Computers & Operations Research 160 (2023) 106369

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Redistricting optimization with recombination: A local search case study

Kiera W. Dobbs [a,*], Douglas M. King [a], Sheldon H. Jacobson [b]

[a] Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, Urbana, 61801, IL, United States
[b] Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, 61801, IL, United States

ARTICLE INFO

Keywords:
Political redistricting
Political fairness
Local search
Recombination

ABSTRACT

In the U.S., states redraw electoral district boundaries every ten years. Given that redistricting affects political representation at both the state and national levels, it is crucial to prevent the manipulation of district boundaries for political gain. Optimization methods can be valuable tools for promoting transparency and fairness in redistricting. Here we examine a novel local search approach for redistricting that transitions between feasible solutions using Recombination (a recently introduced spanning tree iteration). We compare the performance of multiple local search heuristics using both Recombination and more traditional Flip iterations by optimizing congressional plans for Illinois, Missouri, and Tennessee with respect to several common fairness objectives. We evaluate which heuristic produces the best objective value within limited time periods and generate collections of optimized plans. The Recombination heuristics produced excellent objective values, often far superior to the Flip heuristics; they also maintained more compact district shapes when the objective was not compactness. However, the Flip heuristics converged to a local optimum more quickly and occasionally achieved better solutions than the ReCom heuristics within short time periods. Hence, while the use of Recombination within local search frequently improves solution quality, there are some scenarios for which Flip may be preferable.

1. Introduction

Political redistricting in the U.S. can be quite a contentious process, with multiple stakeholders vying for competing interests. Often there is concern about *gerrymandering*, the manipulation of district boundaries for political gain (Ricca et al., 2013; Arizona State Legislature v. Arizona Independent Redistricting Commission, 2015). States in the U.S. redraw their congressional and state legislative district boundaries every ten years, following a decennial census (National Conference of State Legislatures, 2021c); hence, redistricting (and consequently, gerrymandering) can affect political representation for an entire decade.

Political redistricting can be viewed as a graph partitioning problem, since it involves dividing a geographic region into nonempty, pairwise disjoint districts. The region to be divided typically consists of finitely many geographic units (e.g., census blocks), so it is natural to represent the region with a planar graph. The vertex set of this planar graph represents the set of geographic units and there exists an edge between two vertices if and only if the corresponding two units are adjacent; a district plan is a partition of this graph where the parts represent the districts. Typically, the goal of graph partitioning problems is to construct parts that optimize some objective function while adhering to a set of constraints. One example is minimizing the *cut* of a partition (i.e., the sum of edge weights between parts) while satisfying a *balance* constraint (i.e., all parts have roughly equal weights) (Bichot and Siarry, 2011). Similarly, a goal of redistricting may be to minimize district perimeters (to promote district compactness) while maintaining roughly equal district populations. Other potential redistricting objectives may quantify district shapes or political characteristics (e.g., proportionality, competitiveness).

There are several requirements established to standardize political redistricting and support equity in the redistricting process. Federal law requires districts to be equi-populous (Wesberry v. Sanders, 1964; Reynolds v. Sims, 1964) and comply with the Voting Rights Act of 1965; many states impose additional requirements, such as contiguity or compactness (National Conference of State Legislatures, 2021b). To minimize potential manipulation by any particular political party and/or encourage bipartisan cooperation, fifteen states require an independent or bipartisan commission to create their congressional and/or state legislative district plans (National Conference of State Legislatures, 2021a). Several states have also adopted requirements related

* Corresponding author.
E-mail addresses: kdobbs2@illinois.edu (K.W. Dobbs), dmking@illinois.edu (D.M. King), shj@illinois.edu (S.H. Jacobson).

https://doi.org/10.1016/j.cor.2023.106369
Received 4 December 2022; Received in revised form 31 July 2023; Accepted 1 August 2023
Available online 5 August 2023

to a district plan's political characteristics. For example, Missouri and Ohio require the expected fraction of state legislative seats won by each party to be proportional to their statewide fraction of votes (Mo. Const. art. III, §3, 2020; Ohio. Const. art. XI, §6, 2015); Arizona, Colorado, New York, and Washington require district elections to be competitive when possible (Ariz. Const. art. IV, pt. 2, §3, 2000; Colo. Const. art. V, §44, 2018; N.Y. Const. art. III, §4, 2014; Wash. Const. art. II, §43, 1983).

Optimization algorithms can also be useful for supporting fairness in redistricting. They encourage transparency in the redistricting process, since they create district plans based solely on clearly defined objectives, constraints, and parameters. Creating a district plan by transparent means can give insight into the interactions between constitutional requirements, political geography, and political fairness (Dobbs et al., 2023). However, solving a redistricting optimization problem exactly is often intractable for realistic instances. For example, balance-constrained graph partitioning is NP-hard (Bichot and Siarry, 2011); district plans must be similarly population-balanced and typically involve partitioning thousands of geographic units. District plans typically must also satisfy contiguity (i.e., the induced subgraph on each part must be connected) which imposes further computational burden. Solving a redistricting optimization problem becomes increasingly difficult with the incorporation of additional state requirements (Altman and McDonald, 2010; DeFord et al., 2021). Hence, redistricting presents a typically intractable problem.

Moreover, even if redistricting problems can be solved tractably, it is not always clear which objective to optimize, how to balance multiple competing objectives, or how to enforce vague or nuanced requirements (e.g., preserving communities of interest) (Altman and McDonald, 2010). Nevertheless, with increases in computational power and the availability of redistricting data, optimization can be a valuable tool in the redistricting process. For example, Bozkaya et al. (2011) applied optimization heuristics to create electoral districts for the city of Edmonton, Canada that were signed into law and used in the 2010 municipal elections.

Even if algorithmically generated plans are not ultimately enacted as-is, they can demonstrate whether certain redistricting goals are achievable for a particular state's constitutional requirements and political geography, and promote transparency and public discussion around the redistricting process (Altman and McDonald, 2010; Validi et al., 2021). Optimization algorithms can also provide map-makers with a collection of plans that satisfy basic requirements and possess good political fairness scores, which map-makers can then choose to implement directly or adjust based on more nuanced requirements. Therefore, a realistic goal for redistricting optimization is not to construct the singular "best" district plan, but rather create flexible algorithmic tools that can construct a collection of quantifiably good plans and examine the effects of prioritizing different redistricting preferences or requirements.

Local search, a common optimization meta-heuristic, can be one such flexible algorithmic tool for redistricting; in contrast to exact algorithms for redistricting, local search has the potential to quickly improve a given objective while enforcing multiple constraints. This paper evaluates local search algorithms for redistricting following the introduction of a new iteration type. Previous local search algorithms for redistricting use a *Flip* iteration to transition between district plans (e.g., Ricca and Simeone (2008), Bozkaya et al. (2011) and King et al. (2018)); each iteration reassigns a single geographic unit on the border of adjacent districts. However, if the districts are constructed with larger units (e.g., census tracts), it may be difficult to move a single unit from one district to another without violating common redistricting constraints such as population balance; if the districts are constructed with smaller units (e.g., census blocks), flipping a single unit is unlikely to substantially improve the objective. Hence, local search with Flip iterations can be ineffective at exploring the solution space, and therefore often converges to a low quality local optimum.

To allow local search to make substantial changes to a district plan while maintaining redistricting constraints, we consider *Recombination* (ReCom) iterations as an alternative to Flip iterations. ReCom is a spanning tree method originally introduced for district plan ensemble-generation (DeFord et al., 2021). While Flip moves one unit per iteration, ReCom frequently exchanges multiple units between two adjacent districts simultaneously. Exchanging multiple units simultaneously at each iteration allows ReCom to make substantial changes to district boundaries while maintaining redistricting constraints such as population balance. Local search with ReCom iterations can therefore effectively reach a variety of district plans, and consequently can achieve better objective values than with Flip iterations.

This paper empirically analyzes the performance of simple hill-climbing, simulated annealing, and greedy local search optimization algorithms that use ReCom or Flip iterations on the Illinois, Missouri, and Tennessee congressional redistricting instances. Using these algorithms, we heuristically optimize Illinois, Missouri, and Tennessee congressional district plans with respect to four common redistricting metrics (compactness, efficiency gap, mean-median, and competitiveness). To assess the potential trade-off between run time and solution quality, we compare the best objective value that each algorithm achieves within limited time periods. Then, to evaluate consistency and robustness, we use each algorithm to generate a collection of plans for each objective and compare average objective values. In these computational experiments, the ReCom algorithms tended to improve the objectives more substantially and consistently than the Flip algorithms; since ReCom iterations can make more substantial changes to district boundaries than Flip iterations, the ReCom algorithms were able to more radically reshape the districts or considerably alter their political composition. Additionally, the ReCom algorithms better maintained compact districts when the objective was not compactness. However, it took longer on average for the ReCom algorithms to converge to a local optimum; in particular, simulated annealing with Flip iterations occasionally yielded better objective values than the ReCom algorithms when time was limited.

This paper is organized as follows. Section 2 reviews computational methods for redistricting. Section 3 describes the redistricting problem, including fairness objectives, legal constraints, and relevant data for congressional redistricting in Illinois, Missouri, and Tennessee. Section 4 outlines the optimization methods used to construct the plans that optimize the fairness objectives from Section 3. Lastly, Section 5 analyzes the district plans optimized for each objective and Section 6 provides closing remarks.

2. Related work

There is an extensive body of literature exploring optimization algorithms and other computational methods for redistricting. Ricca et al. (2013) and Ricca and Scozzari (2020) review previous work in this area and Ríos-Mercado (2020) provides a more general review of recent techniques for districting, zoning, and territory design problems. Early approaches for redistricting optimization use exact methods, such as enumeration and integer programming (e.g., Garfinkel and Nemhauser (1970) and Hess et al. (1965)). For example, Hess et al. (1965) formulate the basic redistricting problem as a facility-location integer program with a moment of inertia compactness objective. This formulation does not include contiguity constraints, so any solutions obtained may need to be adjusted or discarded. Since this formulation was introduced, several authors have added contiguity constraints (e.g., Shirabe (2009), Oehrlein and Haunert (2017) and Validi et al. (2021)); similarly, Arredondo et al. (2021) extend the basic model to guarantee minority representation. However, the basic redistricting problem can quickly become intractable for a realistic instance with thousands of geographic units (even with the helpful MIP techniques presented in Validi and Buchanan (2022)); hence, realistic problem instances continue to require heuristics.

Not only is redistricting typically intractable, it is also nuanced; it is impractical to precisely model all stakeholder considerations and preferences. Previous work tends to focus on optimizing district shapes (i.e., *compactness*); this focus is natural, since several states require compact districts (National Conference of State Legislatures, 2021b) and a compactness objective lends itself nicely to clustering algorithms and facility location models (e.g., Hess et al. (1965), Validi et al. (2021) and Swamy et al. (2022)). However, since several states have adopted redistricting requirements related to political fairness (National Conference of State Legislatures, 2021b), there is cause to incorporate political fairness objectives into optimization methods for redistricting. For example, King et al. (2018) use a competitiveness objective within local search and Swamy et al. (2022) model several popular political fairness metrics within a facility location mixed-integer optimization framework. This study incorporates various political fairness metrics (efficiency gap, mean-median, and competitiveness) within a novel local search optimization framework.

Local search heuristics using Flip iterations have previously been applied to redistricting problems (e.g., Ricca and Simeone (2008), Bozkaya et al. (2011) and King et al. (2018)). Ricca and Simeone (2008) analyze and compare the performance of Flip-iteration local search algorithms, such as tabu search, simulated annealing, and old bachelor acceptance, and conclude that these algorithms can quickly identify district plans with better objective values than currently enacted plans. However, the only constraint enforced at each iteration is contiguity; other common redistricting criteria such as population balance, compactness, or conformity to administrative boundaries act as the objective functions. Choosing instead to enforce these common redistricting constraints at each iteration would likely prevent a Flip-iteration local search algorithm from reaching a variety of feasible district plans (Chikina et al., 2017; Fifield et al., 2020; DeFord et al., 2021; Cho and Liu, 2021). For example, it may be difficult to move a larger or more populous census tract between two districts without violating compactness or population balance. Bozkaya et al. (2011) similarly use tabu search and an adaptive memory heuristic with a multi-criteria objective function to construct district plans, but their application only involves 400 geographic units. Redistricting for a state in the U.S. can involve thousands (if not hundreds of thousands) of geographic units; therefore, individual Flip iterations may struggle to substantially alter district shapes or political composition (DeFord et al., 2021; Cho and Liu, 2021).

To avoid the limitations of Flip iterations, DeFord et al. (2021) introduce Recombination (ReCom), a spanning tree method for transitioning between plans within an ensemble-generation algorithm. Similar to local search, ensemble-generation methods for redistricting also depend on effective solution space exploration (Cho and Liu, 2021; Validi et al., 2021). The goal of ensemble-generation is to assess whether the partisan characteristics of a proposed district plan are extreme compared to an ensemble of randomly generated district plans. District plan ensemble-generation has been explored by Chen and Rodden (2013), Liu et al. (2016), Chikina et al. (2017), Fifield et al. (2020), Herschlag et al. (2020), and Cho and Liu (2021).

A ReCom iteration merges the units of two adjacent districts, creates a spanning tree on the induced subgraph of those units, and cuts one edge of this tree to create two new districts (DeFord et al., 2021). In contrast to a Flip iteration, a ReCom iteration can completely rearrange two adjacent districts by exchanging multiple units between both districts simultaneously; hence, ReCom random walks can reach a variety of district plans while maintaining redistricting constraints such as population balance. Additionally, ReCom iterations naturally favor compact districts, since compact districts yield more spanning trees than snake-like or otherwise convoluted districts (DeFord et al., 2021). Because of these qualities, we consider ReCom iterations within local search optimization algorithms for this study. Since ReCom iterations use larger transition neighborhoods than Flip iterations, local search can avoid low quality local optima (a concept similar to Very Large-Scale Neighborhood Search from Ahuja et al., 2000). Since ReCom iterations tend to favor compact districts, we can potentially circumvent the need for compactness constraints with political fairness objectives. Dobbs et al. (2023) first use ReCom iterations within local search optimization to construct congressional and state legislative district plans for Missouri. However, the focus of that study was to analyze the partisan impact of Missouri's political geography and new constitutional requirements, not to assess the performance of a novel local search iteration. In contrast, this paper directly compares the performance of several local search heuristics for redistricting that use ReCom or Flip iterations.

3. Problem formulation

Redistricting can be viewed as a graph partitioning problem, since the goal is to divide a region consisting of finitely many geographic units into districts. We can represent the region as a planar graph $G = (V, E)$, where the vertex set V represents the set of geographic units and there exists an edge $(u, v) \in E$ if and only if the units corresponding to u and v share a border segment of positive length. Then a district plan $z : V \to [K]$ is a partition of this graph into K parts representing the districts. For redistricting optimization, the goal is to solve

$$\min_{z \in Z_K(G)} f(z), \tag{1}$$

where $Z_K(G)$ is the set of all feasible district plans of a given region (i.e., the set of all feasible K-partitions of the region's graph representation G) and f is some redistricting objective (e.g., compactness). Ricca et al. (2013) provide additional details on the history of this graph-theoretic model for redistricting.

District plans in the U.S. are constructed using geographic units from a range of granularities such as census blocks, census block groups, census tracts, and counties. For example, Illinois has 369,978 census blocks, 9898 census block groups, 3265 census tracts, and 102 counties (U.S. Census Bureau, 2020h); Missouri and Tennessee have 1654 and 1701 census tracts, respectively. To maintain algorithm tractability, we construct district plans for Illinois, Missouri, and Tennessee using census tracts; these units constitute the vertex set of each state's graph representation. The U.S. Census Bureau provides spatial data for these units (U.S. Census Bureau, 2020h). Using geographic information system (GIS) software, it is possible to determine a unit adjacency list; this list constitutes the edge set of each state's graph representation. Following the 2020 census, Illinois has 17 congressional districts, Missouri has 8, and Tennessee has 9 (U.S. Census Bureau, 2020a). Therefore, the redistricting problems for Illinois, Missouri, and Tennessee are equivalent to partitioning 3265-node, 1654-node, and 1701-node planar graphs into $K = 17$, $K = 8$, and $K = 9$ parts, respectively.

The Illinois, Missouri, and Tennessee Constitutions impose few or no additional requirements for congressional redistricting (beyond the federal requirements), which allows for a wide range of legally viable plans (and potentially a wide range of fairness scores). Illinois's congressional plan from the 2021 redistricting cycle (Illinois House Democrats, 2021) is an example of a plan that satisfies legal requirements *and* possesses extreme partisan qualities and district shapes. Tennessee's congressional plan from the 2021 redistricting cycle (State of Tennessee, 2023) has more reasonable district shapes, but was criticized for cracking the city of Nashville (CNN, 2022). These three states also represent a range of voting tendencies (as mentioned in Section 3.1) and have different redistricting instance sizes. Missouri and Tennessee represent average-sized U.S. redistricting instances, since the average number of congressional districts per state following the 2020 census is 8.7 (U.S. Census Bureau, 2020a). In contrast, Illinois is a large-sized instance that can demonstrate the power of new local search techniques for redistricting; only California, Texas, Florida, and New York have more congressional districts than Illinois (while Pennsylvania has the same number) (U.S. Census Bureau, 2020a).

The rest of this section describes several common fairness objectives, redistricting constraints, and model details specific to congressional redistricting for Illinois, Missouri, and Tennessee (e.g., voting data, ideal district population).

3.1. Fairness objectives

There is not a universally accepted definition of "fairness" for redistricting. Some definitions of fairness quantify district shapes or consider competition between candidates. Other definitions examine the *packing* and *cracking* of political parties, where a party is packed if its voters are concentrated in a few districts where it wins by overwhelming margins, and cracked if its voters are diluted among several districts (Vieth v. Jubelirer, 2004). The local search algorithms in this study optimize district plans with respect to different metrics that quantify these common notions of fairness; the rest of this subsection describes each metric. It is important to note that we choose these metrics because they are thoroughly analyzed in the redistricting literature and are present in state constitutions or court cases, not because we necessarily endorse them as ideal measures of fairness. For more information on the trade-offs between these metrics, see Chen and Rodden (2013), Bernstein and Duchin (2017), Cho (2017), DeFord et al. (2020), Cain et al. (2018), DeFord et al. (2023), and Dobbs et al. (2023).

We refer to metrics that rely on election data as *political fairness metrics*. Election results are tabulated in voting precincts, which do not always coincide with census units and may change between elections. To construct districts with census tracts, we disaggregate precinct-level votes proportionally to census blocks, then aggregate the votes to census tracts. This procedure assumes that Democratic and Republican voters are spread evenly throughout the voting precincts, which might not be the case. To more accurately represent voter behavior, one could directly use voting precincts as units to construct district plans (instead of census tracts). However, voting precincts are often discontiguous, which may lead to discontiguous districts. For example, the Illinois voting precincts from 2016, 2018, and 2020 each have 15 precincts consisting of 10 or more discontiguous parts. In contrast, Illinois has one census tract in two discontiguous parts, which is straightforward to process manually (as discussed in Section 3.2.3). To obtain the precinct-level data needed to produce tract-level estimates according to this disaggregation procedure, we average votes from the 2016, 2018, and 2020 general election races for governor, United States Senate, and President (Voting and Election Science Team, 2021). According to this set of data and not including any third-party votes, Illinois voters are approximately 58.5% Democrat and 41.5% Republican, Missouri voters are approximately 46.5% Democrat and 53.5% Republican, and Tennessee voters are approximately 38.3% Democrat and 61.7% Republican.

3.1.1. Compactness

A *compact* district has a simple shape (such as a circle or square, as in Fig. 1(b)) with no convoluted segments or tendrils (as in Fig. 1(a)). Maintaining simple district shapes is often viewed as a proxy for maintaining political fairness, since intentional boundary manipulation for political gain can result in convoluted district shapes; as an example, the salamander-shaped Massachusetts district that inspired the term *gerrymander* in 1812 was constructed to pack Federalist voters (Griffith, 1907). Additionally, residents in compact districts have more in common geographically (Duchin and Tenner, 2018). It is important to note that prioritizing compactness does not guarantee political fairness and can inadvertently pack geographically clustered political parties (Vieth v. Jubelirer, 2004; Chen and Rodden, 2013); however, it is a common requirement in many state constitutions (National Conference of State Legislatures, 2021b).

There are several ways to quantify district compactness, such as summing district perimeters, comparing a district's area and perimeter, summing the distance from each geographic unit to the center of its district, or counting the number of cut edges in the graph representation (i.e., edges whose endpoints are in different districts) (Young, 1988; Duchin and Tenner, 2018; DeFord et al., 2021). We measure compactness as the sum of all cut edges; hence, we calculate the compactness fairness metric as

$$f_{comp}(z) := \sum_{(u,v)\in E} C(u, v, z), \tag{2}$$

where $C(u, v, z)$ is an indicator function that equals one when $z(u) \neq z(v)$ under district plan z and zero otherwise. This metric can be calculated using unit adjacency information. A convoluted district (such as Illinois's third congressional district from the 2021 redistricting cycle in Fig. 1(a)) contributes a larger number of cut edges to the total sum than a comparable district with a simpler shape (as in Fig. 1(b)). Hence, smaller values of $f_{comp}(z)$ indicate a district plan whose districts are more compact collectively.

3.1.2. Efficiency gap

The efficiency gap quantifies packing and cracking by comparing the percentage of votes wasted by the two parties for a given district plan. A vote in a particular district is considered *wasted* if it is cast for either the losing party, or the winning party in excess of the 50% it needs to win the election. Hence, a packed party wastes votes because it wins districts by excessive margins and a cracked party wastes votes because it loses districts by small margins. The efficiency gap computes the difference between the percentage of wasted votes for both parties (McGhee, 2014; Stephanopoulos and McGhee, 2015):

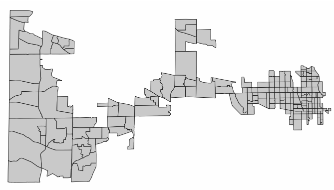
$$f_{eg}(z) := \frac{\sum_{k=1}^{K} \left(W_k^A(z) - W_k^B(z) \right)}{T}. \tag{3}$$

Here, T is the total number of votes cast for both parties, and $W_k^A(z)$ and $W_k^B(z)$ are the number of wasted votes in district k under district plan z for parties A and B, respectively. Efficiency gap values closer to zero indicate that both parties waste a similar number of votes, which means both parties are packed and cracked to a similar degree. A large positive (negative) value indicates that party A (B) wastes significantly more votes than party B (A). In Section 5, plans are optimized with respect to the absolute value of the efficiency gap, since the goal is to minimize bias against either party.

The efficiency gap has grown in popularity for U.S. redistricting because it is straightforward to calculate and it relies solely on past election results (not hypothetical election scenarios) (Bernstein and Duchin, 2017). For example, Missouri passed an amendment in 2020 that requires state legislative district plans to have efficiency gap values below 15% (Missouri Secretary of State, 2020). In Gill v. Whitford (2018) the efficiency gap was used to support claims that Wisconsin's state house district plan gave Republicans an undue advantage.

As the efficiency gap has grown in popularity, concerns have emerged regarding what this metric actually measures and how it is applied. As several authors note, including its creators, the efficiency gap reduces to a measure of *proportionality* (i.e., the idea that the fraction of seats a party wins should be proportional to the fraction of votes that it wins) (McGhee, 2014; Bernstein and Duchin, 2017; Cho, 2017; Warrington, 2018). Although proportionality is a widely known concept of fairness, it is not considered a constitutional right (Davis v. Bandemer, 1986; League of United Latin American Citizens v. Perry, 2006). A traditional view of proportionality is that parties should win seats at the *same* rate that they win votes; in contrast, the efficiency gap favors a "winner's bonus" view of proportionality (i.e., for a district plan with an efficiency gap of zero, parties are expected to win seats at *twice* the rate that they win votes) (McGhee, 2014; Bernstein and Duchin, 2017; Cho, 2017; Warrington, 2018). By favoring a winner's bonus view of proportionality, the efficiency gap penalizes the traditional view of proportionality (Chambers et al., 2017; Bernstein and Duchin, 2017). Additionally, for states with an extremely lopsided partisan split or very few districts, the efficiency gap may not take on any values that

(a) A tract approximation of Illinois's third congressional district from the 2021 redistricting cycle (Illinois House Democrats, 2021) with 232 cut edges



(b) A compact district with 80 cut edges

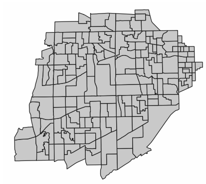


Fig. 1. Two examples of district shapes (for roughly the same location and population in the Chicago area).

sufficiently comply with winner's bonus proportionality (Bernstein and Duchin, 2017; Cho, 2017). This metric can also be sensitive to small changes in voter behavior; if a district plan has multiple competitive seats, one party might win them all in one election and lose them all in another election, producing dramatically different efficiency gap values for different sets of historical data (Bernstein and Duchin, 2017; Cho, 2017).

It is important to emphasize that Stephanopoulos and McGhee (2015) did not intend for the efficiency gap to be the sole judge of a district plan; their proposed doctrine also involved sensitivity analysis (to account for competitive districts) and examining a state's political geography and redistricting requirements (to account for lopsided partisan splits or unique geographic factors). Additionally, although proportionality is not a constitutional right, Stephanopoulos and McGhee (2015) note that the widely accepted goal of partisan gerrymandering is to "win as many seats as possible given a certain number of votes". Therefore, we include the efficiency gap in this study to encapsulate the widely held view that equates proportionality with fairness.

3.1.3. Mean-median

As an alternative to measures of proportionality, measures of symmetry have also been proposed (e.g., Grofman (1983), Grofman and King (2007), McDonald and Best (2015) and Katz et al. (2020)). As summarized in Grofman and King (2007), symmetry "requires that the number of seats one party would receive if it garnered a particular percentage of the vote be identical to the number of seats the other party would receive if it had received the same percentage of the vote". Measures of symmetry often explicitly rely on a political party's vote-seat curve (i.e., a function of the party's seat-share over all possible vote-shares), which can be constructed using a hypothetical uniform partisan swing (i.e., increasing/decreasing the party's vote-share uniformly across all districts) (Grofman and King, 2007; Katz et al., 2020). Therefore, symmetry broadly examines how the distribution of district vote-shares affects potential seat outcomes, rather than solely focusing on the nominal seat outcome. An extremely skewed (i.e., asymmetric) distribution of district vote-shares suggests that packing and cracking has occurred.

Despite its presence in the redistricting literature, there are concerns about symmetry as a standard for redistricting. The first concern is that symmetry metrics consider hypothetical vote-share scenarios. As Justice Kennedy stated in League of United Latin American Citizens v. Perry (2006), "even assuming a court could choose reliably among different models of shifting voter preferences, we are wary of adopting a constitutional standard that invalidates a map based on unfair results that would occur in a hypothetical state of affairs". The second concern is that symmetry can conflict with proportionality. In their practical study on symmetry metrics, DeFord et al. (2023) provide several scenarios in which a good symmetry score would result in an extremely disproportionate seat outcome.

Although the merits of symmetry continue to be debated, we include the mean-median symmetry metric in this study as an alternative to proportionality. Mean-median (also called *symmetry vote bias*) calculates the difference between the median and mean vote-share across all districts for one party (the *reference party*) (McDonald and Best, 2015):

$$f_{mm}(z) := \text{Median}\{F_k(z)\} - \frac{1}{K}\sum_{k=1}^{K} F_k(z). \tag{4}$$

In this formulation, F_k is the reference party's vote-share in district k under district plan z. Note that if each district has the same number of voters, the mean term in Eq. (4) equals the state-wide vote-share of the reference party (McDonald and Best, 2015); since voter turnout varies with changes to a district plan, the mean term varies slightly as well. A positive value for $f_{mm}(z)$ suggests that the reference party can secure half of the seats with fewer than half of the votes; a negative value suggests that the opposite party has the advantage (McDonald and Best, 2015; DeFord et al., 2023). Therefore, values close to zero are preferred, since they indicate symmetry of voting power between parties. As with the efficiency gap, the plans in Section 5 are optimized with respect to the absolute value of mean-median.

Mean-median offers a well-established history within redistricting; McDonald and Best (2015) note that district plan analyses have included comparisons of vote-share means and medians for over a century. Additionally, although mean-median does implicitly examine how the distribution of district vote-shares affects potential seat outcomes, computing this metric does not require the explicit generation and examination of hypothetical vote-share scenarios.

3.1.4. Competitiveness

Fairness toward political parties does not always align with fairness toward voters. Political parties or incumbents may desire guaranteed district wins (i.e., *safe seats*); in contrast, voters may desire competitive districts to feel like their vote truly affects the election outcome, to discourage candidate complacency, and to encourage candidates to cooperate in a bipartisan manner (Abramowitz et al., 2006; McCarty et al., 2009; DeFord et al., 2020). Arizona, Colorado, New York, and Washington require competitive congressional districts when possible (Ariz. Const. art. IV, pt. 2, §3, 2000; Colo. Const. art. V, §44, 2018; N.Y. Const. art. III, §4, 2014; Wash. Const. art. II, §43, 1983). However, as DeFord et al. (2020) note in their analysis of competitiveness, there is not a universally accepted competitiveness metric. Increasing the number of competitive districts might unintentionally crack one party or cause packing in the remaining, non-competitive districts (Swamy et al., 2022). Reducing the political imbalance in each district can decrease packing, but also cause the district vote-shares to homogeneously mirror the statewide vote-shares (King et al., 2018). Despite the acknowledged shortcomings, we measure the competitiveness of a district plan as the number of competitive districts; hence, we calculate the competitive fairness metric as:

$$f_{cmpttv}(z) := |\{k \in [K] \ : \ M_k(z) \le 0.07\}|, \tag{5}$$

where $M_k(z)$ is the difference between the winning party's expected vote-share and the losing party's expected vote-share in district k under district plan z (i.e., the expected *margin of victory*). Here, a competitive district is defined as a district within a 7% margin of victory (i.e., the election is expected to be at least as close as 46.5%–53.5%). Past studies have used a looser threshold of 10% to define a competitive district (e.g., Abramowitz et al. (2006) and DeFord et al. (2020)); we choose 7% because the Arizona Independent Redistricting Commission recently used this threshold in practice to construct congressional district plans for Arizona (Arizona Independent Redistricting Commission, 2021).

3.2. Redistricting constraints

This subsection characterizes the set of feasible congressional district plans (i.e., $Z_K(G)$ from Eq. (1)) for Illinois, Missouri, and Tennessee. The U.S. Constitution requires congressional districts to have nearly equal populations and satisfy the Voting Rights Act of 1965 (National Conference of State Legislatures, 2021b). The Illinois and Tennessee Constitutions do not impose additional requirements for congressional redistricting; the Missouri Constitution requires congressional districts to be contiguous and compact (Mo. Const. art. III, §45, 1945). Although not required by Illinois and Tennessee, we enforce contiguity for all three states. This decision aligns with redistricting practice following the 2000, 2010, and 2020 censuses, since the Illinois congressional districts are all contiguous and all discontiguous districts in the Tennessee congressional plans result from discontiguous census units (U.S. Census Bureau, 2020h). Similarly, we also enforce a compactness constraint for all three states to prevent Flip iterations from creating fractal-like district shapes that inflate district perimeters when the objective is not compactness (DeFord et al., 2020, 2021). Below we describe all requirements.

3.2.1. Population balance

The ideal district population is the total state population divided by the number of districts. The population balance constraint can be written as

$$(1-\delta)\overline{P} \leq P_k(z) \leq (1+\delta)\overline{P} \quad \text{for } k = 1, 2, \ldots, K, \tag{6}$$

where $\overline{P}$ is the ideal district population, $\delta \geq 0$ is the allowed population deviation, and $P_k(z)$ is the population of district k under district plan z. The U.S. Census Bureau provides population counts for geographic units from the 2020 decennial census that can be used to compute district populations (U.S. Census Bureau, 2020e,f,g).

Congressional district populations must be as close to the ideal population as possible (Wesberry v. Sanders, 1964); in practice, congressional districts typically deviate from the ideal district population by at most *one person* (National Conference of State Legislatures, 2021b). Since we construct district plans with census tracts and not census blocks (the finest granularity of census unit), single-person population balance is not readily achievable. For this reason, we allow district populations to deviate by at most 1% from the ideal population (i.e., we set $\delta = 0.01$ in Eq. (6)). This allowed deviation is small enough that the optimized plans could then be manually tuned to single-person population balance without substantially changing district shapes or political fairness (DeFord et al., 2021). Table 1 shows the total populations, ideal district populations, and allowed district population deviations under a 1% population balance for Illinois, Missouri, and Tennessee.

Table 1
Population information for the Illinois, Missouri, and Tennessee congressional redistricting instances.

State	Total population	Ideal district population	Allowed deviation under 1% population balance
Illinois	12,812,508	753,677	±7536
Missouri	6,154,913	769,364	±7693
Tennessee	6,910,840	767,871	±7678

3.2.2. Voting Rights Act

To satisfy the Voting Rights Act of 1965, states construct *majority–minority districts* (i.e., districts in which less than half of the population is non-Hispanic white) (Ballotpedia, 2017). Following the 2021 redistricting cycle, Illinois's congressional plan has five majority–minority districts in the Chicago area: three districts with Black/African American majorities/pluralities and two districts with Latino/Hispanic majorities/pluralities (Illinois House Democrats, 2021). Missouri's and Tennessee's congressional plans each have one district with a Black/African American majority/plurality. To match the number of majority–minority districts in these plans, we require the Missouri and Tennessee plans in this study to each have one plurality-Black/African American majority–minority district and the Illinois plans to have three plurality-Black/African American and two plurality-Latino/Hispanic majority–minority districts. Hence, the Voting Rights Act constraints can be written as

$$\sum_{k=1}^{K} B_k(z) = 3, \quad \sum_{k=1}^{K} L_k(z) = 2 \tag{7}$$

for Illinois and as

$$\sum_{k=1}^{K} B_k(z) = 1 \tag{8}$$

for Missouri and Tennessee. Here, $B_k(z)$ and $L_k(z)$ are indicator functions that equal one when district k under district plan z is a Black/African American or Latino/Hispanic plurality district, respectively, and zero otherwise. To enforce this constraint, we use race/ethnicity data from the 2020 decennial census (U.S. Census Bureau, 2020b,c,d,e,f,g). It is important to note that the creation of majority–minority districts in practice involves a more nuanced consideration of the historical and current discrimination facing particular minority groups, their geographical compactness, and their political cohesion (Thornburg v. Gingles, 1986). Although we do not examine these factors in our model, matching the number of majority–minority districts with particular pluralities in the current congressional plans provides one step toward including the Voting Rights Act in redistricting optimization.

3.2.3. Contiguity

Although not explicitly required by the Illinois and Tennessee Constitutions, we enforce district plan contiguity (i.e., we require the induced subgraph on each part of the partition to be connected). This constraint can be written as

$$Y_k(z) = 1 \quad \text{for } k = 1, 2, \ldots, K, \tag{9}$$

where $Y_k(z)$ is an indicator function that equals one when district k is contiguous under district plan z and zero otherwise.

Section 4.1 explains how each Flip and ReCom iteration maintains *graphical* contiguity (i.e., how it ensures that the induced subgraph on each part of the partition remains connected after an iteration). To ensure that this *graphical* contiguity translates to *visual* contiguity in the optimized district boundaries, the individual census tracts that constitute the vertex set of each state's graph representation must be contiguous. Missouri does not have any discontiguous census tracts. However, Illinois has one census tract (GEOID20 17167003603) that consists of two discontiguous parts; one of the parts is small (a single census block), with zero population. Similarly, Tennessee has five discontiguous census tracts (GEOID20s 47037015635, 47043060201, 47105060700, 47167040100, and 47185935000); for each of these tracts, one part is large and the remaining parts are small with little to no population.

Using GIS software, we treat these small parts as though they were annexed to adjacent tracts. After this pre-processing step, all census tracts are contiguous. Note that due to this pre-processing step, the optimized plans may separate the different parts of these tracts. We also do not include Illinois's two large water-only census tracts that cover Lake Michigan (GEOID20s 17031990000 and 17097990000). Excluding these tracts prevents a scenario in which a Chicago-area district consists of discontiguous population clusters "connected" by a large Lake Michigan tract. Lastly, Illinois, Missouri, and Tennessee have 15, 13, and 3 census tracts, respectively, that completely surround other tracts. To effectively execute Flip iterations, we merge these tracts with the tracts they surround. Since none of the surrounded tracts have sufficient population for an entire district, this merging does not alter the feasible solution space. Merging allows a Flip iteration to change the district membership of these surrounding tracts; without merging, any attempt to reassign one of these tracts will violate contiguity. ReCom iterations do not have this issue, since the formation of a spanning tree will reach every tract (although merging surrounded tracts would slightly reduce the time it takes to form/evaluate a spanning tree).

3.2.4. Compactness

When using a political fairness objective, we also impose a compactness threshold, set at 1.2 times the number of cut edges in the initial plan. In other words, given the initial plan $\hat{z}$, we enforce:

$$f_{comp}(z) \leq 1.2 \times f_{comp}(\hat{z}). \tag{10}$$

This constraint prevents Flip iterations from creating extremely convoluted district shapes when the objective is not compactness. As mentioned in Section 2, ReCom iterations naturally favor compact district shapes, so this constraint should be less necessary when using ReCom iterations. However, we apply the constraint regardless of iteration type for the sake of a controlled comparison. Section 5.4 discusses how optimizing a political fairness metric affects compactness for each iteration type based on empirical data.

Algorithm 1: A Single Iteration of Simple Flip

Input: A graph $G = (V, E)$, an objective f, a constraint set C, and a feasible district plan z
Output: A feasible district plan $\hat{z}$

Choose a node/district pair (u, k_2) uniformly at random such that u is adjacent to k_2 and $z(u) = k_1 \neq k_2$.
Define a district plan $\bar{z}(v) = \begin{cases} k_2, & \text{if } v = u \\ z(v), & \text{otherwise} \end{cases}$
Acceptable = CheckConstraintsObjective$(G, C, f, \bar{z})$
// See Sections 3.1 and 3.2 for examples of objectives/constraints to check.
if *Acceptable* **then**
  Let $\hat{z}(v) = \bar{z}(v)$ for each $v \in V$.
else
  Let $\hat{z}(v) = z(v)$ for each $v \in V$.
end
return $\hat{z}$

Algorithm 2: A Single Iteration of SA Flip

Input: A graph $G = (V, E)$, an objective f, a constraint set C, a feasible district plan z, and SA parameters T and β
Output: A feasible district plan $\hat{z}$ and SA parameter T

Choose a node/district pair (u, k_2) uniformly at random such that u is adjacent to k_2 and $z(u) = k_1 \neq k_2$.
Define a district plan $\bar{z}(v) = \begin{cases} k_2, & \text{if } v = u \\ z(v), & \text{otherwise} \end{cases}$
Feasible = CheckConstraints$(G, C, \bar{z})$
// See Section 3.2 for examples of constraints to check.
Improve = CheckObjective$(G, f, \bar{z})$
// See Section 3.1 for examples of objectives to check.
if *Feasible* and *Improve* **then**
  Let $\hat{z}(v) = \bar{z}(v)$ for each $v \in V$.
else
  Let u be a $U(0, 1)$ variate.
  if *Feasible* and *not Improve* and $u < e^{-\frac{|f(\bar{z}) - f(z)|}{T}}$ **then**
    Let $\hat{z}(v) = \bar{z}(v)$ for each $v \in V$.
    Set $T = \beta T$.
  else
    Let $\hat{z}(v) = z(v)$ for each $v \in V$.
  end
end
return $\hat{z}$, T

Algorithm 3: A Single Iteration of Greedy Flip

Input: A graph $G = (V, E)$, an objective f, a constraint set C, and a feasible district plan z
Output: A feasible district plan *Best*

Set $Best(v) = z(v)$ for each $v \in V$.
for *node-district pairs* (u, k_2) *such that* u *is adjacent to* k_2 *and* $z(u) = k_1 \neq k_2$ **do**
  Define a district plan $\bar{z}(v) = \begin{cases} k_2, & \text{if } v = u \\ z(v), & \text{otherwise} \end{cases}$
  Acceptable = CheckConstraintsObjective$(G, C, f, \bar{z})$
  // See Sections 3.1 and 3.2 for examples of objectives/constraints to check.
  if *Acceptable* and $|f(\bar{z}) - f(z)| > |f(Best) - f(z)|$ **then**
    Set $Best(v) = \bar{z}(v)$ for each $v \in V$.
  else
    Let u be a $U(0, 1)$ variate.
    if *Acceptable* and $|f(\bar{z}) - f(z)| = |f(Best) - f(z)|$ *and* $u < 0.5$ **then**
      Set $Best(v) = \bar{z}(v)$ for each $v \in V$
    end
  end
end
return *Best*

4. Methodology

The goal of this study is to compare the performance of several local search heuristics for redistricting that use either Flip or ReCom iterations. This section describes the algorithms we use to generate district plans and the experimental design we use to gather data.

4.1. Local search algorithms

A local search algorithm for redistricting begins with a feasible district plan, then improves an objective by transitioning between feasible plans with a sequence of local changes to district boundaries. This subsection describes the seven algorithms that we use in our experiments. Simple Flip and Simple ReCom are simple hill-climbing local search algorithms; SA Flip and SA ReCom are simulated annealing local search algorithms; Greedy Flip and Greedy ReCom are greedy local search algorithms; lastly, Sample ReCom is a sampling algorithm. We implement these algorithms because they are based on common local search variants or sampling methods (e.g., as in Ricca and Simeone (2008) and DeFord et al. (2020, 2021)).

The Flip algorithms transition between feasible plans with Flip iterations. Each Flip iteration is based on one or more *Flip proposals*, in which the algorithm selects a single geographic unit and evaluates how moving this unit from its current district to an adjacent district affects the constraints and objective. The three Flip-based algorithms differ in how many Flip proposals are created per iteration and when they are accepted. Simple Flip (Algorithm 1) creates one Flip proposal per iteration and accepts it if the resulting plan satisfies the constraints and does not worsen the objective. SA Flip (Algorithm 2) operates similarly, but allows moves that worsen the objective according to a probability that decreases as the algorithm runs. Occasionally allowing the objective to worsen can allow the algorithm to explore more of the solution space; this algorithm always outputs the best solution found during its run (which is not necessarily the last solution it found). We experimentally calibrate the initial temperature (T_0), cooling rate (β), and threshold (η) simulated annealing parameters for SA Flip based on the four different objectives and list these parameter values in Table 2. The parameters T_0 and β determine the acceptance probabilities and η determines algorithm termination; see Kirkpatrick et al. (1983) and Ricca and Simeone (2008) for more information on simulated annealing and its parameters. Greedy Flip (Algorithm 3) evaluates all possible Flip proposals and accepts one that satisfies the constraints and yields the best objective improvement. When Greedy Flip is evaluating the Flip proposals, if it encounters one that yields the same objective improvement as the current best proposal, it decides randomly whether or not to replace the current best proposal with the new proposal.

Each ReCom algorithm transitions between feasible plans with ReCom iterations. As with the Flip algorithms, each ReCom iteration is based on one or more *ReCom proposals*, in which the algorithm creates a spanning tree on the induced subgraph of two adjacent districts, then

Algorithm 4: A Single Iteration of Simple ReCom

Input: A graph $G=(V,E)$, an objective f, a constraint set C, a feasible district plan z, and a max number of spanning tree attempts M
Output: A feasible district plan $\hat{z}$

Choose an edge (u,v) uniformly at random such that $z(u)=k_1 \neq k_2 = z(v)$.
Let H be the induced subgraph on the nodes of $k_1 \cup k_2$.
Let *ValidEdges* $=\emptyset$ and *Attempts* $=0$.
while *ValidEdges* $=\emptyset$ and *Attempts* $<M$ **do**
  Set $Attempts = Attempts + 1$.
  Form a spanning tree T of H.
  for *edge in T* **do**
    Set $(T_1,T_2)=T\backslash edge$.
    Balanced = CheckPopulationBalance(T_1,T_2)
    // Check if this cut satisfies population balance (Equation (6)).
    if *Balanced* **then**
      Add $edge$ to $ValidEdges$.
    end
  end
end
if *ValidEdges* $=\emptyset$ **then**
  Let $\hat{z}(v)=z(v)$ for each $v\in V$.
else
  Choose an edge uniformly at random from *ValidEdges*.
  Set $(T_1,T_2)=T\backslash edge$.
  Define a district plan $\bar{z}(w)=\begin{cases}k_1, & \text{if } w\in T_1\\ k_2, & \text{if } w\in T_2\\ z(w), & \text{otherwise}\end{cases}$
  Acceptable = CheckConstraintsObjective($G,C,f,\bar{z}$)
  // See Sections 3.1 and 3.2 for examples of objectives/constraints to check.
  if *Acceptable* **then**
    Let $\hat{z}(v)=\bar{z}(v)$ for each $v\in V$.
  else
    Let $\hat{z}(v)=z(v)$ for each $v\in V$.
  end
end
return $\hat{z}$

Algorithm 5: A Single Iteration of SA ReCom

Input: A graph $G=(V,E)$, an objective f, a constraint set C, a feasible district plan z, a max number of spanning tree attempts M, and SA parameters T and β
Output: A feasible district plan $\hat{z}$ and SA parameter T

Choose an edge (u,v) uniformly at random such that $z(u)=k_1 \neq k_2 = z(v)$.
Let H be the induced subgraph on the nodes of $k_1 \cup k_2$.
Let *ValidEdges* $=\emptyset$ and *Attempts* $=0$.
while *ValidEdges* $=\emptyset$ and *Attempts* $<M$ **do**
  Set $Attempts = Attempts + 1$.
  Form a spanning tree T of H.
  for *edge in T* **do**
    Set $(T_1,T_2)=T\backslash edge$.
    Balanced = CheckPopulationBalance(T_1,T_2)
    // Check if this cut satisfies population balance (Equation (6)).
    if *Balanced* **then**
      Add $edge$ to $ValidEdges$.
    end
  end
end
if *ValidEdges* $=\emptyset$ **then**
  Let $\hat{z}(v)=z(v)$ for each $v\in V$.
else
  Choose an edge uniformly at random from *ValidEdges*.
  Set $(T_1,T_2)=T\backslash edge$.
  Define a district plan $\bar{z}(w)=\begin{cases}k_1, & \text{if } w\in T_1\\ k_2, & \text{if } w\in T_2\\ z(w), & \text{otherwise}\end{cases}$
  Feasible = CheckConstraints($G,C,\bar{z}$)
  // See Section 3.2 for examples of constraints to check.
  Improve = CheckObjective($G,f,\bar{z}$)
  // See Section 3.1 for examples of objectives to check.
  if *Feasible* and *Improve* **then**
    Let $\hat{z}(v)=\bar{z}(v)$ for each $v\in V$.
  else
    Let u be a $U(0,1)$ variate.
    if *Feasible* and *not Improve* and $u<e^{-\frac{|f(\bar{z})-f(z)|}{T}}$ **then**
      Let $\hat{z}(v)=\bar{z}(v)$ for each $v\in V$.
      Set $T=\beta T$.
    else
      Let $\hat{z}(v)=z(v)$ for each $v\in V$.
    end
  end
end
return $\hat{z}$, T

chooses an edge to cut in this tree from the set of edges that produce two new population-balanced districts when cut (DeFord et al., 2021). The algorithm then evaluates how this change in district boundaries affects the constraints and objective. Simple ReCom (Algorithm 4) creates one ReCom proposal per iteration, choosing a population-balanced cut uniformly at random. As with Simple Flip, this ReCom proposal is accepted if the two new districts satisfy all constraints and do not worsen the objective. SA ReCom (Algorithm 5) is analogous to SA Flip; Table 2 lists the initial temperature, cooling rate, and threshold parameters we

Algorithm 6: A Single Iteration of Greedy ReCom

Input: A graph $G=(V,E)$, an objective f, a constraint set C, a feasible district plan z, a max number of spanning tree attempts M, and a required number of spanning trees t
Output: A feasible district plan $Best$

Set $Best(v)=z(v)$ for each $v\in V$.
Let $NumTrees=0$.
Let $Candidates=\emptyset$.
while $NumTrees < t$ **do**
  Choose an edge (u,v) uniformly at random such that $z(u)=k_1 \neq k_2 = z(v)$.
  Let H be the induced subgraph on the nodes of $k_1 \cup k_2$.
  Let *ValidEdges* $=\emptyset$ and *Attempts* $=0$.
  while *ValidEdges* $=\emptyset$ and *Attempts* $<M$ **do**
    Set $Attempts = Attempts + 1$.
    Form a spanning tree T of H.
    for *edge in T* **do**
      Set $(T_1,T_2)=T\backslash edge$.
      Balanced = CheckPopulationBalance(T_1,T_2)
      // Check if this cut satisfies population balance (Equation (6)).
      if *Balanced* **then**
        Add $edge$ to $ValidEdges$.
      end
    end
  end
  if *ValidEdges* $\neq\emptyset$ **then**
    Set $NumTrees = NumTrees + 1$.
    for *edge in ValidEdges* **do**
      Set $(T_1,T_2)=T\backslash edge$.
      Define a district plan $\bar{z}(w)=\begin{cases}k_1, & \text{if } w\in T_1\\ k_2, & \text{if } w\in T_2\\ z(w), & \text{otherwise}\end{cases}$
      Add $\bar{z}$ to $Candidates$.
    end
  end
end
for $\bar{z}$ *in Candidates* **do**
  Acceptable = CheckConstraintsObjective($G,C,f,\bar{z}$)
  // See Sections 3.1 and 3.2 for examples of objectives/constraints to check.
  if *Acceptable* and $|f(\bar{z})-f(z)| > |f(Best)-f(z)|$ **then**
    Set $Best(v)=\bar{z}(v)$ for each $v\in V$.
  else
    Let u be a $U(0,1)$ variate.
    if *Acceptable* and $|f(\bar{z})-f(z)| = |f(Best)-f(z)|$ *and* $u<0.5$ **then**
      Set $Best(v)=\bar{z}(v)$ for each $v\in V$
    end
  end
end
return $Best$

Algorithm 7: A Single Iteration of Sample ReCom (DeFord et al., 2021)

Input: A graph $G=(V,E)$, a constraint set C, a feasible district plan z, and a max number of spanning tree attempts M
Output: A feasible district plan $\hat{z}$

Choose an edge (u,v) uniformly at random such that $z(u)=k_1 \neq k_2 = z(v)$.
Let H be the induced subgraph on the nodes of $k_1 \cup k_2$.
Let *ValidEdges* $=\emptyset$ and *Attempts* $=0$.
while *ValidEdges* $=\emptyset$ and *Attempts* $<M$ **do**
  Set $Attempts = Attempts + 1$.
  Form a spanning tree T of H.
  for *edge in T* **do**
    Set $(T_1,T_2)=T\backslash edge$.
    Balanced = CheckPopulationBalance(T_1,T_2)
    // Check if this cut satisfies population balance (Equation (6)).
    if *Balanced* **then**
      Add $edge$ to $ValidEdges$.
    end
  end
end
if *ValidEdges* $=\emptyset$ **then**
  Let $\hat{z}(v)=z(v)$ for each $v\in V$.
else
  Choose $edge$ uniformly at random from *ValidEdges*.
  Set $(T_1,T_2)=T\backslash edge$.
  Define a district plan $\bar{z}(w)=\begin{cases}k_1, & \text{if } w\in T_1\\ k_2, & \text{if } w\in T_2\\ z(w), & \text{otherwise}\end{cases}$
  Acceptable = CheckConstraints($G,C,\bar{z}$)
  // See Section 3.2 for examples of constraints to check.
  if *Acceptable* **then**
    Let $\hat{z}(v)=\bar{z}(v)$ for each $v\in V$.
  else
    Let $\hat{z}(v)=z(v)$ for each $v\in V$.
  end
end
return $\hat{z}$

used. The definition of Greedy ReCom (Algorithm 6) differs slightly from Greedy Flip. A typical greedy local search algorithm (e.g., Greedy Flip) evaluates *all* possible proposals and accepts one that is feasible with the best objective improvement. A greedy local search algorithm with ReCom iterations that is truly analogous to Greedy Flip would consider all population-balanced cuts of all spanning trees for all pairs of adjacent districts; performing a single iteration would be prohibitively

Table 2
Initial temperature (T_0), cooling rate (β), and threshold (η) simulated annealing parameters for SA Flip and SA ReCom, for each objective.

Algorithm	Objective	T_0	β	η
SA Flip	Compactness	2	0.999	0.4
	Efficiency Gap	0.01	0.99	0.0001
	Mean-Median	0.005	0.999	0.00005
	Competitiveness	1.5	0.9	0.5
SA ReCom	Compactness	5	0.98	0.9
	Efficiency Gap	0.01	0.9	0.0015
	Mean-Median	0.005	0.85	0.0001
	Competitiveness	0.5	0.8	0.25

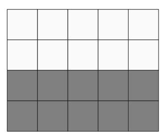


Fig. 2. An example region with 20 units partitioned into two districts of equal size. Allowing single-unit deviation from the ideal district size of 10, there are 10 Simple Flip neighbors and 727 Simple ReCom neighbors from this district plan.

expensive. Therefore, to maintain computational tractability, Greedy ReCom instead considers multiple ReCom proposals for one pair of adjacent districts. Greedy ReCom generates t spanning trees that have at least one population-balanced cut, checks *all* population-balanced cuts from all t trees, and accepts one that satisfies all constraints and yields the best objective improvement (as with Greedy Flip, ties are broken randomly). In our experiments, we set $t = 2$. Lastly, Sample ReCom (Algorithm 7) transitions between feasible plans without considering any objective (i.e., performs a ReCom random walk, as in DeFord et al. (2021)).

Given a feasible district plan, consider the size of its local search neighborhood using the Flip and ReCom algorithms. While each iteration type results in a boundary change for two adjacent districts, the size of their neighborhoods can substantially differ based on the iteration type. For example, the number of neighbors using Simple Flip is equal to the number of units that are on the border between two adjacent districts (i.e., *border nodes*). Using Simple ReCom, the number of neighbors is equal to the number of contiguous, population-balanced bipartitions of the induced subgraph on two adjacent districts. Hence, the feasible Simple Flip neighborhood is a subset of the feasible Simple ReCom neighborhood, and the latter is typically much larger in practice. For example, Fig. 2 shows a two-district plan for a simple 4×5 grid graph that has 727 Simple ReCom neighbors, but only 10 Simple Flip neighbors (under a population balance constraint that requires each district to contain between 9 and 11 units).

It is important to note that we define "improving moves" for these local search algorithms as moves that *do not worsen* the objective, rather than as moves that *strictly improve* the objective. Objectives such as compactness, efficiency gap, and mean-median are sensitive enough that swapping a single unit between districts can change their values. However, a competitiveness objective (as measured by the number of competitive seats) might not change even with multiple unit swaps. Therefore, to encourage exploration of the solution space, improving moves are those that strictly improve *or* maintain the current objective value. DeFord et al. (2020) implement a similar acceptance rule when they examine competitiveness metrics.

When the Flip algorithms (Algorithms 1–3) choose a unit-district pair (u, k_2), they must check whether moving unit u to district k_2 from its current district k_1 creates a feasible plan and determine the new objective value. Population balance, majority–minority districts, and compactness are straightforward to check. One can subtract the population of unit u from its original district k_1, add it to district k_2, then calculate the new percent deviations from $\overline{P}$. Racial/ethnic group district populations can be modified in a similar manner. The total number of cut edges can be modified based on the edges between unit u and districts k_1 and k_2.

The political fairness metrics can also be updated efficiently after unit u is transferred. Updating votes in both districts is similar to updating population; once the votes are updated, one can recalculate a political fairness objective. To quickly recalculate the efficiency gap, we maintain a running tally of wasted votes for each party in each district. Hence, one can use the updated votes to recalculate the wasted votes in districts k_1 and k_2 (i.e., $W_{k_1}^{Dem}(z)$, $W_{k_1}^{Rep}(z)$, $W_{k_2}^{Dem}(z)$, and $W_{k_2}^{Rep}(z)$ from Eq. (3)). For mean-median, one can use the updated votes to calculate the reference party's vote-shares in k_1 and k_2 (i.e., $F_{k_1}(z)$ and $F_{k_2}(z)$ in Eq. (4)), after which the mean and median vote-share can be recalculated. To determine if this transition creates/eliminates a competitive district, one can similarly use the new party vote-shares in k_1 and k_2 to calculate the expected margins of victory in k_1 and k_2 (i.e., $M_{k_1}(z)$ and $M_{k_2}(z)$ in Eq. (5)).

Checking contiguity is more challenging. To simplify the check, we maintain a record of border nodes. Since Algorithms 1–3 explicitly choose a district k_2 that is adjacent to border node u, it is guaranteed that district k_2 remains contiguous after receiving u. Therefore, it is only necessary to check that k_1 remains contiguous after removing u. To quickly evaluate a candidate move for contiguity, we use the hole-free geo-graph method introduced by King et al. (2012). This method relies on the *augmented* neighborhood of unit u, $R(u)$ (i.e., any unit that shares boundary segments *or* isolated points with u) to reduce computation time. Note that $|R(u)| \geq |N(u)|$. To use this method, we additionally require that district plans optimized with the Flip algorithms remain hole-free (i.e., we forbid any district from completely surrounding another district).

To implement ReCom iterations for Algorithms 4–7, we use the open-source GerryChain Python package (Voting Rights Data Institute, 2018). Just as the Flip implementation maintains a record of border nodes, the ReCom implementation maintains a record of cut edges. Each ReCom iteration chooses two adjacent districts to merge and repartition by selecting a cut edge uniformly at random. To create a spanning tree on the induced subgraph of two adjacent districts, GerryChain uses Kruskal's algorithm with randomized edge weights, then selects a root uniformly at random from the set of vertices with degree greater than one. To find population-balanced cuts of this spanning tree, GerryChain implements a memoization procedure: for each node in the tree (beginning with the leaves), the population of the subtree rooted at that node is recorded and checked for population balance. Here a subtree only passes the population balance check if its population *and* the population of its complement in the spanning tree are within the acceptable population deviation. If the spanning tree does not yield any population-balanced cuts, the algorithm draws a new spanning tree; we limit the number of spanning trees to 250. Note that we report this process as a single iteration in Section 5, even if more than one spanning tree is drawn.

Next, the ReCom algorithms must check whether this candidate plan satisfies all other constraints and determine the new objective value. Note that contiguity is automatically preserved, since the two new districts result from cutting exactly one edge of a spanning tree. GerryChain identifies which units have changed district assignments, recomputes district totals for racial/ethnic group populations and votes/vote-shares, identifies which edges are no longer cut edges, and identifies which edges have become cut edges. With these updates, the ReCom algorithms can check that majority–minority districts are maintained and recalculate the four fairness objectives.

It is clear to see that a single Flip iteration has a faster time complexity than a single ReCom iteration. The time complexity of Flip operations largely depends on the size of a unit's augmented neighborhood $|R(u)|$ or the number of districts K (e.g., checking contiguity is $O(|R(v)|)$ time, updating mean-median is $O(K)$ time), while the time complexity of ReCom operations largely depends on the size of the induced subgraph of the two merged districts H (e.g., creating a spanning tree on the planar subgraph H is $O(V[H]\log V[H])$ time, memoizing subtree populations is $O(V[H])$ time). As Section 5 confirms, the slower time complexity of ReCom iterations translates to a slower run time for realistic redistricting instances. However, as DeFord et al. (2021) emphasize, this increase in computational effort comes with more effective solution space exploration. Section 5 shows that the ReCom algorithms frequently produce district plans with higher quality objective values than the Flip algorithms, even within short time periods.

4.2. Experimental design

To assess and compare the performance of the local search algorithms in Section 4.1, we use these algorithms to optimize Illinois, Missouri, and Tennessee congressional district plans with respect to each fairness objective discussed in Section 3.1 (compactness, efficiency gap, mean-median, and competitiveness) under the constraints discussed in Section 3.2. First, to examine the trade-off between solution quality and run time for these algorithms, we perform timed experiments and report the best objective value each algorithm achieves in 30, 300, 1800, and 3600 s. For each timed experiment with a particular state, algorithm, and objective, the algorithm starts at the initial solution, performs iterations until it approximately converges to a local optimum, then restarts at the same initial solution. The algorithm repeats this process until the time limit is reached. The algorithms that converge to a local optimum quickly (e.g., Simple Flip) will be able to produce more optimized plans than the algorithms that converge more slowly (e.g., Greedy ReCom). Note that since Sample ReCom is not an optimization algorithm, it does not converge to a local optimum; therefore, we simply report the best solution from the sample it creates within the time limit.

Second, to provide insight into the average performance of each algorithm, we execute untimed experiments of each algorithm, for each objective. For each untimed experiment with a particular state, algorithm, and objective, the algorithm starts at an initial solution, performs iterations until it approximately converges to a local optimum, then restarts at a different initial solution. The algorithm repeats this process until it has generated 50 optimized plans. For each algorithm's collection of optimized plans, we compare the average run time and best, worst, and average objective values. We also compare the final compactness values for each algorithm when the objective is not compactness. Again, Sample ReCom does not converge to a local optimum; therefore, we exclude Sample ReCom from the untimed experiments. The rest of this subsection describes the initial solutions we use and how we assess algorithm convergence.

For the timed experiments, we use the congressional district plans for Illinois, Missouri, and Tennessee from the 2021 redistricting cycle as initial solutions. Using GIS software, we obtain a census tract approximation of each plan (shown in Fig. 3). Since the original plans splits many tracts, we initially assign each split tract to the district which contains its centroid. No adjustments are needed to maintain majority–minority districts, while minor adjustments are needed to maintain contiguity and population balance.

For the untimed experiments, we mimic a random restart procedure by generating 50 distinct, feasible district plans for each state. Using multiple initial solutions reveals the degree to which each algorithm is dependent on the initial solution, thereby assessing the robustness of each algorithm. To generate these initial plans, we execute Sample ReCom 50 times for 25 iterations each, starting from the tract approximation of each state's current congressional plan. It is possible to implement a Sample ReCom random restart procedure within the local search algorithms, but we chose to generate the plans in advance so that the initial solutions would be consistent across all experiments.

Starting from an initial solution, each algorithm continues until it approximately converges to a local optimum. Convergence is measured by comparing objective values at successive iterations; if the objective has not improved by at least ϵ units for N successive iterations, the algorithm terminates. Based on preliminary experiments, we set $N =$ 10,000 for Simple Flip and SA Flip, $N = 250$ for Greedy Flip with the compactness and competitive objectives, $N = 100$ for Greedy Flip with the efficiency gap and mean-median objectives, and $N = 250$ for Simple ReCom, SA ReCom, and Greedy ReCom. We set $\epsilon = 2$ for compactness, $\epsilon = 0.001$ for efficiency gap and competitiveness, and $\epsilon = 0.0001$ for mean-median. Note that all ϵ-values in $(0, 1)$ impose the same convergence requirement for the competitiveness objective (namely, *no* improvement for N iterations), since this objective only takes integer values.

5. Results and discussion

Here we present congressional district plans for Illinois, Missouri, and Tennessee, satisfying the constraints discussed in Section 3.2 and optimized for the different fairness metrics discussed in Section 3.1. As described in Section 4.2, we performed both timed experiments and untimed experiments. All experiments are run on a 3.10 GHz Core i5 2400-CPU machine with 8 GB of RAM. Code and data can be found on GitHub (https://github.com/kierwynd/Flip_ReCom_LocalSearch).

The timed experiments compare the best objective values each algorithm produced within time limits of 30, 300, 1800, and 3600 s to illustrate the potential trade-off between solution quality and run time for each algorithm. Figs. 4–7 present the best compactness, efficiency gap, mean-median, and competitiveness values, respectively, that each algorithm achieved within each time limit for Illinois, Missouri, and Tennessee. Tables A.4–A.7 in Appendix A report all values used for these figures.

The untimed experiments evaluate the performance of each algorithm as it generated 50 plans to provide insight into the average performance of each algorithm; this type of comparison is also useful if one's goal is to generate several plans with good fairness values, rather than a single fair plan. Table 3 reports the average run time per plan for each algorithm and objective. Figs. 8–11 display the distribution of compactness, efficiency gap, mean-median, and competitiveness values, respectively, for the initial 50 plans and the sets of 50 optimized plans for each algorithm and each state. Then Tables B.8–B.11 in Appendix B report the average, best, and worst objective values for each set of 50 plans to summarize these distributions.

The results of these experiments convey the following key insights:

- The ReCom algorithms consistently achieved better solutions than the Flip algorithms for the efficiency gap and competitiveness objectives across all three states.
- SA Flip occasionally achieved solutions at least as good as the ReCom algorithms for the compactness and mean-median objectives.
- The ReCom algorithms took substantially longer on average to converge than the Flip algorithms across all three states.
- SA Flip frequently produced the best solutions out of the three Flip algorithms.
- SA ReCom and Greedy ReCom tended to produce solutions marginally better than Simple ReCom.
- The ReCom algorithms performed more consistently across all three states than the Flip algorithms.
- The ReCom algorithms maintained compactness better than the Flip algorithms across all three states when the objective was not compactness.

The rest of this section explains these insights in more detail. Section 5.1 compares objective improvement and algorithm run time for both iteration types, Section 5.2 evaluates the performance of each algorithm variant, Section 5.3 discusses how algorithm performance differs between the three states, and Section 5.4 examines the compactness of plans optimized for political fairness metrics.

Table 3
The average run times (seconds) for each algorithm to generate 50 plans with one of four objectives in the untimed experiments.

State	Objective	Sample ReCom (Initial Plans)	Simple Flip	SA Flip	Greedy Flip	Simple ReCom	SA ReCom	Greedy ReCom
Illinois	Compactness	8.08	6.97	23.17	157.84	368.83	533.23	648.21
	Efficiency Gap	8.08	2.06	3.29	26.84	81.71	89.17	221.85
	Mean-Median	8.08	1.94	16.43	28.22	223.05	219.15	370.40
	Competitiveness	8.08	5.09	28.26	94.65	108.98	195.45	215.34
Missouri	Compactness	7.13	2.77	17.46	38.49	179.21	494.30	319.62
	Efficiency Gap	7.13	2.16	6.74	6.53	126.57	126.74	230.39
	Mean-Median	7.13	2.12	18.33	7.69	142.17	148.62	246.06
	Competitiveness	7.13	2.89	9.33	22.15	77.06	132.58	162.18
Tennessee	Compactness	5.14	3.28	18.37	37.26	178.39	419.10	291.74
	Efficiency Gap	5.14	2.54	7.77	6.71	118.83	123.32	235.03
	Mean-Median	5.14	1.74	20.21	4.72	114.75	113.16	216.49
	Competitiveness	5.14	3.02	8.55	21.13	67.12	126.13	130.21

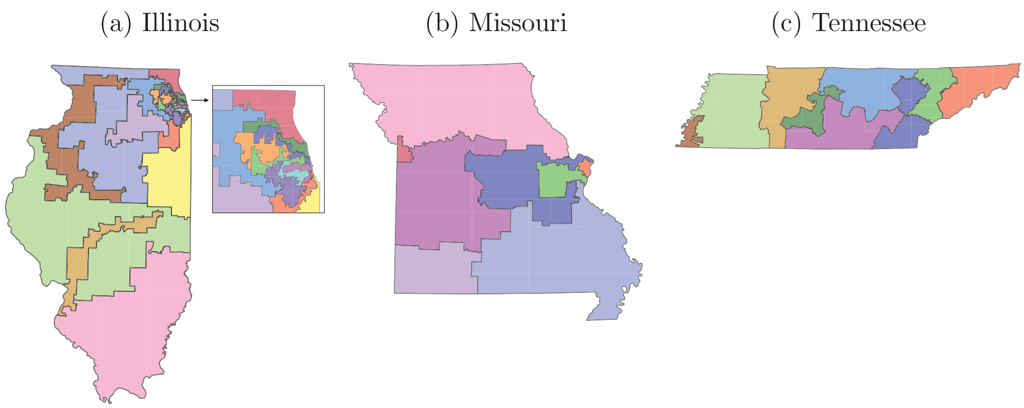


Fig. 3. Census tract approximations of congressional district plans in Illinois, Missouri, and Tennessee from the 2021 redistricting cycle. The inset for Illinois shows the Chicago area.

5.1. Comparison of iteration types

For efficiency gap and competitiveness, all ReCom algorithms yielded better objective values than all Flip algorithms. During the timed experiments (Figs. 5 and 7), each ReCom algorithm produced an objective value similar to or better than each Flip algorithm at all time limits across all three states. During the untimed experiments (Figs. 9 and 11), each ReCom algorithm produced better average efficiency gap and competitiveness values than each Flip algorithm across each collection of 50 optimized plans for all three states.

It is difficult to substantially improve the efficiency gap or competitiveness with small changes in district vote-shares from Flip iterations. As mentioned in Section 3.1.2, the efficiency gap is a measure of proportionality (specifically, winner's bonus proportionality); consequently, district plans with excellent efficiency gap values also have proportional expected seat outcomes. Therefore, while small vote-share changes can marginally improve the efficiency gap, substantial improvements can only result from flipping seats between the two parties (which is less likely with Flip iterations). By similar reasoning, individual Flip iterations are less likely to change a non-competitive district into a competitive district. In contrast, ReCom iterations can make substantial changes to the boundary between two districts; therefore, a single ReCom iteration is more likely to flip a seat from one party to the other or change a district from non-competitive to competitive. Hence, even though the ReCom algorithms took longer on average to *converge* than the Flip algorithms (as shown in Table 3), they were still able to achieve better efficiency gap and competitiveness values, regardless of time limits.

In contrast to efficiency gap and competitiveness, the Flip algorithms occasionally produced objective values at least as good as the ReCom algorithms for compactness and mean-median. The timed experiments (Figs. 4 and 6) show a clearer trade-off between solution quality and run time, since some Flip algorithms yielded better values than the ReCom algorithms for the shorter time limits. For example, all three Flip algorithms achieved a better mean-median value within 30 s for Illinois than SA ReCom, Greedy ReCom, and Sample ReCom. Similarly, SA Flip produced the best compactness value within 30 and 300 s for Missouri and Tennessee. During the untimed experiments (Figs. 8 and 10), SA Flip frequently yielded average compactness and mean-median values similar to the ReCom algorithms. However, Simple Flip and Greedy Flip produced substantially worse average compactness and mean-median values than SA Flip.

Unlike efficiency gap and competitiveness, improving compactness or mean-median does not depend on flipping seats from one party to another or from non-competitive to competitive. Therefore, there is more potential for a Flip algorithm to substantially improve compactness and mean-median via many small, incremental changes. Hence, a Flip iteration local search variant (e.g., SA Flip) may be sufficient to substantially improve these objectives. Since SA Flip converged more than five times faster than the ReCom algorithms (as shown in Table 3), it may be preferable to use SA Flip when optimizing compactness and mean-median.

5.2. Comparison of algorithm variants

Due to their relatively small local search neighborhoods, Flip iterations may create solution spaces with numerous low-quality local optima; hence, Simple Flip and Greedy Flip often produced mediocre solutions. Greedy Flip also tended to have the longest average run time out of the three Flip algorithms (as shown in Table 3). Since simulated annealing allows for better exploration of the solution space in the presence of low quality local optima, SA Flip frequently achieved solutions substantially better than Simple Flip and Greedy Flip. During the timed experiments (Figs. 4–6), SA Flip achieved better objective values than Simple Flip and Greedy Flip at each time limit for compactness, efficiency gap, and mean-median across all three states. Similarly,

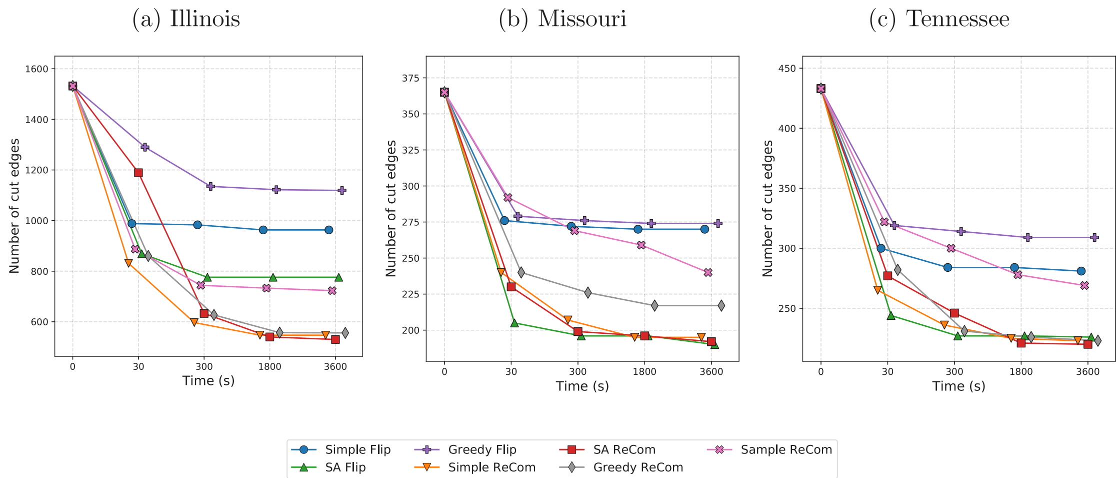


Fig. 4. Plots showing the best compactness values each algorithm achieves in 30, 300, 1800, and 3600 s for Illinois, Missouri, and Tennessee. For visual clarity, the time limits on the horizontal axis are spaced uniformly (rather than spaced to scale) and horizontal noise has been added.

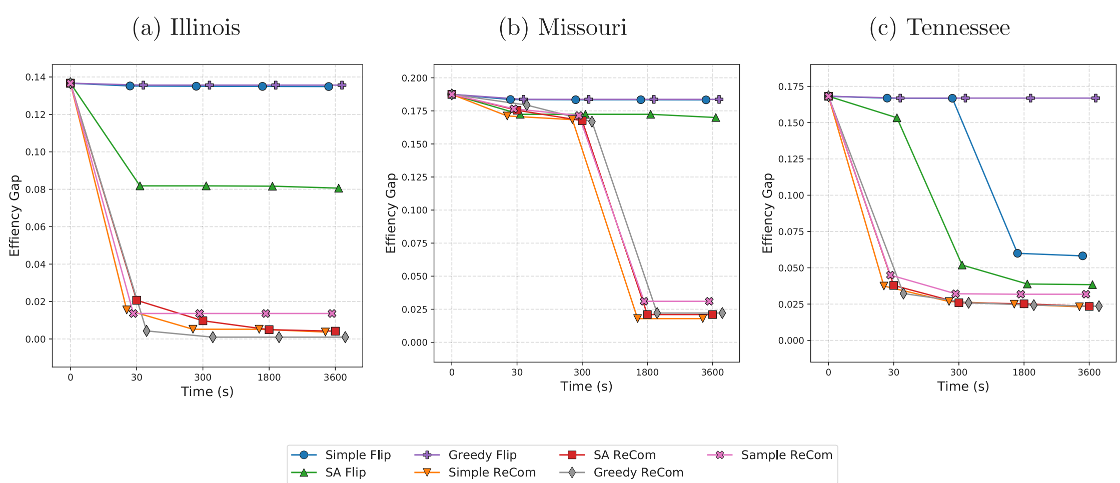


Fig. 5. Plots showing the best efficiency gap values each algorithm achieves in 30, 300, 1800, and 3600 s for Illinois, Missouri, and Tennessee. For visual clarity, the time limits on the horizontal axis are spaced uniformly (rather than spaced to scale) and horizontal noise has been added.

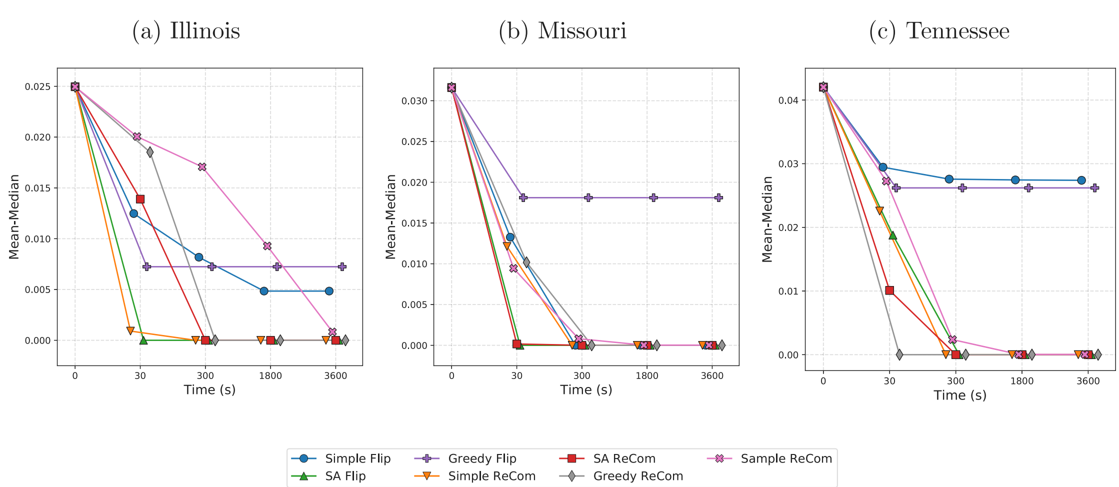


Fig. 6. Plots showing the best mean-median values each algorithm achieves in 30, 300, 1800, and 3600 s for Illinois, Missouri, and Tennessee. For visual clarity, the time limits on the horizontal axis are spaced uniformly (rather than spaced to scale) and horizontal noise has been added.

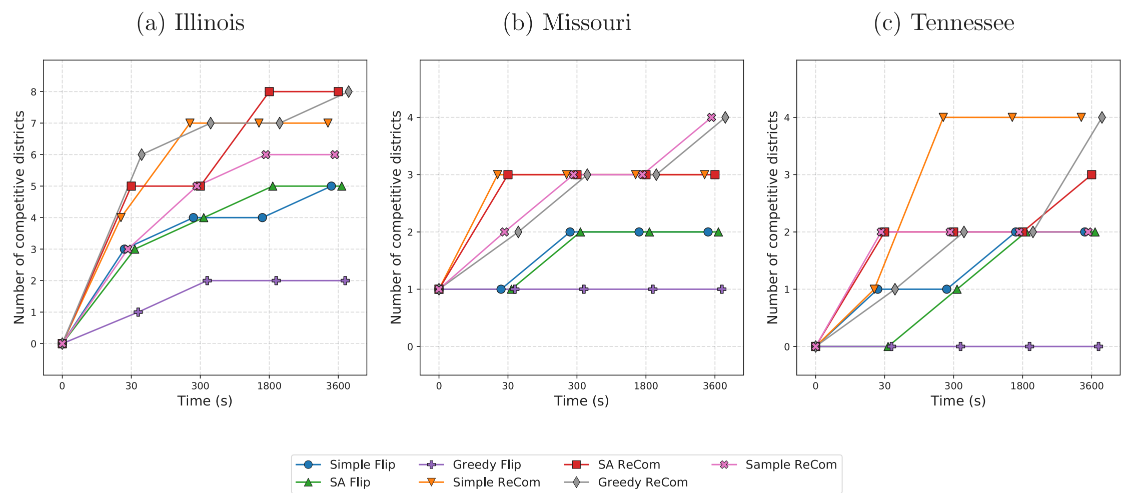


Fig. 7. Plots showing the best competitiveness values each algorithm achieves in 30, 300, 1800, and 3600 s for Illinois, Missouri, and Tennessee. For visual clarity, the time limits on the horizontal axis are spaced uniformly (rather than spaced to scale) and horizontal noise has been added.

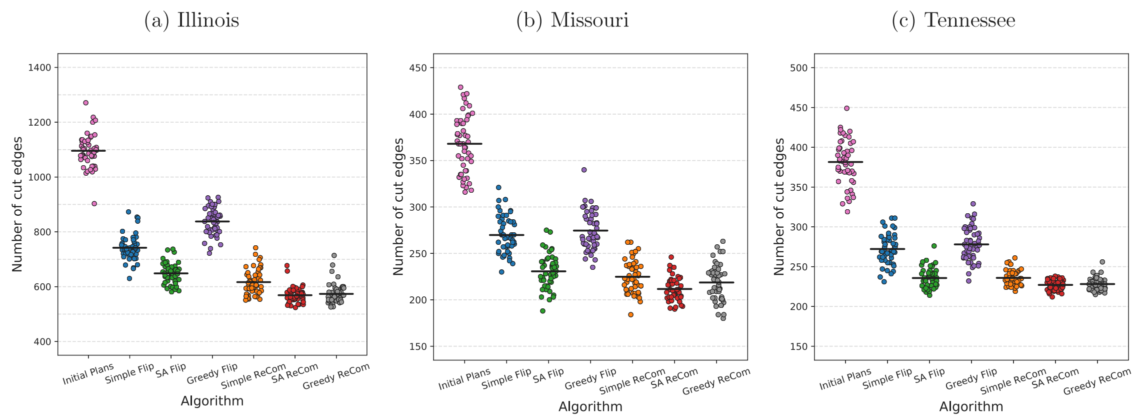


Fig. 8. Plots showing the compactness values for each algorithm's set of 50 plans optimized for compactness, for Illinois, Missouri, and Tennessee. The average value for each set is shown with a black line.

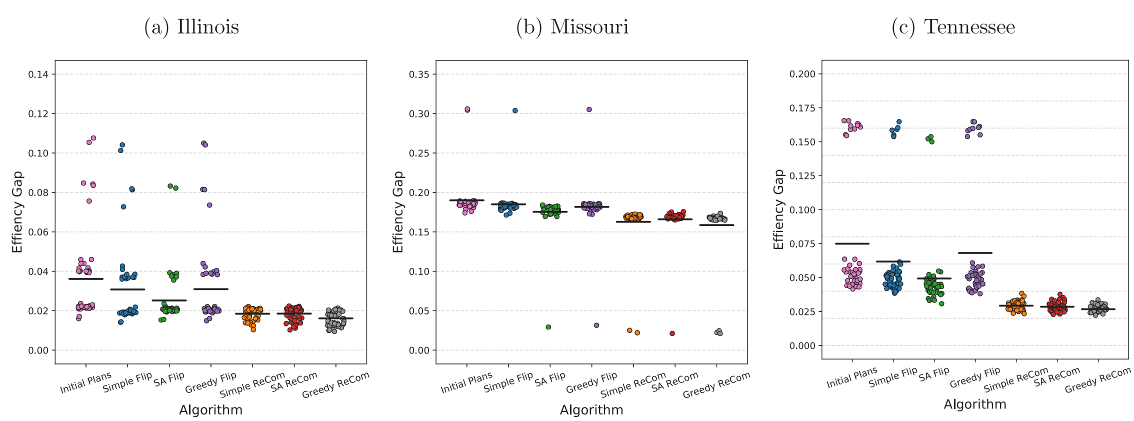


Fig. 9. Plots showing the efficiency gap values for each algorithm's set of 50 plans optimized for efficiency gap, for Illinois, Missouri, and Tennessee. The average value for each set is shown with a black line.

during the untimed experiments (Figs. 8–10), SA Flip achieved better average objective values than Simple Flip and Greedy Flip for compactness, efficiency gap, and mean-median across all three states. For competitiveness (Figs. 7 and 11), Simple Flip frequently produced values at least as good as SA Flip in both the timed and untimed experiments, likely because it was allowed to accept moves that maintained (rather than strictly improved) the objective. Since the number of competitive districts may not change even with several Flip iterations, this acceptance rule allowed for better exploration of the solution space.

For the ReCom local search algorithms, SA ReCom and Greedy ReCom tended to produce slightly better solutions than Simple ReCom. There are a few scenarios during the timed experiments in which Simple ReCom produced a better solution than SA ReCom and Greedy ReCom (e.g., compactness at 30 s for Tennessee, mean-median at 30 s for Illinois), likely because it tended to converge to an optimal solution

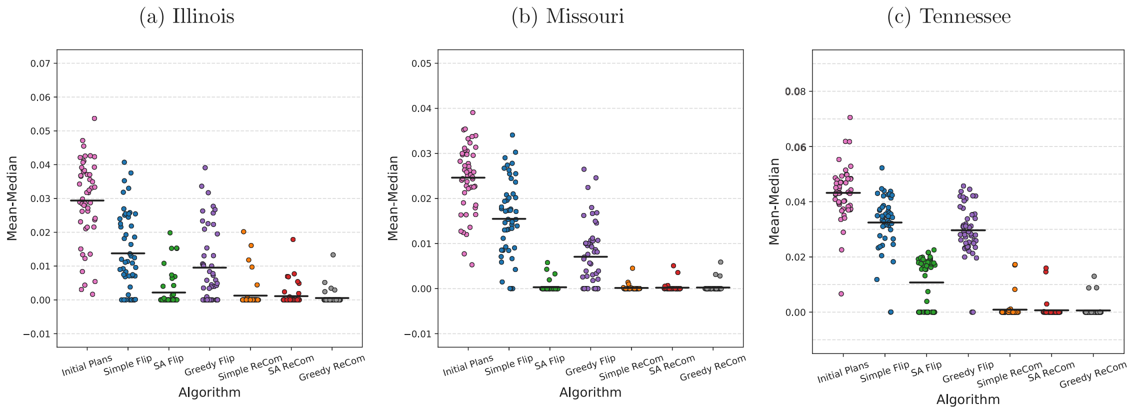


Fig. 10. Plots showing the mean-median values for each algorithm's set of 50 plans optimized for mean-median, for Illinois, Missouri, and Tennessee. The average value for each set is shown with a black line.

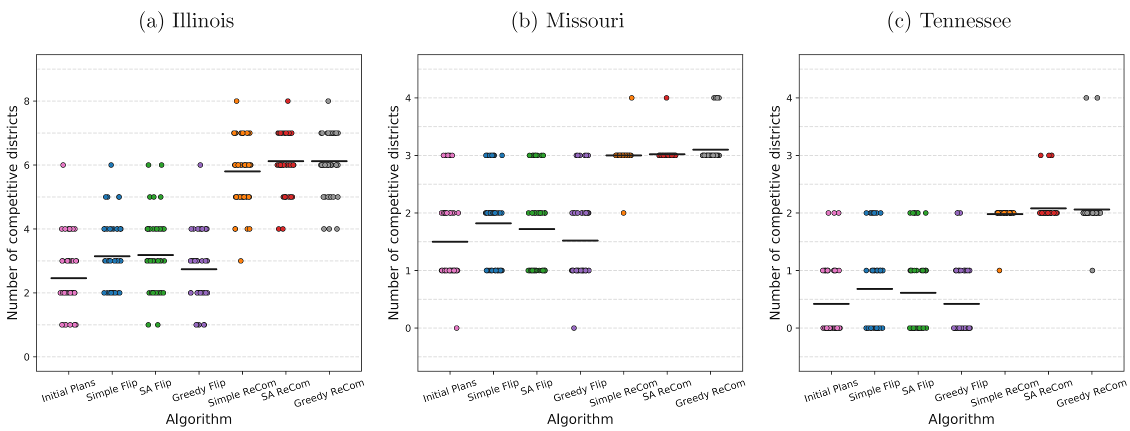


Fig. 11. Plots showing the competitiveness values for each algorithm's set of 50 plans optimized for competitiveness, for Illinois, Missouri, and Tennessee. The average value for each set is shown with a black line.

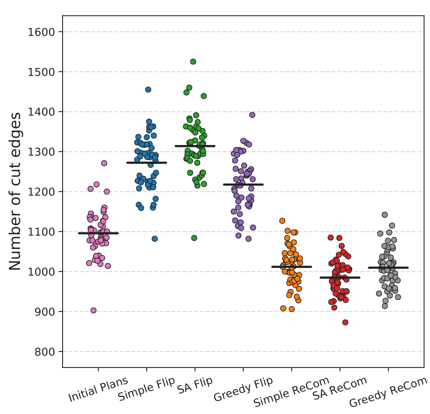


Fig. 12. Plots showing the compactness values for each algorithm's set of 50 Illinois plans optimized for mean-median. The average value for each set is shown with a black line.

more quickly (e.g., as demonstrated by the run times for the untimed experiments in Table 3). However, during the untimed experiments, SA ReCom and Greedy ReCom frequently achieved marginally better average objective values than Simple ReCom.

It is important to note that the typical advantages of simulated annealing and greedy algorithms (i.e., better exploration of the solution space and the ability to choose better improving moves, respectively) have not resulted in substantially better solutions over time in these ReCom experiments compared to simple hill-climbing. Simple ReCom, SA ReCom, and Greedy ReCom produced similar best values in the timed experiments after 300 s and similar average values in the untimed experiments. This similarity likely occurs because the ReCom iteration already allows for good exploration of the solution space and the ability to make substantial changes to a plan in a single iteration. Therefore, in contrast to Flip, these results suggest that the good objective values obtained by a ReCom local search method are more attributable to the ReCom iteration than the specific local search variant. Since Simple ReCom frequently converged to a local optimum faster on average than SA ReCom and Greedy ReCom (as shown in Table 3), it may be preferable to use Simple ReCom, especially if the goal is to generate several good plans.

Although Sample ReCom is a sampling heuristic and not an optimization heuristic, it occasionally produced the best solution for some time limits (e.g., competitiveness at 300 s for Missouri) or produced solutions similar to the ReCom local search algorithms (e.g., efficiency gap at all time limits for all three states) during the timed experiments. These results further suggest that ReCom's ability to explore the solution space well can lead to good objective values regardless of the algorithm variant (i.e., even without explicit optimization). Therefore, in similar scenarios, it may not be necessary to invoke an optimization heuristic to obtain good solutions. These results also provide additional support for the use of Sample ReCom as a random restart procedure (e.g., as in DeFord et al. (2020)).

Table A.4
The best compactness values each algorithm achieves in 30, 300, 1800, and 3600 s, for Illinois, Missouri, and Tennessee. The best value across all algorithms is shown in bold.

State	Algorithm	Best f_{comp} in 30 s	Best f_{comp} in 300 s	Best f_{comp} in 1800 s	Best f_{comp} in 3600 s
Illinois	Simple Flip	988	983	963	963
	SA Flip	869	776	776	776
	Greedy Flip	1290	1135	1122	1119
	Simple ReCom	**831**	**597**	547	547
	SA ReCom	1189	633	**540**	**530**
	Greedy ReCom	860	627	557	556
	Sample ReCom	887	744	733	723
Missouri	Simple Flip	276	272	270	270
	SA Flip	**205**	**196**	196	**190**
	Greedy Flip	279	276	274	274
	Simple ReCom	240	207	**195**	195
	SA ReCom	230	199	196	192
	Greedy ReCom	240	226	217	217
	Sample ReCom	292	269	259	240
Tennessee	Simple Flip	300	284	284	281
	SA Flip	**244**	**227**	227	226
	Greedy Flip	319	314	309	309
	Simple ReCom	265	236	225	223
	SA ReCom	277	246	**221**	**220**
	Greedy ReCom	282	231	226	223
	Sample ReCom	322	300	278	269

Table A.5
The best efficiency gap values each algorithm achieves in 30, 300, 1800, and 3600 s, for Illinois, Missouri, and Tennessee. The best value across all algorithms is shown in bold.

State	Algorithm	Best f_{eg} in 30 s	Best f_{eg} in 300 s	Best f_{eg} in 1800 s	Best f_{eg} in 3600 s
Illinois	Simple Flip	0.135	0.135	0.135	0.135
	SA Flip	0.082	0.082	0.082	0.081
	Greedy Flip	0.136	0.136	0.136	0.136
	Simple ReCom	0.016	0.005	0.005	0.004
	SA ReCom	0.021	0.010	0.005	0.004
	Greedy ReCom	**0.004**	**<0.001**	**<0.001**	**<0.001**
	Sample ReCom	0.014	0.014	0.014	0.014
Missouri	Simple Flip	0.184	0.183	0.183	0.183
	SA Flip	0.172	0.172	0.172	0.170
	Greedy Flip	0.184	0.184	0.184	0.184
	Simple ReCom	**0.171**	0.168	**0.018**	**0.018**
	SA ReCom	0.175	0.168	0.021	0.021
	Greedy ReCom	0.179	**0.167**	0.022	0.022
	Sample ReCom	0.177	0.172	0.031	0.031
Tennessee	Simple Flip	0.167	0.167	0.060	0.058
	SA Flip	0.153	0.052	0.039	0.038
	Greedy Flip	0.167	0.167	0.167	0.167
	Simple ReCom	0.037	0.027	0.025	**0.023**
	SA ReCom	0.038	**0.026**	0.025	**0.023**
	Greedy ReCom	**0.032**	**0.026**	**0.024**	0.024
	Sample ReCom	0.045	0.032	0.032	0.032

5.3. Comparison of states

The ReCom algorithms had fewer inconsistencies in performance across all three states than the Flip algorithms. During the timed experiments, Greedy ReCom produced compactness values similar to Simple ReCom and SA ReCom after 300 s for Illinois and Tennessee, but not for Missouri (Fig. 4); however, in the untimed experiments for Missouri, Greedy ReCom did produce compactness values similar to Simple ReCom and SA ReCom (Fig. 8). Therefore, this inconsistency is likely attributable to the initial Missouri plan for the timed experiments (Fig. 3(b)).

The Flip algorithms had inconsistencies in both the timed and untimed experiments. SA Flip achieved compactness values that were either the best or similar to the best in the timed experiments for Missouri and Tennessee (Fig. 4); however, SA Flip did not achieve the best compactness values for Illinois, likely because the district shapes in the initial Illinois plan (Fig. 3(a)) are so convoluted. The inconsistencies in efficiency gap values that Simple Flip and SA Flip achieved during the timed experiments (Fig. 5) also likely result from the initial plans; as the untimed experiments for efficiency gap show (Fig. 9), the Flip algorithms produced distributions of efficiency gap values quite similar to the initial plans' distribution. Lastly, while SA Flip achieved excellent average mean-median values in the untimed experiments for Illinois and Missouri, it did not in Tennessee (Fig. 10). Since these experiments used several initial plans, this inconsistency may be due to Tennessee's census geography.

It is also important to note that all algorithms failed to substantially improve the efficiency gap in Missouri within 300 s during the timed experiments (Fig. 5(b)); similarly, all algorithms failed to substantially improve the average efficiency gap during the untimed experiments (Fig. 9(b)). This phenomenon is likely due to Missouri's unique political geography. Population clustering, voter concentration, and voter location can affect the fairness values possible for a particular state and whether a local search algorithm is able to readily achieve these values. In Missouri, Democratic voters are concentrated in the Kansas City and St. Louis areas. Additionally, the areas around St. Louis with high concentrations of Democratic voters also have a large Black/African American population; hence, Missouri's one majority–minority district (as described in Section 3.2.2) is often packed with

Table A.6
The best mean-median values each algorithm achieves in 30, 300, 1800, and 3600 s, for Illinois, Missouri, and Tennessee. The best value across all algorithms is shown in bold.

State	Algorithm	Best f_{mm} in 30 s	Best f_{mm} in 300 s	Best f_{mm} in 1800 s	Best f_{mm} in 3600 s
Illinois	Simple Flip	0.0125	0.0082	0.0048	0.0048
	SA Flip	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy Flip	0.0072	0.0072	0.0072	0.0072
	Simple ReCom	0.0009	**<0.0001**	**<0.0001**	**<0.0001**
	SA ReCom	0.0139	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy ReCom	0.0185	**<0.0001**	**<0.0001**	**<0.0001**
	Sample ReCom	0.0201	0.0171	0.0093	0.0008
Missouri	Simple Flip	0.0133	**<0.0001**	**<0.0001**	**<0.0001**
	SA Flip	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy Flip	0.0181	0.0181	0.0181	0.0181
	Simple ReCom	0.0121	**<0.0001**	**<0.0001**	**<0.0001**
	SA ReCom	0.0002	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy ReCom	0.0102	**<0.0001**	**<0.0001**	**<0.0001**
	Sample ReCom	0.0094	0.0008	**<0.0001**	**<0.0001**
Tennessee	Simple Flip	0.0295	0.0276	0.0275	0.0274
	SA Flip	0.0188	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy Flip	0.0262	0.0262	0.0262	0.0262
	Simple ReCom	0.0226	**<0.0001**	**<0.0001**	**<0.0001**
	SA ReCom	0.0101	**<0.0001**	**<0.0001**	**<0.0001**
	Greedy ReCom	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
	Sample ReCom	0.0273	0.0024	**<0.0001**	**<0.0001**

Table A.7
The best competitiveness values each algorithm achieves in 30, 300, 1800, and 3600 s, for Illinois, Missouri, and Tennessee. The best value across all algorithms is shown in bold.

State	Algorithm	Best f_{cmpttv} in 30 s	Best f_{cmpttv} in 300 s	Best f_{cmpttv} in 1800 s	Best f_{cmpttv} in 3600 s
Illinois	Simple Flip	3	4	4	5
	SA Flip	3	4	5	5
	Greedy Flip	1	2	2	2
	Simple ReCom	4	**7**	7	7
	SA ReCom	5	5	**8**	**8**
	Greedy ReCom	**6**	**7**	7	**8**
	Sample ReCom	3	5	6	6
Missouri	Simple Flip	1	2	2	2
	SA Flip	1	2	2	2
	Greedy Flip	1	1	1	1
	Simple ReCom	**3**	**3**	**3**	3
	SA ReCom	**3**	**3**	**3**	3
	Greedy ReCom	2	**3**	**3**	**4**
	Sample ReCom	2	**3**	**3**	**4**
Tennessee	Simple Flip	1	1	2	2
	SA Flip	0	1	2	2
	Greedy Flip	0	0	0	0
	Simple ReCom	1	**4**	**4**	**4**
	SA ReCom	**2**	2	2	3
	Greedy ReCom	1	2	2	**4**
	Sample ReCom	**2**	2	2	2

Democratic voters. Therefore, the contiguity, compactness, and Voting Rights Act constraints reduce the number of feasible district plans for Missouri with proportional expected seat outcomes. See Dobbs et al. (2023) for more information on Missouri's political geography.

5.4. *Compactness of plans optimized for political fairness*

Here we examine how well the algorithms maintained compactness during the untimed experiments when the objective was a political fairness metric. As discussed in Section 3.2, Flip iterations can create fractal-like district shapes when the objective is not compactness, while ReCom iterations naturally favor compact districts. As an example, Fig. 12 shows the compactness values for the 50 Illinois plans optimized with respect to mean-median during the untimed experiments. The three Flip algorithms produced plans with worse average compactness than the initial plans; their compactness values are quite close to the allowed thresholds. In contrast, the ReCom algorithms produced plans with *better* average compactness than the initial plans. Tables B.9–B.11 in Appendix B report the average compactness values for each algorithm's set of plans optimized for efficiency gap, mean-median, and competitiveness, respectively, during the untimed experiments. For these experiments, the ReCom algorithms always produced plans with better average compactness than the Flip algorithms. Therefore, in the scenarios for which a Flip algorithm produced political fairness values close to the ReCom algorithms (e.g., SA Flip with a mean-median objective for Illinois and Missouri), the superior compactness values that the ReCom algorithms produced can be used to further distinguish the ReCom plans from the Flip plans. Since compact districts are desirable in practice, producing superior compactness values elevates the solution quality of the ReCom plans.

6. Conclusion

Optimization methods have the potential to increase transparency and reduce political bias in the political redistricting process. In this study, we evaluate the performance of several local search optimization heuristics for political redistricting that use ReCom or Flip iterations. We optimize congressional district plans for Illinois, Missouri, and Tennessee with respect to four common fairness objectives.

Table B.8
The average run time and the average, best, and worst compactness values for each set of 50 plans optimized for compactness.

State	Algorithm	Average run time (s)	Average f_{comp} value	Best f_{comp} value	Worst f_{comp} value
Illinois	Sample ReCom (Initial Plans)	8.08	1095.86	903	1271
	Simple Flip	6.97	742.02	630	873
	SA Flip	23.17	648.58	581	737
	Greedy Flip	157.84	837.86	722	926
	Simple ReCom	368.83	616.70	551	742
	SA ReCom	533.23	568.96	524	677
	Greedy ReCom	648.21	573.72	526	714
Missouri	Sample ReCom (Initial Plans)	7.13	368.04	316	429
	Simple Flip	2.77	269.78	230	321
	SA Flip	17.46	230.68	188	275
	Greedy Flip	38.49	274.56	235	340
	Simple ReCom	179.22	224.82	184	262
	SA ReCom	494.30	211.64	190	246
	Greedy ReCom	319.62	218.66	180	263
Tennessee	Sample ReCom (Initial Plans)	5.14	381.5	319	449
	Simple Flip	3.28	272.18	231	311
	SA Flip	18.39	235.70	214	276
	Greedy Flip	37.26	277.96	232	329
	Simple ReCom	178.39	235.88	219	261
	SA ReCom	419.10	227.06	212	238
	Greedy ReCom	291.75	228.12	215	256

Table B.9
The average run time, the average, best, and worst efficiency gap values, and the average compactness values for each set of 50 plans optimized for efficiency gap.

State	Algorithm	Average run time (s)	Average f_{eg} value	Best f_{eg} value	Worst f_{eg} value	Average f_{comp} value
Illinois	Sample ReCom (Initial Plans)	8.08	0.036	0.016	0.108	1095.86
	Simple Flip	2.06	0.031	0.014	0.104	1314.00
	SA Flip	3.29	0.025	0.015	0.083	1306.32
	Greedy Flip	26.84	0.031	0.015	0.105	1175.46
	Simple ReCom	81.71	0.018	0.010	0.022	937.56
	SA ReCom	89.17	0.018	0.010	0.022	944.54
	Greedy ReCom	221.85	0.016	0.009	0.021	928.50
Missouri	Sample ReCom (Initial Plans)	7.13	0.190	0.174	0.306	368.04
	Simple Flip	2.16	0.185	0.171	0.304	440.76
	SA Flip	6.74	0.175	0.029	0.184	440.16
	Greedy Flip	6.53	0.182	0.031	0.305	437.82
	Simple ReCom	126.57	0.163	0.022	0.173	363.12
	SA ReCom	126.74	0.166	0.021	0.176	358.56
	Greedy ReCom	230.39	0.159	0.021	0.173	355.24
Tennessee	Sample ReCom (Initial Plans)	5.14	0.075	0.042	0.166	381.5
	Simple Flip	2.54	0.062	0.038	0.165	456.12
	SA Flip	7.77	0.049	0.031	0.154	455.16
	Greedy Flip	6.71	0.068	0.038	0.165	450.36
	Simple ReCom	118.83	0.029	0.023	0.038	396.02
	SA ReCom	123.32	0.029	0.023	0.038	393.24
	Greedy ReCom	235.03	0.027	0.022	0.034	402.38

The results show that Simple ReCom, SA ReCom, and Greedy ReCom yielded excellent values for all four objectives, across all three redistricting instances. Although SA ReCom and Greedy ReCom tended to produce slightly better values than Simple ReCom, the difference was not substantial. SA Flip occasionally achieved similar or better values than the ReCom local search variants, depending on the objective and the redistricting instance. In contrast, Simple Flip and Greedy Flip frequently failed to substantially improve the objectives.

The ReCom local search variants are computationally slower than the Flip local search variants. Therefore, for some objectives (e.g., compactness, mean-median), SA Flip produced a better solution than the ReCom algorithms within short time periods. For other objectives (e.g., efficiency gap, competitiveness), the ReCom algorithms achieved better solutions, even within short time periods.

Hence, the most useful local search algorithm for redistricting depends on one's goals. If the goal is to ascertain objective value bounds for a particular redistricting instance (e.g., by using local search as a heuristic warm-start for exact optimization), then SA ReCom or Greedy ReCom would be useful since they tended to converge to the best solutions (regardless of objective or instance). If time is limited, SA Flip can occasionally provide a better solution than the ReCom algorithms within a short time period for some objectives (e.g., compactness, mean-median). If the goal is to generate a diverse collection of plans with good objective values in a reasonable amount of time (e.g., for a state redistricting committee), Simple ReCom offers a balance between solution quality and run time; similarly, a ReCom-Flip hybrid method (e.g., SA Flip with a Sample ReCom random restart procedure) could quickly generate good solutions.

In general, the use of ReCom iterations within an optimization framework remains underexplored. While this study examines how ReCom iterations perform in different local search variants and compare to Flip iterations within a heuristic optimization framework, it would also be useful to compare a ReCom local search method to redistricting heuristics other than local search (e.g., Gurnee and Shmoys (2021) and Swamy et al. (2022)). It is also possible that other aspects of a ReCom iteration, such as the selection of a candidate bipartition or the bipartition method itself, could be examined and specifically tailored to one or more redistricting objectives. For example, Clelland et al. (2022) adjust the randomized edge weights for Kruskal's algorithm to favor spanning trees with more intra-county edges in ReCom iterations; it may be possible to similarly adjust edge weights to make each candidate bipartition more likely to improve a chosen objective, thus

Table B.10
The average run time, the average, best, and worst mean-median values, and the average compactness values for each set of 50 plans optimized for mean-median.

State	Algorithm	Average run time (s)	Average f_{mm} value	Best f_{mm} value	Worst f_{mm} value	Average f_{comp} value
Illinois	Sample ReCom (Initial Plans)	8.08	0.0294	0.0017	0.0537	1095.86
	Simple Flip	1.94	0.0138	<0.0001	0.0407	1272.22
	SA Flip	16.43	0.0022	<0.0001	0.0198	1313.84
	Greedy Flip	28.22	0.0100	<0.0001	0.0391	1217.58
	Simple ReCom	223.05	0.0013	<0.0001	0.0202	1011.84
	SA ReCom	219.15	0.0011	<0.0001	0.0179	984.86
	Greedy ReCom	370.40	0.0005	<0.0001	0.0133	1009.68
Missouri	Sample ReCom (Initial Plans)	7.13	0.0246	0.0053	0.0391	368.04
	Simple Flip	2.12	0.0155	<0.0001	0.0341	441.06
	SA Flip	18.33	0.0003	<0.0001	0.0058	439.82
	Greedy Flip	7.69	0.0071	<0.0001	0.0265	440.52
	Simple ReCom	142.17	0.0001	<0.0001	0.0045	395.02
	SA ReCom	148.62	0.0002	<0.0001	0.0051	384.64
	Greedy ReCom	246.06	0.0002	<0.0001	0.0059	388.24
Tennessee	Sample ReCom (Initial Plans)	5.14	0.0432	0.0066	0.0705	381.5
	Simple Flip	1.74	0.0325	<0.0001	0.0523	444.16
	SA Flip	20.21	0.0107	<0.0001	0.0225	456.36
	Greedy Flip	4.72	0.0297	<0.0001	0.0457	431.80
	Simple ReCom	114.75	0.0009	<0.0001	0.0173	411.20
	SA ReCom	113.16	0.0007	<0.0001	0.0160	409.98
	Greedy ReCom	216.49	0.0006	<0.0001	0.0130	413.42

Table B.11
The average run time, the average, best, and worst competitiveness values, and the average compactness values for each set of 50 plans optimized for competitiveness.

State	Algorithm	Average run time (s)	Average f_{cmpttv} Value	Best f_{cmpttv} Value	Worst f_{cmpttv} Value	Average f_{comp} Value
Illinois	Sample ReCom (Initial Plans)	8.08	2.46	6	1	1095.86
	Simple Flip	5.09	3.15	6	2	1314.58
	SA Flip	28.26	3.18	6	1	1314.35
	Greedy Flip	94.65	2.74	6	1	1189.12
	Simple ReCom	108.98	5.8	8	3	891.82
	SA ReCom	195.45	6.12	8	4	883.50
	Greedy ReCom	215.34	6.12	8	4	882.14
Missouri	Sample ReCom (Initial Plans)	7.13	1.50	3	0	368.04
	Simple Flip	2.89	1.82	3	1	440.90
	SA Flip	9.33	1.72	3	1	440.70
	Greedy Flip	22.15	1.52	3	0	436.68
	Simple ReCom	77.06	3.00	4	2	353.72
	SA ReCom	132.58	3.02	4	3	359.60
	Greedy ReCom	162.18	3.1	4	3	359.56
Tennessee	Sample ReCom (Initial Plans)	5.14	0.42	2	0	381.5
	Simple Flip	3.02	0.68	2	0	456.58
	SA Flip	8.55	0.61	2	0	456.65
	Greedy Flip	21.13	0.42	2	0	451.94
	Simple ReCom	67.12	1.98	2	1	380.24
	SA ReCom	126.13	2.08	3	2	377.68
	Greedy ReCom	130.21	2.06	4	1	377.44

reducing the number of ReCom iterations local search takes to converge to a local optimum. Another logical extension for any redistricting optimization method is to adapt it for multi-criteria optimization to better incorporate multiple competing stakeholder preferences.

CRediT authorship contribution statement

Kiera W. Dobbs: Conceptualization, Methodology, Software, Data curation, Investigation, Visualization, Writing – original draft, Writing – review & editing. **Douglas M. King:** Supervision, Writing – review & editing. **Sheldon H. Jacobson:** Supervision, Writing – review & editing.

Data availability

I have shared the link to the code/data in the main manuscript text.

Appendix A. Tables for timed experiments

See Tables A.4–A.7.

Appendix B. Tables for untimed experiments

See Tables B.8–B.11.

References

Abramowitz, A.I., Alexander, B., Gunning, M., 2006. Incumbency, redistricting, and the decline of competition in U.S. House elections. J. Politics 68 (1), 75–88.
Ahuja, R.K., Orlin, J.B., Sharma, D., 2000. Very large-scale neighborhood search. Int. Trans. Oper. Res. 7 (4-5), 301–317.
Altman, M., McDonald, M., 2010. The promise and perils of computers in redistricting. Duke J. Const. Law Public Policy 5 (69), 69–111.
2000. Ariz. const. art. iv, pt. 2, §3.
Arizona Independent Redistricting Commission, 2021. Grid map. URL https://redistricting-irc-az.hub.arcgis.com/pages/grid-map. (Date accessed: 2022-03-10).
Arizona State Legislature v. Arizona Independent Redistricting Commission, 2015. 576 U.S. 787.
Arredondo, V., Martínez-Panero, M., Peña, T., Ricca, F., 2021. Mathematical political districting taking care of minority groups. Ann. Oper. Res. 305 (1-2), 375–402.
Ballotpedia, 2017. Majority-minority districts. URL https://ballotpedia.org/Majority-minority_districts, (Date accessed: 2022-01-26).
Bernstein, M., Duchin, M., 2017. A formula goes to court: Partisan gerrymandering and the efficiency gap. Notices Amer. Math. Soc. 64 (9), 1020–1024.
Bichot, C.-E., Siarry, P. (Eds.), 2011. Graph Partitioning. ISTE Ltd., London, UK.

Bozkaya, B., Erkut, E., Haight, D., Laporte, G., 2011. Designing new electoral districts for the city of Edmonton. Interfaces 41 (6), 534–547.
Cain, B.E., Cho, W.K.T., Liu, Y.Y., Zhang, E.R., 2018. A reasonable bias approach to gerrymandering: Using automated plan generation to evaluate redistricting proposals. William Mary Law Rev. 59 (5), 1521–1557.
Chambers, C., Miller, A., Sobel, J., 2017. Flaws in the efficiency gap. J. Law Politics 33 (1), 1–33.
Chen, J., Rodden, J., 2013. Unintentional gerrymandering: Political geography and electoral bias in legislatures. Q. J. Political Sci. 8 (3), 239–269.
Chikina, M., Frieze, A., Pegden, W., 2017. Assessing significance in a Markov chain without mixing. Proc. Natl. Acad. Sci. 114 (11), 2860–2864.
Cho, W.K.T., 2017. Measuring partisan fairness: How well does the efficiency gap guard against sophisticated as well as simple-minded modes of partisan discrimination? Univ. Pa. Law Rev. Online 166 (17), 17–36.
Cho, W.K.T., Liu, Y.Y., 2021. A parallel evolutionary multiple-try Metropolis Markov chain Monte Carlo algorithm for sampling spatial partitions. Stat. Comput. 31 (10), 2860–2864.
Clelland, J., Colgate, H., DeFord, D., Malmskog, B., Sancier-Barbosa, F., 2022. Colorado in context: Congressional redistricting and competing fairness criteria in Colorado. J. Comput. Soc. Sci. 5 (1), 189–226.
CNN, 2022. Redistricting in Tennessee. URL https://www.cnn.com/interactive/2022/politics/us-redistricting/tennessee-redistricting-map/, (Date accessed: 2023-05-09).
2018. Colo. const. art. v, §44.
Davis v. Bandemer, 1986. 478 U.S. 109.
DeFord, D., Dhamankar, N., Duchin, M., Gupta, V., McPike, M., Schoenbach, G., Sim, K.W., 2023. Implementing partisan symmetry: Problems and paradoxes. Political Anal. 31 (3), 1–20.
DeFord, D., Duchin, M., Solomon, J., 2020. A computational approach to measuring vote elasticity and competitiveness. Stat. Public Policy 7 (1), 69–86.
DeFord, D., Duchin, M., Solomon, J., 2021. Recombination: A family of Markov chains for redistricting. Harvard Data Sci. Rev. 3 (1).
Dobbs, K.W., Swamy, R., King, D.M., Ludden, I.G., Jacobson, S.H., 2023. An optimization case study in analyzing Missouri redistricting. INFORMS J. Appl. Anal. to appear.
Duchin, M., Tenner, B.E., 2018. Discrete geometry for electoral geography. arXiv preprint, 1–18, arXiv:1808.05860v1.
Fifield, B., Higgins, M., Imai, K., Tarr, A., 2020. Automated redistricting simulation using Markov Chain Monte Carlo. J. Comput. Graph. Statist. 29 (4), 715–728.
Garfinkel, R., Nemhauser, G., 1970. Optimal political districting by implicit enumeration techniques. Manage. Sci. 16 (8), 495–508.
Gill v. Whitford, 2018. 585 U.S..
Griffith, E.C., 1907. The Rise and Development of the Gerrymander. Scott, Foresman and Co., Chicago.
Grofman, B., 1983. Measures of bias and proportionality in seats-votes relationships. Political Methodol. 9 (3), 295–327.
Grofman, B., King, G., 2007. The future of partisan symmetry as a judicial test for partisan gerrymandering after LULAC v. Perry. Elect. Law J. 6 (1), 2–35.
Gurnee, W., Shmoys, D.B., 2021. Fairmandering: A column generation heuristic for fairness-optimized political districting. Proceedings of the 2021 SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21), pp. 88–99, URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976830.9.
Herschlag, G., Kang, H., Luo, J., Graves, C.V., Bangia, S., Ravier, R., Mattingly, J., 2020. Quantifying gerrymandering in North Carolina. Stat. Public Policy 7 (1), 30–38.
Hess, S., Weaver, J., Siegfeldt, H., Whelan, J., Zitlau, P., 1965. Nonpartisan political redistricting by computer. Oper. Res. 13 (6), 998–1006.
Illinois House Democrats, 2021. Illinois redistricting. URL https://ilhousedems.com/redistricting/, (Date accessed: 2021-10-29).
Katz, J.N., King, G., Rosenblatt, E., 2020. Theoretical foundations and empirical evaluations of partisan fairness in district-based democracies. Am. Political Sci. Rev. 114 (1), 164–178.
King, D.M., Jacobson, S.H., Sewell, E.C., 2018. The geo-graph in practice: Creating United States congressional districts from census blocks. Comput. Optim. Appl. 69 (1), 25–49.
King, D.M., Jacobson, S.H., Sewell, E.C., Cho, W.K.T., 2012. Geo-graphs: An efficient model for enforcing contiguity and hole constraints in planar graph partitioning. Oper. Res. 60 (5), 1213–1228.
Kirkpatrick, S., Gelatt Jr., C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.
League of United Latin American Citizens v. Perry, 2006. 548 U.S. 399.
Liu, Y.Y., Cho, W.K.T., Wang, S., 2016. PEAR: A massively parallel evolutionary computation approach for political redistricting optimization and analysis. Swarm Evol. Comput. 30, 78–92.
McCarty, N., Poole, K.T., Rosenthal, H., 2009. Does gerrymandering cause polarization? Am. J. Political Sci. 53 (3), 666–680.
McDonald, M.D., Best, R.E., 2015. Unfair partisan gerrymanders in politics and law: A diagnostic applied to six cases. Elect. Law J. 14 (4), 312–330.
McGhee, E., 2014. Partisan bias in single-member district electoral systems. Legislative Stud. Q. 39 (1), 55–85.
Missouri Secretary of State, 2020. 2020 Ballot measures. URL https://www.sos.mo.gov/elections/petitions/2020BallotMeasures, (Date accessed: 2021-09-01).
2020. Mo. const. art. iii, §3.
1945. Mo. const. art. iii, §45.
National Conference of State Legislatures, 2021a. Creation of redistricting commissions. URL https://www.ncsl.org/redistricting-and-census/creation-of-redistricting-commissions.
National Conference of State Legislatures, 2021b. Redistricting criteria. URL https://www.ncsl.org/research/redistricting/redistricting-criteria.aspx, (Date accessed: 2021-09-01).
National Conference of State Legislatures, 2021c. Redistricting systems: A 50-state overview. URL https://www.ncsl.org/research/redistricting/redistricting-systems-a-50-state-overview.aspx#, (Date accessed: 2021-09-01).
2014. N.y. const. art. iii, §4.
Oehrlein, J., Haunert, J.-H., 2017. A cutting-plane method for contiguity-constrained spatial aggregation. J. Spatial Inf. Sci. 15, 89–120.
2015. Ohio. const. art. xi, §6.
Reynolds v. Sims, 1964. 377 U.S. 533.
Ricca, F., Scozzari, A., 2020. Mathematical programming formulations for practical political districting. In: Ríos-Mercado, R.Z. (Ed.), Optimal Districting and Territory Design. Springer, Cham, Switzerland, pp. 105–128.
Ricca, F., Scozzari, A., Simeone, B., 2013. Political districting: From classical models to recent approaches. Ann. Oper. Res. 204 (1), 271–299.
Ricca, F., Simeone, B., 2008. Local search algorithms for political districting. European J. Oper. Res. 189 (3), 1409–1426.
Ríos-Mercado, R.Z. (Ed.), 2020. Optimal Districting and Territory Design. Springer, Cham, Switzerland.
Shirabe, T., 2009. Districting modeling with exact contiguity constraints. Environ. Plan. B: Plann. Des. 36 (6), 1053–1066.
State of Tennessee, 2023. State of Tennessee downloadable GIS data. URL https://tn-tnmap.opendata.arcgis.com/datasets/90e4742978674ef4aaf08eb9f9f845bb/explore, (Date accessed: 2023-05-09).
Stephanopoulos, N., McGhee, E., 2015. Partisan gerrymandering and the efficiency gap. Univ. Chicago Law Rev. 82 (2), 831–900.
Swamy, R., King, D.M., Jacobson, S.H., 2022. Multi-objective optimization for politically fair districting: A scalable multilevel approach. Oper. Res. 71 (2), 536–562.
Thornburg v. Gingles, 1986. 478 U.S. 30.
U.S. Census Bureau, 2020a. 2020 Census: Apportionment of the U.S. House of Representatives. URL https://www.census.gov/library/visualizations/2021/dec/2020-apportionment-map.html, (Date accessed: 2022-05-27).
U.S. Census Bureau, 2020b. Hispanic or Latino, and not Hispanic or Latino by race. URL https://data.census.gov/cedsci/table{q}=&g=0400000US17%24100000&y=2020&tid=DECENNIALPL2020.P2, (Date accessed: 2021-09-16).
U.S. Census Bureau, 2020c. Hispanic or Latino, and not Hispanic or Latino by race. URL https://data.census.gov/cedsci/table{q}=&g=0400000US29%24100000&y=2020&tid=DECENNIALPL2020.P2, (Date accessed: 2021-09-16).
U.S. Census Bureau, 2020d. Hispanic or Latino, and not Hispanic or Latino by race. URL https://data.census.gov/table{t}=Hispanic+or+Latino&g=040XX00US47$1000000&y=2020, (Date accessed: 2022-06-28).
U.S. Census Bureau, 2020e. Race. URL https://data.census.gov/cedsci/table{q}=&g=0400000US17%24100000&y=2020&tid=DECENNIALPL2020.P1, (Date accessed: 2021-09-16).
U.S. Census Bureau, 2020f. Race. URL https://data.census.gov/cedsci/table{q}=&g=0400000US29%24100000&y=2020&tid=DECENNIALPL2020.P1, (Date accessed: 2021-09-16).
U.S. Census Bureau, 2020g. Race. URL https://data.census.gov/table{g}=040XX00US47$1000000&y=2020, (Date accessed: 2022-06-28).
U.S. Census Bureau, 2020h. TIGER/Line shapefiles. URL https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.2020.html, (Date accessed: 2021-09-16).
Validi, H., Buchanan, A., 2022. Political districting to minimize cut edges. Math. Program. Comput. 14 (4), 623–672.
Validi, H., Buchanan, A., Lykhovyd, E., 2021. Imposing contiguity constraints in political districting models. Oper. Res. 70 (2), 867–892.
Vieth v. Jubelirer, 2004. 541 U.S. 267.
Voting and Election Science Team, 2021. Voting and Election Science Team. URL https://dataverse.harvard.edu/dataverse/electionscience/?q=illinois, (Date accessed: 2021-10-07).
Voting Rights Data Institute, 2018. GerryChain. URL https://github.com/mggg/GerryChain, (Date accessed: 2022-02-01).
Warrington, G., 2018. Quantifying gerrymandering using the vote distribution. Elect. Law J. 17 (1), 39–57.
1983. Wash. const. art. ii, §43.
Wesberry v. Sanders, 1964. 376 U.S. 1.
Young, H., 1988. Measuring the compactness of legislative districts. Legislative Stud. Q. 13 (1), 105–115.