Comput Optim Appl (2018) 69:25–49
https://doi.org/10.1007/s10589-017-9936-3

The geo-graph in practice: creating United States Congressional Districts from census blocks

D. M. King[1] · S. H. Jacobson[2] · E. C. Sewell[3]

Received: 4 May 2016 / Published online: 22 August 2017

Abstract Every 10 years, United States Congressional Districts must be redesigned in response to a national census. While the size of practical political districting problems is typically too large for exact optimization approaches, heuristics such as local search can help stakeholders quickly identify good (but suboptimal) plans that suit their objectives. However, enforcing a district contiguity constraint during local search can require significant computation; tools that can reduce contiguity-based computations in large practical districting problems are needed. This paper applies the *geo-graph* framework to the creation of United States Congressional Districts from census blocks in four states—Arizona, Massachusetts, New Mexico, and New York—and finds that (a) geo-graph contiguity assessment algorithms reduce the average number of edges visited during contiguity assessments by at least three orders of magnitude in every

This research was supported in part by the National Science Foundation [IIS-0827540]. The second author was supported in part by the Air Force Office of Scientific Research [FA9550-10-1-0387, FA9550-15-1-0100].

✉ D. M. King
dmking@illinois.edu

S. H. Jacobson
shj@illinois.edu

E. C. Sewell
esewell@siue.edu

1 Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, 117 Transportation Building, 104 S. Mathews Ave., MC-238, Urbana, IL 61801, USA

2 Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

3 Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL, USA

problem instance when compared with simple graph search, and (b) the assumptions of the geo-graph model are easily adapted to the sometimes-irregular census block geography with only superficial impact on the solution space. These results show that the geo-graph model and its associated contiguity algorithms provide a powerful constraint assessment tool to political districting stakeholders.

Keywords Planar graphs · Graph partitioning · Geographic districting · Graph connectivity

1 Introduction

Political districting is an important practical problem with significant impact on political representation; an important example of the political districting process involves the creation of United States Congressional Districts, which occurs every 10 years after a national census [4]. The districting process is highly contentious, with many individuals and public interest groups seeking to have their voices heard in the redistricting process; hence, it is critical for stakeholders to be able to quickly develop and propose *district plans* (i.e., divisions of a state into Congressional Districts) that reflect their goals. District plans are constrained by relevant laws, with the most common of these requiring each political district to comprise a contiguous area (i.e., a *district contiguity* constraint) and all districts to be approximately equipopulous (i.e., a *population balance* constraint) [6]. Districting problems are naturally discrete, as each district is composed of a set of smaller areas called *units*. While units can be chosen at a variety of granularities, census blocks represent the smallest land areas within a state for which relevant data (e.g., population counts) are known, and hence, provide the highest degree of flexibility when constructing United States Congressional districts.

While the process of developing a district plan can be formulated as a discrete optimization problem, the resulting problems are typically NP-hard [1], and hence, a formulation that includes both a large number of census blocks and a computationally-challenging district contiguity constraint makes exact algorithms intractable in practice. For example, the districting problems encountered after the 2010 census required the 350,169 census blocks in New York to be divided among 27 districts, and the 710,145 census blocks in California to be allocated among 53 districts [3,27]. Though one can undertake these problems at a more coarse level of detail by defining units in a more coarse way (e.g., block groups, census tracts) or combining units in an intelligent way before optimization methods are applied (e.g., [14]), doing so eliminates many feasible district plans from the solution space. Tools that can efficiently analyze more finely-grained units are critical for optimization approaches that seek to consider these units in practice.

A balance can be struck between solution quality and algorithm tractability by using heuristics, which explore the solution space and typically report a high quality (though suboptimal) solution in far less time than exact algorithms. Heuristics such as local search have been widely used in political districting and other geographic districting problems (e.g., [2,5,17,20]). A common local search approach creates an initial district plan, and moves one unit to a new district in each iteration. To ensure that the new district plan remains feasible, local search must determine that removing this unit from its previous district does not violate the district contiguity constraint. One method for implementing such contiguity assessments applies a simple graph search on this district (e.g., [2,5,17,20]); however, a simple breadth-first search or depth-first search approach may require substantial computation when the district contains many units. The *geo-graph* modeling framework has been proposed to reduce the computational overhead of simple graph search by exploiting the planar geographic nature of the units [12,13]; preliminary computational studies applying the geo-graph model to census blocks in the state of Kansas reduced the number of edges visited and computation time expended during contiguity assessments by three orders of magnitude compared to simple breadth-first search and depth-first search methods [13].

This paper further integrates the geo-graph into practical districting scenarios by (a) conducting comprehensive computational studies of full runs of local search on practical United States Congressional Districting problems in the states of Arizona, Massachusetts, New Mexico, and New York to further demonstrate the effectiveness of geo-graph contiguity methods in reducing computation in practice, and (b) proposing strategies for adapting irregular census block geometries (i.e., those that contain holes, or are composed of multiple disconnected pieces) into simple shapes that can be considered in the geo-graph framework. This paper is organized as follows. Section 2 discusses existing approaches to political districting, with an emphasis on local search methods, as well as the structure of the geo-graph model and its two contiguity assessment algorithms, termed the *basic* and *efficient* geo-graph contiguity algorithms. Section 3 discusses strategies for adapting the geo-graph framework to census blocks whose boundaries are not simple closed curves, while Sect. 4 discusses the particular problem instances encountered when creating United States Congressional Districts in Arizona, Massachusetts, New Mexico, and New York. Section 5 details the experimental procedure used to create districts in this paper, and Sect. 6 summarizes the districts generated by this procedure and discusses the computational savings provided by the geo-graph. Section 7 draws conclusions on the outcomes of these experiments.

2 Background

Political districting is an intractable discrete optimization problem. While the structure and objectives of a political districting problem vary according to the goals of the designer, Altman [1] shows that one of the most basic districting problems, in which one must identify contiguous districts whose maximum population difference is less than a given positive real number, is NP-hard. While this fact implies that exact methods for redistricting are unlikely to be fruitful when applied to large problem instances, such problems can still be treated as binary integer programs, though enforcing contiguity requires a number of inequality constraints that grows exponentially with the number of units [7]; a fluid flow model for contiguity avoids this exponential growth by adding continuous variables, but remains intractable when the number of units is large [22,23]. Heuristic methods have been adopted to cope with the inherent intractability of exact solution methods. Kalcsics et al. [9] discuss factors that influence several types of geographic districting problems (e.g., political districting, school districting, service districting) and review several heuristics methods that can be applied to districting problems, with particular attention to location-allocation algorithms. Ricca et al. [19] focus on political districting problems, particularly to recent approaches using local search and computational geometry.

Local search is a discrete optimization metaheuristic that allows its user to define many facets of the optimization process; for example, many methods have been proposed and studied for selecting a neighbor in each iteration (e.g., steepest descent, simulated annealing, tabu search). Moreover, the user can tailor the structure of the neighborhood to suit the problem being investigated. In political districting problems, each feasible solution (i.e., district plan) corresponds to a particular partitioning of the units into contiguous, equipopulous districts. The neighborhood of this solution typically consists of feasible district plans that can be generated by transferring one unit from its current district into a different district (e.g., [2,5,17,20]). However, some studies consider variations of this strategy; Bozkaya et al. [2] consider a neighborhood structure that includes two-unit trades (i.e., where two districts are chosen, each of which gains a unit from the other) in addition to one-unit transfers, while Ricca et al. [18] propose approaches for transferring multiple units in each iteration. Yamada [29] stores the current district plan as a spanning forest with each tree corresponding to a district; when a vertex (i.e., unit) is transferred to a new district, all of its descendants are transferred as well.

When political districting creates contiguous districts from a predefined set of discrete units, the districting process is equivalent to a *graph partitioning* problem on a graph, $G = (V, E)$, whose vertices correspond to the units and whose edges join vertices whose units share a common boundary segment. A graph partition divides the vertex set into subsets denoted by $(V_1, V_2, \ldots, V_k)$, where $V = \bigcup_{j=1}^{k} V_j$ and $V_{j_1} \cap V_{j_2} = \emptyset$ for all $1 \leq j_1 < j_2 \leq k$, and the vertices in subset V_j correspond to the units in the j-th of k total districts. This formulation allows straightforward integration of district contiguity and population balance constraints. If each vertex $v \in V$ has population $pop(v)$, then the population of district j is $DistPop(j) = \sum_{v \in V_j} pop(v)$; population balance requires that these district populations fall within an acceptable range of values. District j is contiguous if and only if V_j induces a connected subgraph in G. While district contiguity and population balance constraints are pervasive in political districting problems, designers may wish to include other factors, such as population demographics, voting patterns, conformity to administrative boundaries, district compactness, and integrity of communities of interest [4,6,9].

In their concluding section, Ricca et al. [19] discuss the difficulties that arise from adding strict contiguity constraints to a model. When local search removes a single unit from district j, it must ensure that removing this unit does not cause district j to be split into multiple pieces. From a graph theory perspective, this requirement is equivalent to requiring that the vertex corresponding to that unit is not a *cut-vertex* of the subgraph induced by V_j. The cut-vertices of this subgraph can be assessed in many ways; two common search-based approaches either (a) enumerate all cut-vertices of district j in $O(|V_j|)$ time [8,24], or (b) determine whether a single vertex of district j is a cut-vertex in $O(|V_j|)$ time [20]. While the first of these approaches immediately seems more appealing than the second, since it locates all cut-vertices in the district, rather than just one, the second approach may be able to terminate very quickly if the chosen vertex is not a cut-vertex. However, both algorithms exhibit the same time complexity, and may need to visit every vertex in the district.

The *geo-graph* modeling framework exploits knowledge about the arrangement of units in the plane to reduce the amount of computation required to assess district contiguity during local search [13]. A geo-graph, $G = (V, E, B, z)$, augments the set of vertices and edges in the graph with a boundary function, B, that associates each unit $v \in V$ with the simple closed curve, $B(v)$, that draws the boundary of that unit in the plane. The zoning function, z, associates each unit $v \in V$ with its zone (i.e., district), $z(v)$; the number of zones is $m(G)$ with the related set of zone labels $M(G) = \{1, 2, \ldots, m(G)\}$, and hence, $z(v) \in M(G)$ for every $v \in V$. By this definition, the set of units in zone $j \in M(G)$ is given by $V(j) \equiv \{v \in V : z(v) = j\}$. This notation is slightly extended to include the dummy vertex v_0 that corresponds to the single infinite area of the plane outside the units, such that $V_0 \equiv V \cup \{v_0\}$ and $M_0(G) \equiv M(G) \cup \{0\}$, where zone 0 is a dummy zone that always contains only v_0 in any geo-graph [i.e., $z(v_0) \equiv 0$]. If $J \subseteq M_0(G)$ is a subset of the set of zones, then this definition is generalized such that $V(J) \equiv \{v \in V_0 : z(v) \in J\}$ is the set of units among all of the zones in J. A geo-graph is called *zone-connected* if for each $j \in M(G)$, the subgraph induced by $V(j)$ is connected, and hence, a district plan satisfies the district contiguity constraint if and only if its associated geo-graph is zone-connected. In districting problems involving a geo-graph, local search visits a sequence of geo-graphs that are identical except for their zoning functions, with the zoning functions of adjacent geo-graphs in this sequence differing only in the zone assignment of one vertex (i.e., the unit being transferred). To ensure feasibility, local search should be limited to visiting zone-connected geo-graphs that satisfy other relevant constraints (e.g., population balance).

Incorporating knowledge of the unit boundaries allows the geo-graph to limit the number of vertices that must be visited while assessing contiguity with graph search. In particular, these searches can be restricted to the set of vertices whose associated unit boundaries share at least a single point with the boundary of the transferred unit. For each vertex, this set of vertices is called its *augmented neighborhood*, defined as $R(v) \equiv \{x \in V : B(x) \cap B(v) \neq \emptyset\}$, while the standard graph neighborhood contains vertices whose associated units share a common boundary segment of non-zero length and is defined in the usual way, $N(v) \equiv \{x \in V : xv \in E\}$. Moreover, $N_j(v) \equiv N(v) \cap V(j)$ is the set of neighbors of v that reside in zone j. Theorem 1 [13] shows how to assess the connectivity of zone $z(v)$ after vertex v is removed from it by executing as many as $m(G)$ searches on the subgraph induced by a subset of the vertices in $R(v)$.

Theorem 1 (King et al. [13]) *Let $G = (V, E, B, z)$ be a zone-connected geo-graph with $v \in V$. The subgraph induced by $V(z(v)) - v$ is connected if and only if $N_{z(v)}(v)$ is contained in a single component of the subgraph induced by $R(v) \cap V(\Pi(z(v)) \cup \{z(v)\})$, and $N_{z(v)}(v)$ is contained in a single component of the subgraph induced by $R(v) \cap V(M_0(G) - \Pi_j(z(v)))$ for every $j \in \{1, 2, \ldots, \pi(z(v))\}$.*

The number of these searches and the set of vertices participating in each search are determined by the *pocket set* of zone $z(v)$, which defines the set of zones that reside in *holes* of zone $z(v)$. In particular, the set of zones $\Pi(z(v)) \subseteq M(G)$ is the set of all zones contained in holes of zone $z(v)$, and $\Pi(z(v))$ is divided into mutually exclusive and exhaustive subsets $\Pi_1(z(v)), \Pi_2(z(v)) \ldots, \Pi_{\pi(z(v))}(z(v))$, each containing the zones that reside in a single hole. This decomposition of $\Pi(z(v))$, called the *pocket set* of zone $z(v)$, can be evaluated in $O(m(G)^2)$ time by analyzing an auxiliary graph to G that summarizes zone-level adjacencies. A complete description of this auxiliary graph and the assessment of the pocket set of each zone is provided in King et al. [13]. Once the pocket set of zone $z(v)$ has been evaluated, the conditions on $R(v)$ contained in Theorem 1 can be assessed directly in $O(m(G)|R(v)|)$ time using $m(G)$ separate searches on $R(v)$. King et al. [12] propose a more efficient algorithm for evaluating these conditions in $O(|R(v)|)$ time by eliminating redundant exploration that may arise among these searches. For the purposes of comparing their performance in the of the numerical experiments conducted in this paper, the $O(m(G)|R(v)|)$ time algorithm will be referred to as the *basic* geo-graph contiguity algorithm, while the $O(|R(v)|)$ time algorithm will be called the *efficient* geo-graph contiguity algorithm.

While Theorem 1 limits the maximum number of vertices that need to be visited to assess district contiguity during local search, average case analysis is more critical in practice. To estimate the benefit of the geo-graph model in practical districting problems, King et al. [13] consider the state of Kansas and the partitioning of its census blocks into the districts of 109th United States Congress, finding that evaluating contiguity through the conditions in Theorem 1 reduced the number of edges visited during contiguity assessments by three orders of magnitude when compared to simple graph search. Both simple (breadth-first and depth-first) search and geo-graph search were applied to the census blocks in these districts, finding that geo-graph search visited 7.92 edges on average, while simple search averaged 11,449 edges for breadth-first search (BFS) and 91,894 edges for depth-first search (DFS). While the results for Kansas are significant, the computational savings realized by geo-graph search in other scenarios will depend on several factors, such as the average number of vertices per district and the number of cut-vertices in each district. These factors will vary from problem instance to problem instance, and may evolve as local search progresses. For example, a local search procedure that favors compact districts would be less likely to produce districts with cut-vertices than one optimizing a different objective. Therefore, applying the geo-graph approach to a diverse set of large practical districting problems will provide a more comprehensive estimate of how its computational savings depends on these factors.

3 Practical district design

The geo-graph model makes two key assumptions about the geometry of unit boundaries in a districting problem; these assumptions must be satisfied if a districting problem is to be modeled with a geo-graph. First, each boundary must be a single simple closed curve, and second, the boundary curves must collectively divide the plane into regions corresponding to the $|V| + 1$ units. The latter assumption is true for the census blocks of each state; unpopulated areas within the state border, including bodies of water, are covered by census blocks. However, the boundary of a census block

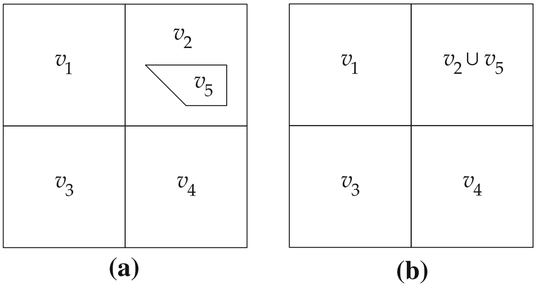

Fig. 1 Example of a unit containing a hole and its resolution by merging units. **a** Unit containing a hole. **b** Hole resolved by merging units

is defined by natural and man-made geography; while these boundaries are always closed curves, they may consist of multiple or non-simple curves. In the shapefile for each state, each census block is stored as a polygon that is bounded by one or more simple closed curves called "rings" [25]. The number of rings for each block can be extracted using GIS software; any block with more than one ring exhibits a *boundary violation*.

Boundary violations can be broadly categorized into two types: *hole violations* and *piece violations*. Hole violations occur when one boundary ring is nested inside another (e.g., unit v_2 in Fig. 1a), while piece violations occur when a single census block is composed of several non-contiguous areas (e.g., unit v_3 in Fig. 2a). While boundary violations may arise in census blocks, they are uncommon. For example, of the 350,169 census blocks in the state of New York during the 2010 census, 6225 blocks have hole violations and 263 blocks have piece violations; roughly 1.9% of census blocks have boundary violations, the majority of which are hole violations. Of the 157,508 census blocks in Massachusetts, 4747 blocks have hole violations and 125 blocks have piece violations; roughly 3.1% of census blocks have boundary violations, with the majority being hole violations. These examples demonstrate that boundary violations are relatively infrequent in practice when compared to the number of units. The rest of this section proposes methods for eliminating boundary violations and discusses their superficial impact on the solution space.

3.1 Hole violations

As with zone holes, a unit is said to contain a hole if one can draw a closed curve through the interior of the unit, such that this curve encloses one or more points of

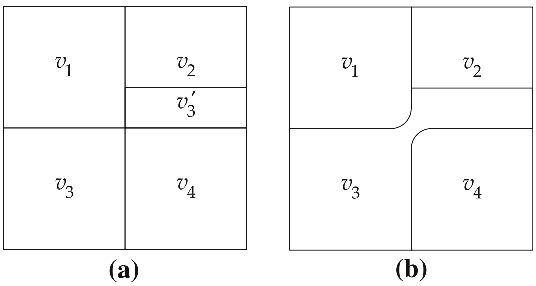

Fig. 2 Example of a unit with two point-connected pieces (v_3 and v_3') and their resolution by opening. **a** Unit with two point-connected pieces. **b** Pieces combined by opening

another unit; if such a curve passes through points on the boundary of the unit, then a *degenerate hole* occurs. For example, unit v_2 in Fig. 1a contains a hole that is filled by unit v_5, and the boundary of unit v_2 is composed of two simple closed curves. If unit v_5 were translated to its left, such that its upper-left corner touched the left outer boundary of v_2, then v_2 would exhibit a degenerate hole. While Fig. 1a depicts one hole that contains one unit, this definition can be generalized consider a unit with multiple holes, each of which could contain multiple units.

One logical approach for eliminating a hole violation is by *merging* the unit(s) in a hole with the violating unit, combining all of their land areas and their populations. This strategy is depicted in Fig. 1b, where units v_2 and v_5 have been merged into a single unit $v_2 \cup v_5$. This approach eliminates solutions that place these units in different districts. However, the pruned solutions tend to be infeasible, as the units located inside a hole of another unit are unlikely to be sufficiently populous to compose an entire district. For example, the maximum population in a hole of any unit of the four states considered in this paper (Arizona, Massachusetts, New Mexico, and New York) is 10,172 people in Massachusetts, where a single census block surrounds all of Nantucket, yet this hole contains less than 2% of the average district population in the state; such a small population would certainly violate any reasonable measure of population balance among the districts. This example demonstrates that, in practice, merging such units can eliminate hole violations without pruning feasible solutions from the solution space, though infeasible solutions will be pruned. This lack of impact on the feasible solution space is important, as hole violations tend to be more frequent than piece violations.

3.2 Piece violations

Piece violations occur when a census block is not a single contiguous area, but comprises several non-contiguous areas called *pieces*. In many cases, all of the pieces of a unit may reside in one or more holes of another single unit; this category includes 56 of the 72 units with piece violations in Arizona and 247 of 263 units with piece violations in New York. These piece violations are automatically eliminated when hole violations are eliminated by merging. Other strategies can be applied to resolve piece violations that do not fall in this category. When multiple pieces of the same unit share isolated points on their common boundary, this piece violation can be eliminated by *opening* these isolated points; for example, the two areas marked v_3 and v_3' in Fig. 2a are two pieces of one unit connected at a common corner point, and Fig. 2b depicts how this corner point can be opened to combine both pieces into a single area. Graphically, this resolution is depicted by rounding off the two corners that pinch together at this common point in Fig. 2a, which is equivalent to redrawing parts of the boundaries on the plane. Theorem 2 shows that redrawing these boundaries neither removes nor alters edges in the geo-graph.

Theorem 2 *Let $G_B = (V_B, E_B)$ be the plane embedding of the graph whose faces are each bounded by simple closed curves, such that the faces of G_B are F_B. For any two faces $f_{\mathrm{I}}, f_{\mathrm{II}} \in F_B$ that share a vertex $v \in V_B$ (but do not share an edge), there is an embedded plane graph, $G_B' = (V_B', E_B')$, with faces F_B' that are each bounded* *by simple closed curves, and a surjective function* $\mu_F : F_B \to F'_B$ *and a bijective function* $\mu_E : E_B \to E'_B$ *such that*

1. $\mu_F(f_{\mathrm{I}}) = \mu_F(f_{\mathrm{II}})$
2. *Every pair of faces* $f_1, f_2 \in F_B$ *with* $f_1 \neq f_{\mathrm{I}}, f_{\mathrm{II}}$ *has* $\mu_F(f_1) \neq \mu_F(f_2)$
3. *If edge* $e \in E_B$ *separate faces* f_a *and* f_b *in* G_B*, then edge* $\mu_E(e) \in E'_B$ *separates faces* $\mu_F(f_a)$ *and* $\mu_F(f_b)$ *in* G'_B

Proof Let $(e_1, e_2, \ldots, e_k)$ be the sequence of edges incident to v, ordered clockwise, and let the faces incident to v be $(f_1, f_2, \ldots, f_k)$, such that face f_j is bounded by edges e_j and e_{j+1} for $j \in 1, 2, \ldots, k$ (for convenience of notation, let $e_{k+1} = e_1$). Assume, without loss of generality, let $f_1 = f_{\mathrm{I}}$ and $f_{k_0} = f_{\mathrm{II}}$ for some $k_0 \in \{3, 4, \ldots, k-1\}$. Since the boundary of each face is a simple closed curve, no edge e_j can be a loop, and every f_j must be distinct. Subdivide each edge e_j by adding vertex v_j, and rename the subdivided pieces as $e_{j,1}$ and $e_{j,2}$, where $e_{j,2}$ is incident to v. For each face f_j, add an edge c_j between vertices v_j and v_{j+1}, with c_j is contained entirely in the interior of f_j (other than its two endpoints); the edge sequence $C = (c_1, c_2, \ldots, c_k)$ is a cycle whose corresponding curve encloses v. Moreover, c_j divides face f_j into two faces: call these $f_{j,1}$ and $f_{j,2}$, where $f_{j,2}$ includes v on its boundary. Delete the edges $e_{j,2}$, combining the area inside the cycle C into a single area; furthermore, delete the edges c_1 and c_{k_0}, combining this area with the areas of faces $f_{1,1}$ and $f_{k_0,1}$, and call this combined face f'_0.

To contract the remaining edges of cycle C, let $e'_2 = e_{2,1}$ and $f^0_2 = f_{2,1}$, then execute the following algorithm for $i = 3, 4, \ldots, k_0 - 1$.

1. Let u_j be the endpoint of $e_{j,1}$ that is not v_j
2. Add the edge $e'_j = v_2 u_j$ to the graph and embed it in the area enclosed by face f^0_{i-1}
3. This edge splits f^0_{i-1} into two faces; call the new face that includes e'_{j-1} as f'_{j-1}
4. Delete edge $e_{j,1}$ to combine the other new face with face $f_{j,1}$, and call this combined face f^0_j

When this algorithm is complete, vertices $v_3, v_4, \ldots, v_{k_0}$ each have degree two. Remove these vertices, combining edges $c_2, c_3, \ldots, c_{k_0-1}$ with edge $e_{k_0,1}$; call this combined edge e'_{k_0}. Similarly, let $e'_{k_0+1} = e_{k_0+1,1}$ and $f^0_{k_0+1} = f_{k_0+1,1}$ and execute the above algorithm for $i = k_0+2, k_0+3, \ldots, k-1$, then remove vertices $v_{k_0+2}, v_{k_0+3}, \ldots, v_{k-1}$ to combine edges $c_{k_0+1}, c_{k_0+2}, \ldots, c_{k-1}$ with edge $e_{k,1}$ to form edge e'_k. Also, rename faces $f^0_{k_0-1}$ and f^0_{k-1} as f'_{k_0-1} and f'_{k-1}, respectively.

For any face $f \in F_B$ such that $f \notin \{f_1, f_2, \ldots, f_k\}$, there is a face embedded identically in F'_B; call this identical face $f' \in F'_B$ and let $\mu_F(f) = f'$. Similarly, for any edge $e \in E_B$ such that $e \notin \{e_1, e_2, \ldots, e_k\}$, there is an edge embedded identically in E'_B; call this edge $e' \in E'_B$ and let $\mu_E(e) = e'$. Let $\mu_F(f_I) = \mu_F(f_{II}) = f'_0$, and for each f_i with $i \in \{2, 3, \ldots, k_0-1, k_0+1, \ldots, k\}$, let $\mu_F(f_i) = f'_i$. For each $i \in \{1, 2, \ldots, k\}$, let $\mu_E(e_j) = e'_j$.

From these definitions, μ_F is surjective and μ_E is bijective. Moreover, $\mu_F(f_{\mathrm{I}}) = \mu_F(f_{\mathrm{II}})$, but otherwise $\mu_F(f_1) \neq \mu_F(f_2)$ for any $f_1, f_2 \in F_B$ with $f_1 \neq f_{\mathrm{I}}, f_{\mathrm{II}}$. For any $e \in E_B$, there are four cases:

1. $e \in \{e_1, e_2\}$: In G_B, edges e_1 and e_2 separate face f_{I} from faces f_k and f_2, respectively. In G'_B, edges $e'_1 = \mu_E(e_1)$ and $e'_2 = \mu_E(e_2)$ separate face $f'_0 = \mu_F(f_{\mathrm{I}})$ from faces $f'_k = \mu_F(f_k)$ and $f'_2 = \mu_F(f_2)$, respectively.
2. $e \in \{e'_{k_0}, e'_{k_0+1}\}$: In G_B, edges e_{k_0} and e_{k_0+1} separate face f_{II} from faces f_{k_0-1} and f_{k_0+1}, respectively. In G'_B, edges $e'_{k_0} = \mu_E(e_{k_0})$ and $e'_{k_0+1} = \mu_E(e_{k_0+1})$ separate face $f'_0 = \mu_F(f_{\mathrm{II}})$ from faces $f'_{k_0-1} = \mu_F(f_{k_0-1})$ and $f'_{k_0+1} = \mu_F(f_{k_0+1})$, respectively.
3. $e = e_i$ for some $i \in \{3, 4, \ldots, k_0-2, k_0-1, k_0+2, k_0+3, \ldots, k\}$: In G_B, edge e_i separates face f_{i-1} from face f_i. By the algorithm, the $e'_i = \mu_E(e_i)$ separates face $f'_{i-1} = \mu_F(f_{i-1})$ from face $f'_i = \mu_F(f_i)$ in G'_B.
4. *Other edges*: By the construction of G'_B, any other edge $e \in E_B$ is on the boundary of face f in G_B if and only if edge $\mu_E(e) \in E'_B$ is on the boundary of face $\mu_F(f)$ in G'_B.

Therefore, for any $e \in E_B$ that separates faces f_a and f_b in G_B, edge $\mu_E(e) \in E'_B$ separates faces $\mu_F(f_a)$ and $\mu_F(f_b)$ in G'_B □

While no edges are removed by the transformation presented in Theorem 2, any units that are point-adjacent across the opened point will no longer be point-adjacent after this point is opened; for example, the boundaries in Fig. 2a would include v_1 in $R(v_4)$ (and vice versa), but those in Fig. 2b would not. While this reduction in the size of these augmented neighborhoods will not affect the solution space, it may reduce the number of vertices that must be analyzed in Theorem 1; however, these saving are unlikely to be substantial in practice due to the small number of units with piece violations.

When two pieces of a unit are entirely disconnected (i.e., they do not share an isolated point on their common boundary), *annexing* may be able to resolve piece violations without pruning feasible solutions. For example, suppose all but one of the pieces of a unit are contained in holes of other units that share a common boundary with the last piece. If each hole-filling piece is annexed by the unit that surrounds it, the single remaining piece does not reside in a hole, and hence, satisfies the boundary assumptions. Moreover, no unit adjacencies are lost during annexation, and the solution space is unaffected. As another example, unit v_3 in Fig. 3a has two entirely disconnected pieces, each of which is adjacent to the same set of units (i.e., units v_1 and v_2). If one of these pieces is annexed to v_2 (Fig. 3b), the piece violation is resolved without altering the solution space (i.e., v_1 remains adjacent to v_2 and v_3). In the examples considered in this paper, unit populations are not altered when resolving

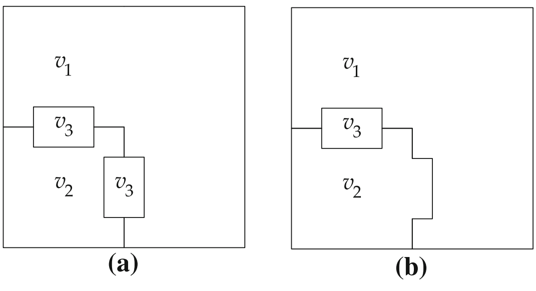

Fig. 3 Example of a unit with two disconnected pieces and their resolution by annexing. **a** Unit with two disconnected pieces. **b** Violation resolved by annexing

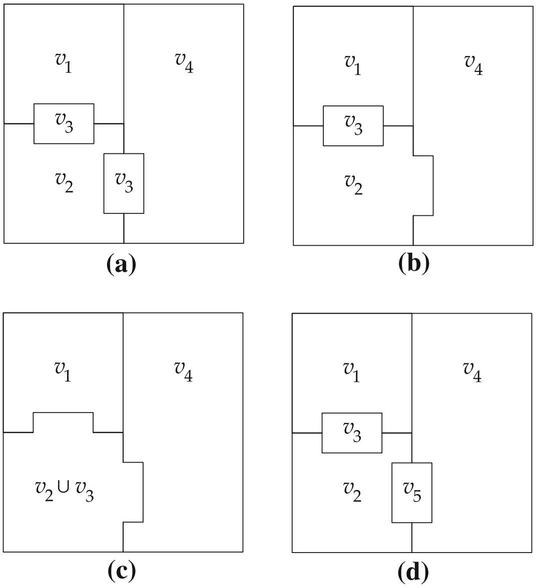

Fig. 4 Example of a unit with two disconnected pieces and their resolution by annexing, splitting, and merging. **a** Unit with two disconnected pieces. **b** Violation resolved by annexing. **c** Violation resolved by merging. **d** Violation resolved by splitting

a piece violation by annexation; hence, this analysis assumes that any parcel of land annexed by a census block either (a) is unpopulated, or (b) attributes its population to its original census block. As any such annexation is performed only to adapt a multi-piece census block to the geo-graph model, rather than enforce an actual annexation of land, these assumptions are reasonable.

There are cases when annexation cannot be applied without affecting the solution space. For example, unit v_3 in Fig. 4a has two pieces, each adjacent to a different set of units; one is adjacent to v_1 and v_2, while the other is adjacent to v_2 and v_4. While annexing can resolve this piece violation, doing so eliminates an edge from the graph; for example, the annexation depicted in Fig. 4b eliminates the adjacency between v_3 and v_4, which may prune feasible solutions from the solution space. This piece violation can also be resolved by merging, but doing so may also prune feasible solutions from the solution space; for example, Fig. 4c merges units v_2 and v_3, which eliminates solutions that place these units in different districts. As a third alternative, *splitting* such pieces into separate units, as depicted by v_3 and v_5 in Fig. 4d, will not prune feasible solutions, but will introduce new infeasible solutions (i.e., those where v_3 and v_5 are assigned to different districts). While these cases cannot be resolved without affecting the solution space, the scope of their impact depends on the number of such resolutions that need to be made. In New York, only 7 of 350,169 census blocks need to be resolved in this way, while none of the 168,609 census blocks in New Mexico require these resolutions. Due to the relative scarcity of these resolutions, any impact on the solution space is likely to be superficial.

4 Input data

The experiments conducted in this paper draw data from several public sources. These data fall into four categories: geographic data describing the arrangement of the census blocks in the plane, district data assigning these blocks to their current districts, population data measuring the number of residents in each block, and voting data that describe the political preferences of these residents. The United States Census Bureau provides geographic data from the 2010 Census as shapefiles for use in geographic information system (GIS) software [26]. The sets of neighbors and augmented neighbors for each block were extracted from these files using the ArcGIS software package. District data for the 111th United States Congress and population data from the 2010 Census were extracted from MABLE/Geocorr2010 [15]; two census blocks in Massachusetts (with Geographic Identifier 250250606001015 and 250250701011005) were moved from the eighth district to the ninth district to create a contiguous ninth district. Voting data for the two major political parties (Democratic and Republican) were retrieved from the Public Mapping Project [16], which gathers these data at the voting tabulation district level and disaggregates them to the census block level. This methodology can produce non-integer numbers of votes for each party in a census block, emphasizing the approximate nature of the disaggregation procedure. Nonetheless, if one had access to exact measures of voting preference, these data could be substituted before applying local search. Furthermore, voting data for some census blocks are missing in each state; in these cases no political preferences are inferred and vote counts are assumed to be zero for both parties. In the experimental results reported in this paper, voting data in Arizona, New Mexico, and New York were based on data from the 2008 presidential election, while voting data in Massachusetts were based on party registration.

Four states were analyzed in this paper: Arizona, Massachusetts, New Mexico, and New York. Table 1 describes some of their key characteristics. Among these states, New York has both the most census blocks (350,169) and the most districts (29), while Massachusetts has the fewest census blocks (157,508) and New Mexico has the fewest districts (three). These district counts do not take into account the congressional reapportionment that occurred after the 2010 Census, in which Arizona gained one district (to nine), Massachusetts lost one district (to nine), New York lost two districts (to 27), and New Mexico neither gained nor lost a district [3]. Therefore, the experiments conducted in this paper are not applicable to the 2010 redistricting process (except in New Mexico), but rather demonstrate the ability of the geo-graph model to efficiently assess contiguity during local search.

Since the number of districts in a state is not necessarily proportional to its number census blocks, the average number of blocks per district varies by state, from 12,075 in New York to 56,203 in New Mexico. This variation is important when assessing contiguity using simple search, since simple search may need to visit every vertex in a district. The number of census blocks in a district will depend on the population residing in those blocks, and hence, the actual number of blocks in each district will vary. In contrast, geo-graph contiguity algorithms visit only vertices in $R(v)$ when the block corresponding to vertex $v \in V$ is removed from its current district. While the number of census blocks in each state varies from 157,508 in Massachusetts to

Table 1 State characteristics for Arizona (AZ), Massachusetts (MA), New Mexico (NM), and New York (NY)

State	AZ	MA	NM	NY
Census blocks (2010)	241,666	157,508	168,609	350,169
Congressional districts (111th Congress)	8	10	3	29
Population (2010)	6,392,017	6,547,629	2,059,179	19,378,102
Average blocks per district	30,208	15,751	56,203	12,075
Unpopulated blocks	126,924	61,174	107,799	107,362
Blocks containing holes	5114	4747	3310	6225
Blocks with multiple pieces	72	125	102	263
Blocks remaining after preprocessing	230,234	147,565	162,565	339,933
Unpopulated blocks after preprocessing	120,438	55,040	103,380	101,546

Table 2 Statistics on the size of the augmented neighborhood, $R(v)$, among the set of census blocks in each state (all statistics measured in number of census blocks)

State	AZ	MA	NM	NY
Average	6.49	6.56	6.59	6.81
SD	16.68	12.57	15.36	9.25
Min.	2	2	2	2
Max.	127	72	219	74
Median	6	6	6	6

350,169 in New York, the average size of $R(v)$ remains very small, from 6.49 census blocks in Arizona to 6.81 census blocks in New York, as shown in Table 2. Though the range of values of $R(v)$ can be somewhat large, with one census block in New Mexico having 219 augmented neighbors, these average values demonstrate that the augmented neighborhood of each unit remains quite small on average.

Finally, one complicating factor encountered when creating districts at the census block level is that many blocks are unpopulated (e.g., blocks that lie entirely in a body of water). The number of such blocks will depend on how residents are distributed in the state, and can be substantial. For example, nearly 64% of the census blocks in New Mexico are unpopulated. While such blocks arise naturally, they can complicate the local search process since they do not affect relevant constraints and objectives that emphasize population characteristics of the districts (e.g., population balance, election competitiveness). The impact of these unpopulated blocks on local search will be discussed in Sect. 5.

5 Experimental design

This paper applies local search at the census block level to demonstrate the ability of the basic and efficient geo-graph contiguity algorithms to efficiently assess contiguity while identifying locally optimal United States Congressional Districts. Two constraints are enforced within local search: district contiguity and population balance, where population balance provides an explicit upper and lower bound for the population of an individual district. The specific upper and lower population bounds considered in each local search instance are discussed in Sect. 6. Several objectives are available, reflecting the wide breadth of objectives available to district designers.

5.1 Objectives

Compactness, maximum population balance, and maximum election competitiveness objectives are considered in these experiments. While each objective is considered separately in this section, practitioners can consider multiple objectives by combining their individual values into a weighted sum or through other standard methods of multi-objective optimization. A minimum cut objective, common in classical graph partitioning problems involving VLSI and parallel computing (e.g., [10]), provides one measure of district compactness. By noting that a circle is the most compact shape containing a two-dimensional area, the Schwartzberg index quantifies the relative compactness of a district as the ratio between the district's perimeter and the perimeter of a circle that contains the same area [21]. Hence, districts with shorter perimeters are considered more compact. Since each edge in a geo-graph corresponds to a segment of shared boundary between two units, placing these units in different districts indicates that this length of boundary is included in the perimeter of both districts. Hence, the total perimeter among all districts is related to the set of edges whose endpoints are in different districts. Minimizing the perimeter of these districts is therefore related to minimizing the size of this edge cut, computed as

$$cut(G) = |\{v_1 v_2 \in E : v_1, v_2 \in V, z(v_1) \neq z(v_2)\}| , \quad (1)$$

where $G = (V, E, B, z)$ is a geo-graph. While population always serves as a constraint in the districting process, some designers wishing to place more emphasis on population balance can formulate a maximum population balance objective that seek district assignments that minimize population imbalance, computed as

$$pbal(G) = \sum_{j=1}^{m(G)} \left(\sum_{v \in V(j)} pop(v) - \overline{p} \right)^2 , \quad (2)$$

where $pop(v)$ is the population of the census block associated with vertex $v \in V$, $\overline{p} = \sum_{v \in V} pop(v)/m(G)$ is the average district population, and $V(j)$ is the set of all vertices assigned to district $j \in M(G)$. Maximum election competitiveness seeks to reduce the disparity between voters associated with each political party in a district, such that shifts in the preferences of the electorate lead to a change in election results [4]. While no single numerical metric encapsulating all facets of competitiveness has been proposed, one approach could seek to minimize the political imbalance (i.e., difference in the number of voters aligned with each party) in each district, corresponding to minimization of

$$comp_1(G) = \sum_{j=1}^{m(G)} \left(\sum_{v \in V(j)} party_1(v) - \sum_{v \in V(j)} party_2(v) \right)^2, \tag{3}$$

where $party_i(v)$ is the number of residents of the census block associated with vertex $v \in V$ who are aligned with party i, where there are assumed to be two major parties (e.g., the Republican and Democratic parties in the United States). Another approach could seek to maximize the number of districts with competitive elections; this goal is accomplished by maximizing

$$comp_2(G) = \sum_{j=1}^{m(G)} I\left\{ \left| \sum_{v \in V(j)} party_1(v) - \sum_{v \in V(j)} party_2(v) \right| \Big/ \sum_{v \in V(j)} pop(v) < 0.01k \right\}, \tag{4}$$

where $I\{.\}$ is an indicator function that takes a value of one when its argument is true and zero when its argument is false, and an election is considered competitive when voters aligned with the two parties differ by at most $k\%$ of the total district population. However, these formulations for competitiveness are not exhaustive; other formulations could be proposed to capture the wishes of a district designer. Moreover, the specific types of objective presented here reflect only a subset of those that a designer may wish to consider; additional objectives are discussed and formulated by, for example, di Cortona et al. [6] and Kalcsics et al. [9]. While these examples quantify potential numerical objectives for the redistricting process, Webster [28] and Butler and Cain [4] present further discussion of a wider variety of redistricting principles.

5.2 Local search

Algorithm 1 summarizes the steepest descent local search algorithm used in these experiments. Beginning with an initial set of districts summarized by the geo-graph $G_0 = (V, E, B, z_0)$, this algorithm iteratively identifies a vertex and zone pair, (v, j), such that moving the vertex v to zone j improves the objective the most among all feasible pairs. Only the zone assignments change over these iterations, and hence, at any iteration i the geo-graph $G_i = (V, E, B, z_i)$ differs from G_0 only in its zoning function. The pocket set of the zones must be updated after each iteration, which can be completed in $O(m(G)^2)$ time [12]. As Algorithm 1 iterates, executing Line 5 may require significant computation to determine whether moving vertex v to zone j both improves the objective and is feasible. In this paper, feasibility considers contiguity constraints (i.e., each zone must be contiguous) and population balance (i.e., district populations must fall within specified upper and lower bounds). Since a transfer must satisfy all three conditions (objective improvement, district contiguity, and population balance) to be added to $BestMoves$, a transfer can be discarded when one of these conditions is violated, and hence, it may be possible to reduce computation by changing the order in which local search evaluates these conditions.

Algorithm 1: Perform local search on a geo-graph using steepest descent

```
Input  : Zone-connected and population-balanced geo-graph G_0 = (V, E, B, z_0)
Output: Locally optimal, zone-connected, and population-balanced geo-graph G_k = (V, E, B, z_k)

1  k ← 0;
2  repeat
3      BestMoves ← ∅;
4      forall the v ∈ V and j ∈ M(G) − z(v) do
5          if ImprovesObj(v, j) ∩ IsFeasible(v, j) then
6              if BetterThanBest(v, j) then
7                  BestMoves ← {(v, j)};
8              else if AsGoodAsBest(v, j) then
9                  BestMoves ← BestMoves ∪ {(v, j)};
10     if BestMoves ≠ ∅ then
11         k ← k + 1;
12         Choose (v*, j*) randomly from BestMoves;
13         z_k(x) ← z_{k−1}(x) for all x ∈ V − v*;
14         z_k(v*) ← j*;
15         G_k ← (V, E, B, z_k);
16 until BestMoves = ∅;
17 return G_k;
```

Moving vertex v to zone j changes the populations of two districts; the new populations arising from this transfer can be computed and compared to the feasible population bounds in $O(1)$ time. Using the efficient geo-graph contiguity algorithm in [12], zone contiguity can be assessed in $O(|R(v)|)$ time. Determining how this transfer affects the objective will depend on the type of objective chosen by the designer. Consider each of the objectives in (1) through (4); the impact of transferring a single vertex on the minimum cut objective can be assessed in $O(|N(v)|)$ time by iterating through the edges adjacent to v and determining whether its transfer causes them to become cut or uncut, while the impact on maximum population balance and maximum election competitiveness can be assessed in $O(1)$ time. When considering one of these objectives, it seems prudent to assess population balance and objective improvement due to a vertex transfer before assessing its impact on district contiguity, since $|R(v)| \geq |N(v)|$ by definition. However, if a more computationally intensive objective were chosen, a different order could become preferable.

While delaying contiguity assessments until necessary provides one avenue to reduce computations when assessing contiguity, further reductions can be realized by recycling the outcomes of these contiguity assessments. Each iteration of local search alters the contents of two districts: one that loses a vertex and one that gains a vertex. If the geo-graph contains more than two districts [i.e., $m(G) > 2$], then at least one district is not altered in that iteration, allowing some contiguity assessments to be stored and recycled. For example, suppose zone $j \in M(G)$ was last altered in iteration k_j of local search, and vertex $v \in V(j)$ last had its impact on contiguity assessed (with outcome q) in iteration k_v of local search; if $k_v > k_j$, then outcome q can be recycled when local search tests whether vertex v can be removed from district j, rather than reassessing this outcome. To reduce the need for contiguity assessments, the computational studies presented in this paper recycle contiguity assessments in this way.

The search process is complicated by the presence of a large number of unpopulated census blocks in each state, as shown in Table 1. Under a maximum population balance or maximum election competitiveness objective, local search will have no incentive to move an unpopulated census block into a new district, since doing so does not affect either the populations or party affiliations in the districts. Therefore, using either of these objectives in isolation may be insufficient to explore the solution space, as the unpopulated blocks create vast flat plateaus of unchanging objective values in the solution space that steepest descent local search will not cross. To encourage exploration of these plateaus, a secondary minimum-cut objective can be added to maximum population balance and maximum political balance objectives. To avoid sacrificing the emphasis on the primary objective, the two objectives are not combined; instead, each solution represented by a geo-graph, G, produces an ordered pair of objective values $f(G) = (f_1(G), f_2(G))$ such that $f_1(G)$ measures the primary objective (i.e., either population imbalance or political imbalance) of the districts in G, and $f_2(G)$ counts the number of edges cut in G. The objective values of two solutions, G_1 and G_2, are compared *lexicographically*, such that the secondary minimum cut objective is used solely to break ties in the primary objective; in other words, solution G_1 is considered superior to G_2 if and only if either $f_1(G_1) < f_1(G_2)$, or $f_1(G_1) = f_1(G_2)$ and $f_2(G_1) < f_2(G_2)$. Under this formulation, local search can move unpopulated census blocks between districts to reduce the number of edges cut by a geo-graph, even though such transfers do not affect the primary objective.

6 Numerical results

The computational studies presented in this paper apply local search, as formulated in Algorithm 1, to develop United States Congressional Districts in four states: Arizona, Massachusetts, New Mexico, and New York. The initial solution in each local search run (i.e., G_0 in Algorithm 1) is the set of districts of the 111th United States Congress, using population data from the national census conducted in 2010. Numerical summaries for the initial solution in each state are provided in Table 3. Within local search, the population balance constraint sets the maximum and minimum permitted populations of any district to the maximum and minimum populations of the districts in the initial solution. Two different objectives scenarios are considered, each with two lex-

Table 3 Population and voting data of districts of the 111th Congress, measured by the 2010 Census and 2008 party voting data (voting disparity in a district is the magnitude of the difference between votes for the two major parties)

State	AZ	MA	NM	NY
Total population	6,392,017	6,547,629	2,059,179	19,378,102
Total Democratic votes	1,033,834	1,515,999	471,647	4,802,461
Total Republican votes	1,229,227	468,565	346,386	2,751,169
Maximum district population	972,839	664,919	701,939	713,512
Minimum district population	656,833	644,956	663,956	611,838
Average district population	799,002	654,763	686,393	668,210
Maximum district voting disparity	85,542	184,088	66,661	211,095
Minimum district voting disparity	13,438	69,264	3165	4052
Average district voting disparity	43,597	104,743	43,864	73,766

icographic objectives. The first scenario considers a primary objective of minimizing population imbalance as formulated in (2) and a secondary objective of minimizing the number of cut edges as formulated in (1). The second scenario considers a primary objective of minimizing political imbalance in the districts as formulated in (3) and a secondary objective of minimizing the number of cut edges as formulated in (1). Though a geo-graph can be a multi-graph when units share more than one segment on their common boundary, multiple edges joining two vertices are considered a single edge for the purpose of computing the size of a cut in these numerical experiments. In each case, local search was allowed to run until encountering a local optimum.

The primary goal of these computational studies is to quantify the ability of the geo-graph framework to reduce contiguity-related computations. To this end, four methods were used to assess district contiguity during local search: the basic geo-graph search method that directly verifies the conditions of Theorem 1 in $O(m(G)|R(v)|)$ time using simple search, the efficient geo-graph algorithm described in [12] that assesses contiguity in $O(|R(v)|)$ time, and simple breadth-first and depth-first search algorithms applied to the entire district using the approach of Ricca and Simeone [20]. Summary statistics for the number of edges visited by each type of approach will be reported and compared for each instance of local search.

6.1 Population balance objective

Since these districts of the 111th United States Congress were drawn based on the 2000 Census, population shifts that occurred in the intervening decade led the populations of these districts to become unbalanced in the initial solutions considered by local search. The maximum and minimum district populations in each state are shown in Table 3; while the districts in Massachusetts remained relatively balanced with a range of 20,000 people, the districts in Arizona became highly unbalanced with a range of more than 300,000 people. Table 4 describes the performance of local search when applied to the four states when maximizing population balance, with a secondary objective of minimizing the number of cut edges. Figure 5 depicts the evolution of these objectives over local search iterations. In general, local search is able to find a local optimum that exhibits a high degree of population balance in each state. Once local search terminates, the districts' populations range from 799,001 to 799,003 in Arizona, 654,761 to 654,765 in Massachusetts, and 668,118 to 668,361 in New York, with all districts in New Mexico attaining the optimal district population of 686,393; in each state, population imbalance as measured by (2) is reduced by more than 99.99% from its initial value. While the number of edges cut by these districts grows over the course of local search, this outcome arises from the lexicographic nature of the objectives, with local search seeking to reduce the number of cut edges only when doing so does not affect population balance. A designer seeking to emphasize district compactness more strongly could employ an objective that reflects this preference.

Of the four states, New York requires the most iterations of local search to arrive at a local optimum, which is consistent with its high level of initial population imbalance, coupled with its large number of census blocks and districts. In contrast, Arizona requires a much larger number of contiguity assessments during local search than New York despite requiring fewer iterations of local search; this difference arises from the relatively high proportion of cut-vertices that local search attempts to move in Arizona when compared to New York. In Arizona, only 2.4% of the census blocks investigated by local search can be removed from their districts, while 6.4% of the blocks investigated in New York can be removed. In contrast, local searches executed in Massachusetts and New Mexico can remove 68.2 and 22.3% of their investigated blocks, respectively. Local search chooses to investigate the contiguity of blocks whose transfer improves the population balance and minimum cut objectives, and hence, these

Table 4 Local search performance when maximizing population balance with a secondary objective of minimizing cut edges [population imbalance and cut size are computed as in (2) and (1)]

State	AZ	MA	NM	NY
Local search iterations	4158	296	276	7096
Number of contiguity assessments	622,137	1398	5424	388,765
Initial population imbalance	1.045×10^{11}	4.433×10^8	7.926×10^8	1.441×10^{10}
Final population imbalance	4.875	10.9	0	205,739
Initial cut size	4426	3811	1502	11,096
Final cut size	12,822	4398	2233	26,624

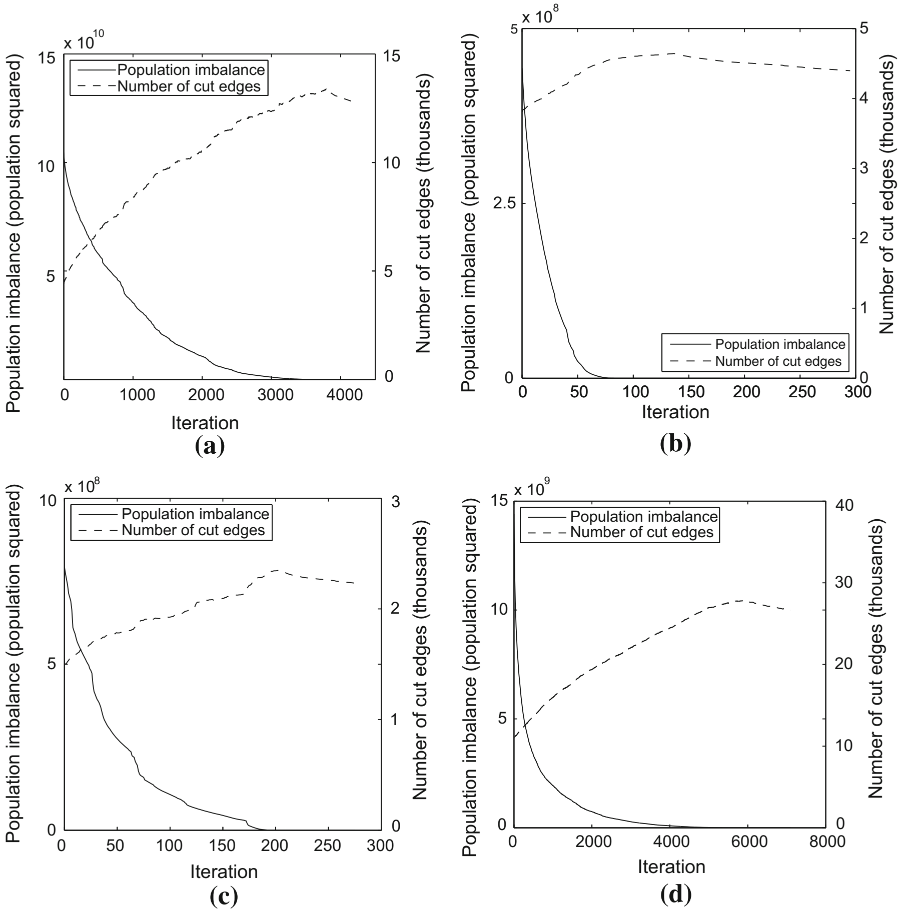

Fig. 5 Progression of population imbalance (population squared) and number of cut edges over iterations of local search in Arizona, Massachusetts, New Mexico, and New York, with a primary objective of maximizing population balance and a secondary objective of minimizing cut edges. **a** Arizona. **b** Massachusetts. **c** New Mexico. **d** New York

trends are not necessarily related to the number of districts or census blocks in each state, but rather, how the state population is distributed among these blocks and how these blocks are arranged in the plane.

Table 5 describes the number of edges visited by each type of contiguity algorithm during local search. Consistent with the results reported by King et al. [13], basic geo-graph search visits far fewer edges than either of the pure simple-search approaches; on average, basic geo-graph visits fewer than ten edges in each contiguity assessment, while simple search visits more than ten thousand edges on average. While efficient geo-graph search does not produce significant savings over basic geo-graph search, this result is due to the lack of pockets encountered during local search. The initial districts in each state do not contain any holes and, though not forbidden, local search does not create holes by transferring blocks to new districts. Since the efficient geo-

Table 5 Number of edges visited while assessing contiguity in local search when maximizing population balance in each state (all statistics except sample size measure numbers of edges)

State	Statistic	Basic geo-graph	Efficient geo-graph	Simple (BFS)	Simple (DFS)
AZ	Mean	7.36	7.36	72,147	72,948
	Median	4	4	79,942	80,002
	Max.	155	155	453,428	453,428
	Sample size	622,137	622,137	622,137	622,137
MA	Mean	9.48	10.42	13,093	24,512
	Median	3	4	13	45.5
	Max.	82	83	114,510	117,724
	Sample size	1398	1398	1398	1398
NM	Mean	9.84	10.07	45,300	54,825
	Median	7	7	50	194
	Max.	107	107	383,270	383,270
	Sample size	5424	5424	5424	5424
NY	Mean	7.53	7.26	57,779	59,575
	Median	4	4	52,192	73,478
	Max.	98	98	183,132	183,132
	Sample size	388,765	388,765	388,765	388,765

Table 6 Local search performance when maximizing political balance with a secondary objective of minimizing cut edges [political imbalance and cut size are computed as in (3) and (1)]

State	AZ	MA	NM	NY
Local search iterations	17,727	1090	1503	57,570
Number of contiguity assessments	4,501,969	23,758	55,047	15,786,347
Initial political imbalance	2.023×10^{10}	1.207×10^{11}	8.269×10^{9}	2.759×10^{11}
Final political imbalance	8.597×10^{9}	1.194×10^{11}	7.487×10^{9}	2.271×10^{11}
Initial cut size	4426	3811	1502	11,096
Final cut size	22,493	5902	4649	71,786

graph algorithm was developed to avoid additional computation in Theorem 1 when pockets are present, the performance of efficient geo-graph search is comparable to that of basic geo-graph search in these computational studies.

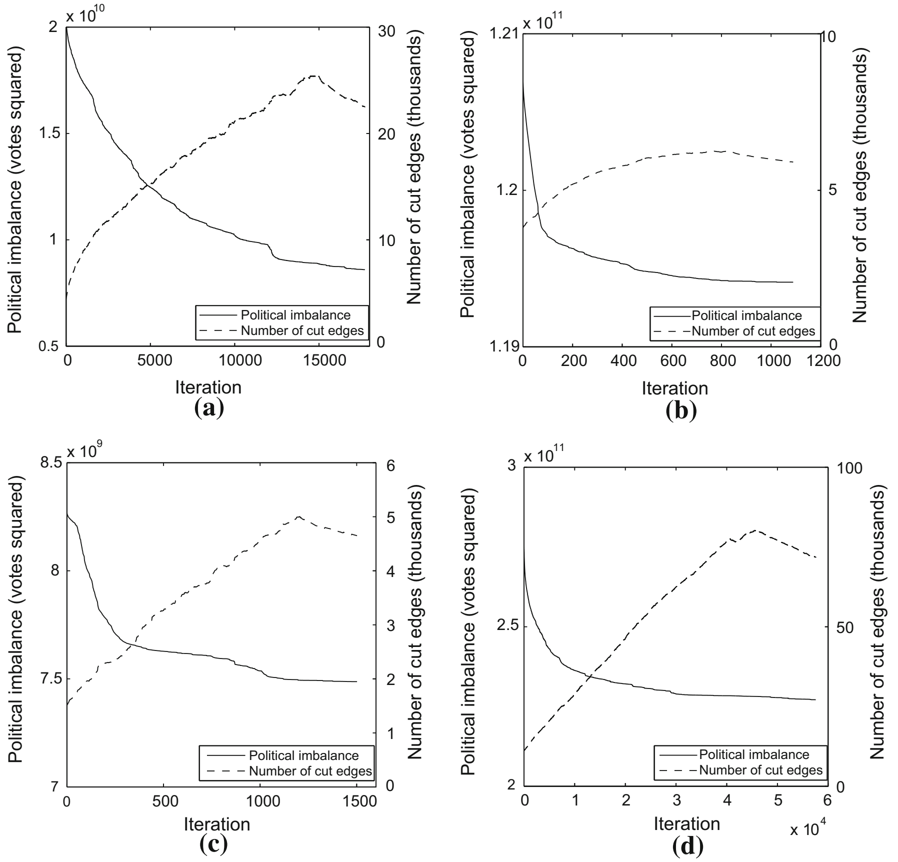

Fig. 6 Progression of political imbalance (votes squared) and number of cut edges over iterations of local search in Arizona, Massachusetts, New Mexico, and New York, with a primary objective of maximizing political balance and a secondary objective of minimizing cut edges. **a** Arizona. **b** Massachusetts. **c** New Mexico. **d** New York.

6.2 Political balance objective

Table 6 describes the performance of local search when optimizing political balance in the four states, with Fig. 6 depicting how the objectives evolve over local search iterations. Local search required many more iterations to terminate when compared to the population balance scenario; local search in Massachusetts required the fewest iterations (1090), while New York required the most iterations (57,570). Table 7 describes the performance of the four contiguity algorithms; as with the population balance scenario, geo-graph contiguity algorithms reduce the average number of edges visited during continuity assessments by approximately three orders of magnitude when compared to simple graph search.

Political balance differs from population balance in the optimal objective values observed at the end of local search; while it is theoretically possible to perfectly

Table 7 Number of edges visited while assessing contiguity in local search when maximizing political competition in each state (all statistics except sample size measure numbers of edges)

State	Statistic	Basic geo-graph	Efficient geo-graph	Simple (BFS)	Simple (DFS)
AZ	Mean	7.13	6.98	86,361	86,916
	Median	4	4	45,478	46,178
	Max.	188	193	452,520	452,552
	Sample size	4,501,969	4,501,969	4,501,969	4,501,969
MA	Mean	6.42	6.71	28,181	31,189
	Median	4	4	159	15,044
	Max.	78	78	114,580	117,995
	Sample size	23,758	23,758	23,758	23,758
NM	Mean	7.55	9.50	137,411	144,715
	Median	4	5	361	81,194
	Max.	112	112	383,268	383,268
	Sample size	55,047	55,047	55,047	55,047
NY	Mean	4.79	4.75	54,176	54,475
	Median	4	4	16,368	18,036
	Max.	109	109	183,118	183,118
	Sample size	15,786,347	15,786,347	15,786,347	15,786,347

balance population in a state, the ability to balance political affiliation depends on the overall political affiliations in the state. If the overall voting data in a state favor one party over the other, as in the four states considered in this paper (see Table 3), then perfect balance will be impossible and political imbalance cannot converge to zero. From another perspective, balancing political preferences at the district level can thwart political balance at the state level, since districts will be more likely to favor the party that is favored by the state. This influence is seen in the districts produced by local search in this paper, as all but one of the districts produced favors the same party as the overall state voting data. Hence, a designer seeking to balance political affiliation at the state level would not find the minimization objective in (3) suitable.

7 Conclusion

As political stakeholders seek to propose district plans that exhibit specific characteristics (e.g., political competitiveness) that suit their goals, computationally efficient tools that can help stakeholders develop these district plans are needed. The results presented in this paper demonstrate that the geo-graph model can serve as a platform for the development of these efficient tools. First, this paper proposes strategies for adapting practical unit shapes to fit the geo-graph requirement that each unit boundary is composed of a simple closed curve; by establishing methods to eliminate piece violations and hole violations that may be observed in practical unit shapes (e.g., census blocks), the results of this study allow the geo-graph model to be applied to a broad set of practical districting problems.

Second, the numerical experiments conducted in this paper demonstrate that geo-graph contiguity algorithms [12, 13] substantially reduce computation when assessing contiguity during local search approaches for creating practical political districts. When using local search to create United States Congressional Districts from census blocks in Arizona, Massachusetts, New Mexico, and New York, these geo-graph contiguity algorithms visit a number of edges that is three orders of magnitude smaller on average than the number visited by simple graph search algorithms (i.e., breadth-first search and depth-first search). On average, geo-graph contiguity algorithms visit fewer than eleven edges in all experiments, while simple graph search algorithms visit more than 13,000 edges on average in all experiments. While geo-graph contiguity algorithms allow local search to assess the satisfaction of contiguity constraints more efficiently, these contiguity algorithms do not limit a district designer's choices within other facets of the local search. For example, both the objective function used to evaluate district plans and the method for choosing a perturbation in each local search iteration (e.g., steepest descent, simulated annealing) can be selected by the district designer to best suit their goals.

Acknowledgements The computational work was conducted with support from the Simulation and Optimization Laboratory at the University of Illinois. This research was supported in part by the National Science Foundation [IIS-0827540]. The second author was supported in part by the Air Force Office of Scientific Research [FA9550-10-1-0387, FA9550-15-1-0100]. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the National Science Foundation, or the United States Government. This work originally appeared in the Ph.D. dissertation of the first author [11].

References

1. Altman, M.: Is automation the answer? The computational complexity of automated redistricting. Rutgers Comput. Technol. Law J. **23**(1), 81–142 (1997)
2. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. Eur. J. Oper. Res. **144**(1), 12–26 (2003)
3. Burnett, K.D.: Congressional apportionment. http://www.census.gov/prod/cen2010/briefs/c2010br-08.pdf (2011). Accessed 29 Dec 2011
4. Butler, D., Cain, B.E.: Congressional Redistricting: Comparative and Theoretical Perspectives. MacMillan Publishing Company, New York (1992)
5. D'Amico, S.J., Wang, S.J., Batta, R., Rump, C.M.: A simulated annealing approach to police district design. Comput. Oper. Res. **29**, 667–684 (2002)
6. di Cortona, P.G., Manzi, C., Pennisi, A., Ricca, F., Simeone, B.: Evaluation and Optimization of Electoral Systems. Society for Industrial and Applied Mathematics, Philadelphia (1999)
7. Drexl, A., Haase, K.: Fast approximation methods for sales force deployment. Manag. Sci. **45**(10), 1307–1323 (1999)
8. Horowitz, E., Sahni, S.: Fundamentals of Computer Algorithms. Computer Science Press, Rockville (1978)
9. Kalcsics, J., Nickel, S., Schröder, M.: Towards a unified territorial design approach—applications, algorithms and GIS integration. TOP **13**(1), 1–56 (2005)
10. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. **49**(1), 291–307 (1970)
11. King, D.M.: Graph theory models and algorithms for political districting: an approach to inform public policy. Ph.D. thesis, University of Illinois at Urbana-Champaign (2012)

12. King, D.M., Jacobson, S.H., Sewell, E.C.: Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning. Math. Program. Ser. A **149**(1–2), 425–457 (2015)
13. King, D.M., Jacobson, S.H., Sewell, E.C., Cho, W.K.T.: Geo-graphs: an efficient model for enforcing contiguity and hole constraints in planar graph partitioning. Oper. Res. **60**(5), 1213–1228 (2012)
14. Mehrotra, A., Johnson, E.L., Nemhauser, G.L.: An optimization based heuristic for political districting. Manag. Sci. **44**(8), 1100–1114 (1998)
15. Missouri Census Data Center: MABLE/Geocorr2010: Geographic correspondence engine with census 2010 geography. http://mcdc1.missouri.edu/MableGeocorr/geocorr2010.html (2011). Accessed 29 Dec 2011
16. Public Mapping Project: About the data. http://www.publicmapping.org/resources/data (2011). Accessed 29 Dec 2011
17. Ricca, F.: A multicriteria districting heuristic for the aggregation of zones and its use in computing origin-destination matrices. INFOR **42**(1), 61–77 (2004)
18. Ricca, F., Scozzari, A., Simeone, B.: Weighted Voronoi region algorithms for political districting. Math. Comput. Model. **48**(9), 1468–1477 (2008)
19. Ricca, F., Scozzari, A., Simeone, B.: Political districting: from classical models to recent approaches. 4OR **9**(3), 223–254 (2011)
20. Ricca, F., Simeone, B.: Local search algorithms for political districting. Eur. J. Oper. Res. **189**, 1409–1426 (2008)
21. Schwartzberg, J.E.: Reapportionment, gerrymanders, and the notion of compactness. Minn. Law Rev. **50**, 443–452 (1965)
22. Shirabe, T.: A model of contiguity for spatial unit allocation. Geogr. Anal. **37**(1), 2–16 (2005)
23. Shirabe, T.: Districting modeling with exact contiguity constraints. Environ. Plan. B Plan. Des. **36**(6), 1053–1066 (2009)
24. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)
25. Theobald, D.M.: Understanding topology and shapefiles. ArcUser. http://www.esri.com/news/arcuser/0401/topo.html (2001). Accessed 12 May 2011, April–June
26. United States Census Bureau: 2010 Census TIGER/Line Shapefiles. http://www.census.gov/geo/www/tiger/tgrshp2010/tgrshp2010.html (2011). Accessed 29 Dec 2011
27. United States Census Bureau: Tallies of census blocks by state or state equivalent. http://www.census.gov/geo/www/2010census/census_block_tally.html (2011). Accessed 27 May 2011
28. Webster, G.R.: Reflections on current criteria to evaluate redistricting plans. Polit. Geogr. **32**, 3–14 (2013)
29. Yamada, T.: A mini-max spanning forest approach to the political districting problem. Int. J. Syst. Sci. **40**(5), 471–477 (2009)