposals. We built similar rudimentary computer-assisted design features into DistrictBuilder. A tool identifies all the unassigned geography and helps users find these slivers of land, such as a stray census block located in a stream or road median. This can be a tedious-but-important task, because a plan that does not assign all geography to a district is not a legal plan. A completion tool automatically assigns these orphaned pieces to the nearest district.

We believe this automation approach can be improved upon using a process similar to Mexico and creating more sophisticated optimization algorithms. In this way, humans and computers can together search for the best possible plans. However, given the complexity of the redistricting problem, and given the current limits of computers, we can never know for sure whether we have found the absolute best plan. Still, slight deviations from the ideal is far better than what many states have today.