



## Operations Research

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### Imposing Contiguity Constraints in Political Districting Models

Hamidreza Validi, Austin Buchanan, Eugene Lykhovyd

#### To cite this article:

Hamidreza Validi, Austin Buchanan, Eugene Lykhovyd (2022) Imposing Contiguity Constraints in Political Districting Models. *Operations Research* 70(2):867-892. <https://doi.org/10.1287/opre.2021.2141>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2021, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

## Crosscutting Areas

## Imposing Contiguity Constraints in Political Districting Models

Hamidreza Validi,<sup>a</sup> Austin Buchanan,<sup>b</sup> Eugene Lykhovyd<sup>c</sup><sup>a</sup>Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005; <sup>b</sup>School of Industrial Engineering & Management, Oklahoma State University, Stillwater, Oklahoma 74078; <sup>c</sup>Department of Industrial and Systems Engineering, Texas A&M University, College Station, Texas 77843Contact: hamidreza.validi@rice.edu,  <https://orcid.org/0000-0002-7983-7262> (HV); buchanan@okstate.edu,  <https://orcid.org/0000-0003-2999-9666> (AB); lykhovyd@tamu.edu,  <https://orcid.org/0000-0001-9679-1537> (EL)

Received: January 27, 2020

Revised: July 24, 2020; October 27, 2020

Accepted: April 6, 2021

Published Online in Articles in Advance:  
December 28, 2021Area of Review: Policy Modeling and Public  
Sector OR<https://doi.org/10.1287/opre.2021.2141>

Copyright: © 2021 INFORMS

**Abstract.** Beginning in the 1960s, techniques from operations research began to be used to generate political districting plans. A classical example is the integer programming model of Hess et al. [Hess SW, Weaver JB, Siegfeldt HJ, Whelan JN, Zitlau PA (1965) *Oper. Res.* 13(6):998–1006.]. Because of the model's compactness-seeking objective, it tends to generate contiguous or nearly contiguous districts, although none of the model's constraints explicitly impose contiguity. Consequently, Hess et al. had to manually adjust their solutions to make them contiguous. Since then, there have been several attempts to adjust the Hess model and other models so that contiguity is explicitly ensured. In this paper, we review two existing models for imposing contiguity, propose two new ones, and analytically compare them in terms of their strength and size. We conduct an extensive set of numerical experiments to evaluate their performance. Although many believe that contiguity constraints are particularly difficult to deal with, we find that the districting problem considered by Hess et al. does not become harder when contiguity is imposed. In fact, a branch-and-cut implementation of a cut-based model generates, for the first time, optimally compact districting plans for 21 different U.S. states at the census tract level. To encourage future research in this area, and for purposes of transparency, we make our test instances and source code publicly available.

**Funding:** This material is based on work supported by the National Science Foundation [Grants 1662757 and 1942065]. Some of the computing for this project was performed at the OSU High Performance Computing Center at Oklahoma State University supported in part through the National Science Foundation [Grant 1126330].

**Keywords:** political redistricting • contiguity • connectivity • integer programming • branch-and-cut • Lagrangian • moment-of-inertia

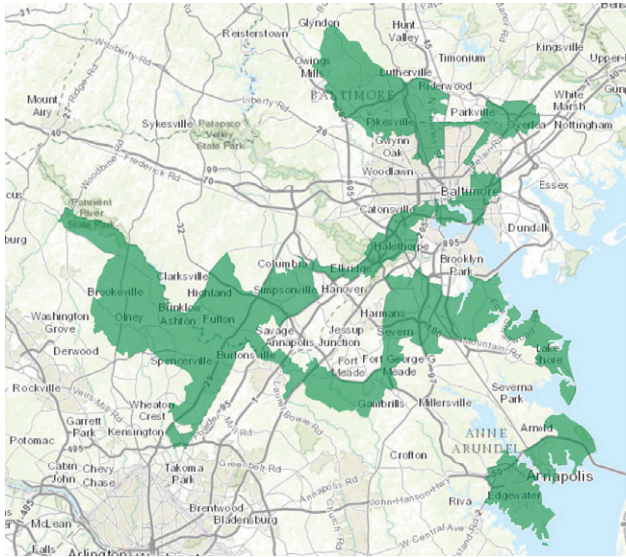
## 1. Introduction

In the United States, congressional redistricting occurs every 10 years, soon after the census has been taken and the number of representatives for each state has been determined based on their populations (a process called reapportionment). A redistricting plan, which specifies how the district lines will be drawn within a U.S. state, must satisfy certain constraints. Two typical constraints are that (i) each district must be contiguous, and (ii) each district must have the “same” population. There are other properties that redistricting plans must satisfy (e.g., the Voting Rights Act prohibits racial gerrymandering), but they are often not as clear cut as contiguity and population equality or may vary by state (e.g., some states require the preservation of political subdivisions like counties).

Many redistricting plans will satisfy the contiguity and population equality constraints, allowing redistricters to optimize a particular objective or to satisfy a set of additional constraints. This opens the door for state legislatures (who often control redistricting) to

leverage the process to their benefit, resulting in partisan or incumbent gerrymanders. Indeed, gerrymandering has become so contentious since the 2010 Census that it has led to several cases before the Supreme Court of the United States. These cases include *Lamone v. Benisek* regarding Democratic gerrymandering in Maryland, *Rucho v. Common Cause* regarding Republican gerrymandering in North Carolina, and *Abbott v. Perez* regarding racial gerrymandering in Texas. Preliminary evidence of a gerrymander often includes disproportionate electoral outcomes (e.g., in 2016, Republicans won 77% of North Carolina's congressional seats despite winning only 53% of the votes) or unusually shaped, noncompact districts (e.g., Maryland's third congressional district, depicted in Figure 1, was drawn by the state's Democrats and has been compared with a “broken-winged pterodactyl, lying prostrate across the center of the state” by a federal judge (Linskey 2012)). However, neither a disproportionate outcome nor an unusually shaped district is a telltale sign of gerrymandering (Duchin et al. 2019). In June

**Figure 1.** (Color online) Maryland's Third Congressional District Following the 2010 Census



*Note.* Operations researchers might be interested to know that the INFORMS offices sit just outside of this district.

2019, the Supreme Court decided that partisan gerrymandering falls outside the purview of the federal courts, unleashing a new era of gerrymandering.

In an effort to de-politicize the redistricting process, some have suggested that “redistricting should be a bureaucratic, boring process where you get the census data, you turn the crank, and you get new maps for the next decade” (Boehm 2018). Others have commented unfavorably on automated redistricting because (1) reasonable people can disagree about what properties the best redistricting plan should satisfy, and (2) even if there were universal agreement on the desiderata, the resulting problem would almost certainly be NP-hard (Altman and McDonald 2010). Nevertheless, the show must go on. Some redistricting plan must be chosen, and computers will inevitably be used in its creation, whether to provide starting points for discussion, to refine preliminary plans, or to understand the limits of what is possible. In any case, imposing contiguity will be important; 23 states require contiguity by law, and the others almost always practice it anyway (Altman 1998, Duchin et al. 2019).

Researchers have struggled to handle the contiguity constraints inherent in redistricting problems, despite a long history. Hess et al. (1965) proposed perhaps the first optimization model for redistricting. It sought an optimally compact redistricting plan, where compactness was measured in terms of a moment-of-inertia objective, subject to constraints on population equality. Because of the model’s compactness-seeking objective, it tends to generate contiguous or nearly contiguous districts, although none of the model’s

constraints explicitly impose contiguity. Consequently, Hess et al. manually adjusted their solutions to make them contiguous.

Since then, there have been several attempts to adjust the Hess model (or others similar to it) so that contiguity is explicitly ensured (Zoltners and Sinha 1983, Drexler and Haase 1999, Caro et al. 2004, Shirabe 2009, Duque et al. 2011, Kim and Xiao 2017, Oehrlein and Haunert 2017). Nevertheless, researchers have remained pessimistic.

- “[Contiguity] constraints make [districting] much more difficult than other partitioning problems in combinatorial optimization, such as coloring or frequency assignment” (Ricca and Simeone 2008, p. 1412).

- “[Contiguity] is particularly difficult to deal with and, sometimes, it is even discarded from [political districting] models and considered only a posteriori” (Ricca et al. 2013, p. 272).

- “Ensuring contiguity efficiently seems to be an issue in exact methods [for political districting]” (Goderbauer and Winandy 2018, p. 34).

- “For exact methods, contiguity enforcement has been a major challenge” (Swamy et al. 2019b, p. 4).

With this in mind, this paper studies how to best impose contiguity in the context of the Hess model. We consider the following models:

1. SHIR, a flow-based model credited to Shirabe (2005, 2009) and detailed by Oehrlein and Haunert (2017);

2. CUT, a cut-based model proposed by Oehrlein and Haunert (2017) that draws from Carvajal et al. (2013) and others;

3. MCF, a new flow-based model that we show is equivalent in strength to CUT and stronger than SHIR at the cost of having more variables; and

4. LCUT, a new cut-based model that is shown to be stronger than the other models.

We also examine the separation problems associated with the CUT and LCUT models; the former is shown to be solvable in time  $O(n^2 \log^3 n)$ , whereas the procedure used by Oehrlein and Haunert (2017) takes time  $O(n^4)$ . Here,  $n$  is the number of vertices in a graph associated with the instance.

To evaluate the performance of the four models, we conduct a thorough set of computational experiments, testing the four models on redistricting instances for every U.S. state at the county and census tract levels. In a nod to Hess et al. (1965), we use the original moment-of-inertia objective function in our experiments. We find that the new LCUT model is the best-performing formulation on the county-level instances where the problem has more of a combinatorial flavor. On the tract-level instances, which have significantly more granularity, CUT and LCUT perform nearly the same and solve 21 instances to optimality.

The largest instance that CUT solves is for Indiana. This instance has 1,511 census tracts and uses  $(1,511)^2 = 2,283,121$  binary variables. To our knowledge, this is significantly larger than any districting instance ever solved in the literature by an exact method—with or without contiguity constraints. For example, Mehrotra et al. (1998) use branch-and-price for South Carolina, essentially at the county level, having approximately 50 vertices and six districts. Several of their steps were not automated, including the splitting and joining of several counties presolve, and manual adjustments postsolve for population equality and were ultimately unable to guarantee optimality. A more recent paper by Swamy et al. (2019b) heuristically reduces the sizes of instances by a series of graph contractions until  $n \leq 200$  ( $n^2 \leq 40,000$ ) at which point they are able to deploy their exact method. They motivate this graph contraction procedure by noting that Wisconsin has 1,409 tracts and by pointing out the enormous size of the resulting mixed integer program (MIP). This instance can also be solved with our techniques if the original objective function of Hess et al. is used.

To encourage future research in this area, and for purposes of transparency, we make our test instances, C++ source code, and redistricting plans (maps and block equivalency files) available at <https://github.com/zhelih/districting>. The source code is released under a GNU General Public License, which gives users the ability to run, study, share, and modify it.

## 2. Background and Literature Review

In this section, we give a brief overview of redistricting, particularly congressional redistricting for the United States, and approaches for constructing redistricting plans. The literature on these topics is vast, and we can only cover the highlights. Interested readers are encouraged to refer to Di Cortona et al. (1999), Murphy et al. (2013), Ricca et al. (2013), and Goderbauer and Winandy (2018) for perspectives on redistricting from operations researchers and Grofman (1985), Arrington (2010), and Bullock (2010) for perspectives from social scientists.

### 2.1. Redistricting Principles and Laws in the United States

A redistricting plan must satisfy certain state and federal laws. These laws are often crafted to ensure that *traditional redistricting principles* are followed (e.g., population equality, contiguity, compactness, preservation of political subdivisions, and communities of interest) or that the districting process does not disadvantage a particular group (e.g., a racial minority or members of a political party). We mention some examples of redistricting laws in the United States. Our intent is to provide some context for the stylized

redistricting problem that we will consider in this paper while also recognizing that the districting plans from our computational experiments will not consider all of these redistricting principles or laws (which also vary by state).

**2.1.1. Population Balance.** Federal laws in the United States require that congressional districts within a state all have the same population. This *one-person, one-vote* principle was formally interpreted by the Supreme Court to be a consequence of Article I, Section 2 of the U.S. Constitution in the 1964 case *Reynolds v. Sims*. In the decades following *Reynolds*, the courts established tighter and tighter restrictions on how much population deviation is allowed. There is currently no threshold for population deviation beyond which a districting plan will necessarily be deemed legal (i.e., a safe harbor), and a population deviation of just 19 *people* (0.0029%!) was ruled unconstitutional in 2002 by a federal district court in Pennsylvania (Hebert et al. 2010). Nevertheless, larger population deviations of up to 1% have been allowed if there is a compelling justification, such as the desire to satisfy a traditional redistricting principle. For example, West Virginia kept all its counties intact at the price of a 0.79% population deviation (NCSL 2019).

**2.1.2. Race.** Federal law also dictates what role race should (or should not) play in redistricting. For example, Section 2 of the Voting Rights Act (VRA) prohibits racial gerrymandering, disallowing any practice or procedure that inhibits a protected minority group from electing candidates of their choice. In the 1986 case *Thornburg v. Gingles*, the Supreme Court established when states must create “majority-minority” or minority-opportunity districts with the three-pronged *Gingles* test, the first prong of which requires that the minority group be sufficiently numerous and geographically compact. There are also *constitutional* limits on racial gerrymandering. For example, in the 1993 case *Shaw v. Reno*, the Supreme Court established that the Equal Protection Clause of the 14th Amendment prohibits states from the excessive or unjustified use of race when redistricting, especially if race predominates the map-making process to the exclusion of traditional redistricting principles (Hebert et al. 2010).

**2.1.3. State Laws (e.g., Contiguity).** States also enact laws regarding congressional redistricting. For example, contiguity is not federally required, so 23 states have imposed this requirement themselves; the other states almost always enact contiguous districts anyway (Duchin et al. 2019). States such as Iowa have additional laws regarding compactness, the preservation of political subdivisions, and the nonuse of partisan data (NCSL 2019).

## 2.2. Algorithms and Models for Redistricting

Any practical variant of redistricting is NP-hard (Altman 1997), leading many researchers to propose their own heuristics (Ricca et al. 2013). A nonexhaustive list of examples include greedy construction heuristics (Vickrey 1961, Kim 2019), local search heuristics (King et al. 2012, 2015, 2018), metaheuristics like simulated annealing and tabu search (Bozkaya et al. 2003, Ricca and Simeone 2008, Altman and McDonald 2011, Guo and Jin 2011, Liu et al. 2016, Olson 2019, Gutiérrez-Andrade et al. 2019), and generalizations of Voronoi diagrams (Miller 2007, Svec et al. 2007, Ricca et al. 2008, Cohen-Addad et al. 2018, Levin and Friedler 2019).

Recently, several researchers proposed Markov chain Monte Carlo (MCMC) methods for generating large collections of redistricting plans, where the aim is understand the *distribution* of redistricting plans, which can provide a baseline with which to compare proposed or implemented plans (Fifield et al. 2015, Cho and Liu 2018, Adler and Wang 2019, DeFord et al. 2021). If a proposed redistricting plan is an outlier (say, with respect to seat share distribution), this might suggest an intent to gerrymander. MCMC sampling methods for redistricting are quite similar to local search in that they move from one feasible solution to a neighboring feasible solution.

The literature also contains many exact methods for redistricting, including numerous IP formulations (Ricca et al. 2013). Perhaps the two most notable are what we will call the *Hess model* and the *set partition model*.

The Hess model, detailed in Section 2.4, is a constrained  $k$ -median model. Its essence can be found in numerous papers on redistricting in the operations research (OR) literature (Hess et al. 1965, Hojati 1996, Gentry et al. 2015). The most popular technique for imposing contiguity in the context of the Hess model is a flow-based formulation credited to Shirabe (2005, 2009) (cf. Oehrlein and Haunert 2017). This formulation, which is detailed in Section 3.1, is easy to implement and has been used in one form or another in several papers (Duque et al. 2011, Gopalan et al. 2013, Haase and Müller 2014, Oehrlein and Haunert 2017, Kong et al. 2019, Swamy et al. 2019b). Another notable approach for imposing contiguity uses graph cuts. Specifically, Oehrlein and Haunert (2017) propose to use  $a, b$ -separator inequalities for redistricting (detailed in Section 3.3); others have proposed related inequalities that are weaker (Drexler and Haase 1999) or not valid (Zoltners and Sinha 1983).

In the set partition model, there is a binary variable for each possible district, and the task is to select  $k$  of them such that every part of the state is covered exactly once. The approaches of Garfinkel and Nemhauser (1970) and Mehrotra et al. (1998) are based on formulations of this type. It should be noted that, in general, the set of possible districts grows exponentially, meaning that formulations of this type are solved

either using a select subset of district variables or the variables are introduced on-the-fly via column generation. Contiguity is handled during pricing.

## 2.3. Notation and Problem Definition

When trying to impose the contiguity constraints involved in political districting, it is helpful to use the so-called *contiguity graph* (Ricca et al. 2013), also known as the adjacency graph or dual graph. In this graph  $G = (V, E)$ , each vertex  $v \in V$  represents a contiguous parcel of land (e.g., a county or census tract), and there is an edge  $\{u, v\} \in E$  connecting vertices  $u$  and  $v$  when the corresponding land parcels share a border of non-zero length (e.g., it is not enough to meet at a point). By this construction,  $G$  will be simple and planar, and thus sparse. Indeed, its number of edges  $m := |E|$  is linear with respect to the number of vertices  $n := |V|$ ; Euler's polyhedral formula implies that  $m \leq 3n - 6$  when  $n \geq 3$ . (In our experiments, we will come across a handful of cases where a land parcel consists of multiple disconnected pieces, e.g., a few lakes that form a single census tract, which can result in a nonplanar contiguity graph; however, all of the instances in our experiments will nevertheless be sparse and satisfy  $m \leq 3n - 6$ .) The (open) neighborhood of vertex  $i$  is denoted by  $N(i) := \{j \in V \mid \{i, j\} \in E\}$ , and the closed neighborhood is denoted by  $N[i] := N(i) \cup \{i\}$ . Other necessary data include the following:

- The number  $k$  of districts to be created;
- The population  $p_v$  of each land parcel  $v \in V$ ; and
- The minimum and maximum population ( $L$  and  $U$ ) allowed in a district.

Another piece of data that might be used to construct a “compact” districting plan is the distance  $d_{ij}$  between (the centers of) land parcels  $i$  and  $j$ . Indeed, these distances appear in the compactness-seeking objective function in the model of Hess et al. (1965).

For the purposes of this paper, the districting problem is defined as follows. If any exist, find a partition of the vertex set  $V$  into *districts* such that

1. Each vertex belongs to exactly one district;
2. There are  $k$  districts;
3. Each district  $V'$  satisfies the population bounds, that is,  $L \leq \sum_{i \in V'} p_i \leq U$ ; and
4. Each district  $V'$  is contiguous, that is,  $G[V']$  is connected.

We call these four constraints the *bright-line rules*, and any districting plan that satisfies them is said to be *feasible*. If there are multiple feasible solutions, the task is to find one that is most compact with respect to the penalties  $w$ , which are discussed next.

## 2.4. The Hess Model

Hess et al. (1965) introduced *the* classical integer program for political districting, which uses the following  $n^2$  binary variables:

$$x_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is assigned to (the district centered at) vertex } j \\ 0 & \text{otherwise.} \end{cases}$$

The Hess formulation is as follows, where  $w_{ij}$  is a penalty charged for assigning  $i$  to  $j$ .

$$\min \sum_{i \in V} \sum_{j \in V} w_{ij} x_{ij} \quad (1a)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \quad (1b)$$

$$\sum_{j \in V} x_{jj} = k \quad (1c)$$

$$Lx_{jj} \leq \sum_{i \in V} p_i x_{ij} \leq Ux_{jj} \quad \forall j \in V \quad (1d)$$

$$x_{ij} \leq x_{jj} \quad \forall ij \in V \quad (1e)$$

$$x_{ij} \in \{0,1\} \quad \forall ij \in V \quad (1f)$$

Constraints (1b) ensure that each vertex is assigned to a district. Constraint (1c) ensures that  $k$  districts are chosen. Constraints (1d) ensure that the population of each district lies between  $L$  and  $U$ . Constraints (1e) were not originally included by Hess et al. (1965) but are usually added for strength (Ricca et al. 2013). In our code, we also introduce a new variable to replace the expression  $\sum_{i \in V} p_i x_{ij}$  in Constraints (1d), thus reducing the formulation's size in computer memory by 20%.

Hess et al. (1965) considered a *moment-of-inertia* objective, defining the penalties  $w_{ij}$  as follows:

$$(\text{penalty for moment-of-inertia objective}) \quad w_{ij} := p_i d_{ij}^2.$$

Meanwhile, the traditional objective used in facility location would capture the total number of miles traveled if the state's inhabitants were to drive to their district centers with the penalties  $w_{ij} := p_i d_{ij}$ , where  $d$  is set using road distances (Daskin and Tucker 2018). Duchin (2018) has humorously described this measure of compactness as an "appealing one, particularly if you imagine replacing distance with travel time" because you can think of it as answering the question: "How long does it take you to go yell at your representative?" Another objective that has been considered by Hojati (1996) and Gopalan et al. (2013) uses squared Euclidean distances  $w_{ij} := d_{ij}^2$ . Others such as Swamy et al. (2019a) and Mehrotra et al. (1998) use the simpler penalty  $w_{ij} := d_{ij}$ . In fact, Mehrotra et al. (1998) take  $d_{ij}$  as the hop-based distance between  $i$  and  $j$  in the contiguity graph. In a nod to Hess et al. (1965), we use the original moment-of-inertia objective in our experiments; however, our analysis and techniques apply regardless of which penalties are used. For more information about the many measures of compactness in the literature, we refer the reader to the compactness critiques of Young (1988) and the classification by Niemi et al. (1990).

The Hess model is the foundation of our proposed formulations. The feasible region of its linear programming

(LP) relaxation, which we denote by  $\mathcal{P}_{\text{HESS}}$ , is defined as follows:

$$\mathcal{P}_{\text{HESS}} := \{x \in \mathbb{R}_+^{n \times n} \mid x \text{ satisfies constraints (1b), (1c), (1d), (1e)}\}.$$

The bounds  $x_{ij} \leq 1$  are implied by nonnegativity and the assignment Constraints (1b).

### 3. Contiguity Models

Here we consider four different models for imposing contiguity in the context of the Hess model: SHIR, CUT, MCF, and LCUT. Two of them have appeared in the previous literature (SHIR and CUT), whereas two others are new (MCF and LCUT). The flow-based models SHIR and MCF refer to the "bidirected" version of the contiguity graph that we denote by  $D = (V, A)$ . This directed graph  $D$  is obtained from  $G = (V, E)$  by replacing each undirected edge  $\{i, j\} \in E$  by its directed counterparts  $(i, j)$  and  $(j, i)$ . Thus,  $|A| = 2|E|$ . The set of edges pointing away from vertex  $i$  is denoted by  $\delta^+(i)$ , and the set of edges pointing toward vertex  $j$  is denoted by  $\delta^-(j)$ .

#### 3.1. SHIR

Oehrlein and Haurert (2017) propose a flow-based formulation, adapted from Shirabe (2009). We provide (essentially) the same formulation below and call it SHIR. This formulation uses the following flow variables:

$$f_{ij}^v = \text{the amount of flow, originating at district center } v, \text{ that is sent across edge } (i, j).$$

The SHIR formulation is as follows, where  $f^j(S)$  for  $S \subseteq A$  is shorthand for  $\sum_{(u,v) \in S} f_{uv}^j$ .

$$x \in \mathcal{P}_{\text{HESS}} \quad (2a)$$

$$f^j(\delta^-(i)) - f^j(\delta^+(i)) = x_{ij} \quad \forall i \in V \setminus \{j\}, \quad \forall j \in V \quad (2b)$$

$$f^j(\delta^-(i)) \leq (n-1)x_{ij} \quad \forall i \in V \setminus \{j\}, \quad \forall j \in V \quad (2c)$$

$$f^j(\delta^-(j)) = 0 \quad \forall j \in V \quad (2d)$$

$$f_{ij}^v \geq 0 \quad \forall (i, j) \in A, \quad \forall v \in V \quad (2e)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V. \quad (2f)$$

Constraints (2b) ensure that if vertex  $i$  is assigned to center  $j$ , then  $i$  consumes one unit of flow of type  $j$ ; otherwise, it consumes none. Constraints (2c) ensure that vertex  $i$  can receive flow of type  $j$  only if  $i$  is assigned to center  $j$ . Constraints (2d) prevent flow circulations.

Now, we define the polytope  $\mathcal{P}_{\text{SHIR}}$  as follows:

$$\mathcal{P}_{\text{SHIR}} := \{(x, f) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{2mn} \mid (x, f) \text{ satisfies Constraints (2a)–(2e)}\}.$$

**Remark 1.** The following equations are implied in  $\mathcal{P}_{\text{SHIR}}$ .

$$f^j(\delta^+(j)) = \sum_{i \in V \setminus \{j\}} x_{ij} \quad \forall j \in V. \quad (3)$$

**Proof.** Consider a point  $(\hat{x}, \hat{f})$  that belongs to  $\mathcal{P}_{\text{SHIR}}$ . Then, for every vertex  $j \in V$ ,

$$\hat{f}^j(\delta^+(j)) = \hat{f}^j(\delta^+(j)) - \hat{f}^j(\delta^-(j)), \tag{4a}$$

$$= \hat{f}^j(\delta^+(j)) - \hat{f}^j(\delta^-(j)) - \sum_{i \in V} (\hat{f}^j(\delta^+(i)) - \hat{f}^j(\delta^-(i))), \tag{4b}$$

$$= - \sum_{i \in V \setminus \{j\}} (\hat{f}^j(\delta^+(i)) - \hat{f}^j(\delta^-(i))), \tag{4c}$$

$$= \sum_{i \in V \setminus \{j\}} (\hat{f}^j(\delta^-(i)) - \hat{f}^j(\delta^+(i))) = \sum_{i \in V \setminus \{j\}} \hat{x}_{ij}. \tag{4d}$$

Here, Equation (4a) holds by Constraints (2d), Equation (4b) holds because the flow variables in the summation cancel each other, and Equation (4d) holds by Constraints (2b). □

### 3.2. MCF

The SHIR formulation uses “big-M” Constraints (2c), which can result in a weak linear programming relaxation. With this in mind, we propose a different flow-based formulation that avoids the big-M constraints via a different variable definition:

$$f_{ij}^{ab} = \begin{cases} 1 & \text{if edge } (i, j) \in A \text{ is on the path to vertex } \\ & a \text{ from its district's center } b \\ 0 & \text{otherwise.} \end{cases}$$

The resulting MCF formulation, which can be viewed as a disaggregation of SHIR, is as follows, where  $f^{ab}(S)$  for  $S \subseteq A$  is shorthand for  $\sum_{(u,v) \in S} f_{uv}^{ab}$ .

$$x \in \mathcal{P}_{\text{HESS}} \tag{5a}$$

$$f^{ab}(\delta^+(b)) - f^{ab}(\delta^-(b)) = x_{ab} \quad \forall a \in V \setminus \{b\}, \forall b \in V \tag{5b}$$

$$f^{ab}(\delta^+(i)) - f^{ab}(\delta^-(i)) = 0 \quad \forall i \in V \setminus \{a, b\}, \forall a \in V \setminus \{b\} \quad \forall b \in V \tag{5c}$$

$$f^{ab}(\delta^-(b)) = 0 \quad \forall a \in V \setminus \{b\}, \forall b \in V \tag{5d}$$

$$f^{ab}(\delta^-(j)) \leq x_{jb} \quad \forall j \in V \setminus \{b\}, \forall a \in V \setminus \{b\}, \forall b \in V \tag{5e}$$

$$f_{ij}^{ab} \geq 0 \quad \forall (i, j) \in A, \forall a \in V \setminus \{b\}, \quad \forall b \in V \tag{5f}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V. \tag{5g}$$

Constraints (5b) ensure that if vertex  $a$  is assigned to center  $b$ , then  $b$  sends out a unit of flow of type  $ab$ , which ultimately must be consumed by  $a$  because of the flow conservation Constraints (5c) at other vertices. Constraints (5d) prevent flow circulations. Constraints (5e) ensure that vertex  $j$  can receive a flow of type  $ab$  only if  $j$  is assigned to center  $b$ .

To our knowledge, this formulation is new in the context of districting. Now, we define the polytope  $\mathcal{P}_{\text{MCF}}$  as follows:

$$\mathcal{P}_{\text{MCF}} := \left\{ (x, f) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{2mn(n-1)} \mid (x, f) \text{ satisfies constraints (5a)–(5f)} \right\}.$$

The bounds  $f_{ij}^{ab} \leq 1$  are implied in  $\mathcal{P}_{\text{MCF}}$  by nonnegativity and Constraints (5e).

**Remark 2.** The following equations are implied in  $\mathcal{P}_{\text{MCF}}$ :

$$f^{ab}(\delta^+(a)) - f^{ab}(\delta^-(a)) = -x_{ab} \quad \forall a \in V \setminus \{b\}, \quad \forall b \in V \tag{6a}$$

$$f^{ab}(\delta^+(a)) = 0 \quad \forall a \in V \setminus \{b\}, \quad \forall b \in V. \tag{6b}$$

**Proof.** Consider a point  $(\hat{x}, \hat{f})$  that belongs to  $\mathcal{P}_{\text{MCF}}$ . For distinct vertices  $a, b \in V$ ,

$$\begin{aligned} & \hat{f}^{ab}(\delta^+(a)) - \hat{f}^{ab}(\delta^-(a)) \\ &= \hat{f}^{ab}(\delta^+(a)) - \hat{f}^{ab}(\delta^-(a)) - \sum_{j \in V} (\hat{f}^{ab}(\delta^+(j)) - \hat{f}^{ab}(\delta^-(j))) \end{aligned} \tag{7a}$$

$$= - \sum_{j \in V \setminus \{a\}} (\hat{f}^{ab}(\delta^+(j)) - \hat{f}^{ab}(\delta^-(j))) \tag{7b}$$

$$= -(\hat{f}^{ab}(\delta^+(b)) - \hat{f}^{ab}(\delta^-(b))) = -\hat{x}_{ab}. \tag{7c}$$

Here, Equation (7a) holds because the flow variables in the summation cancel each other. The first equation in Line (7c) holds by Constraints (5c) and the second holds by Constraints (5b). Therefore,, the point  $(\hat{x}, \hat{f})$  satisfies Equations (6a). Furthermore,

$$0 \leq \hat{f}^{ab}(\delta^+(a)) = -\hat{x}_{ab} + \hat{f}^{ab}(\delta^-(a)) \leq -\hat{x}_{ab} + \hat{x}_{ab} = 0.$$

Here, the first equation holds by the implied Equations (6a), and the last inequality holds by Constraints (5e). Thus, Equations (6b) are implied in  $\mathcal{P}_{\text{MCF}}$ . □

### 3.3. CUT

The CUT formulation, which was first used for districting by Oehrlin and Haunert (2017), is based on the concept of  $a, b$ -separators (Carvajal et al. 2013, Ahn and Park 2015, Buchanan et al. 2015, Fischetti et al. 2017, Wang et al. 2017). An example is given in Figure 2.

**Definition 1** ( $a, b$ -Separator). A subset  $C \subseteq V \setminus \{a, b\}$  of vertices is called an  $a, b$ -separator for  $G = (V, E)$  if there is no path from  $a$  to  $b$  in  $G - C$ .

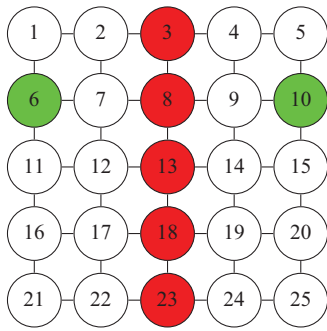
The resulting  $a, b$ -separator inequalities (8b) are written for every ordered pair  $(a, b)$  of nonadjacent vertices and every  $a, b$ -separator  $C$ , which we denote by the shorthand  $\forall (a, b, C)$ . As is usual, it is sufficient to consider *minimal*  $a, b$ -separators, where minimality is taken by inclusion.

$$x \in \mathcal{P}_{\text{HESS}}, \tag{8a}$$

$$x_{ab} \leq \sum_{c \in C} x_{cb} \quad \forall (a, b, C) \tag{8b}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \tag{8c}$$

**Figure 2.** (Color online) Example of an  $a,b$ -Separator



Note.  $a = 6$  and  $b = 10$ , and  $C = \{3, 8, 13, 18, 23\}$

Constraints (8b) ensure that if vertex  $a$  is assigned to center  $b$ , then at least one vertex from every  $a,b$ -separator  $C$  is assigned to  $b$ . The inequality associated with the  $a,b$ -separator from Figure 2 is  $x_{6,10} \leq x_{3,10} + x_{8,10} + x_{13,10} + x_{18,10} + x_{23,10}$ .

Now, we define the polytope  $\mathcal{P}_{\text{CUT}}$  as follows:

$$\mathcal{P}_{\text{CUT}} := \{x \in \mathbb{R}^{n \times n} \mid x \text{ satisfies constraints (8a)–(8b)}\}.$$

### 3.4. LCUT

The  $a, b$ -separator Inequalities (8b) use the fact that the vertices assigned to vertex  $b$  are required to induce a connected subgraph. By exploiting additional information (e.g., that the population bound  $U$  must also be satisfied), we can write stronger inequalities.

We formalize this using the concept of a length- $U$   $a,b$ -separator. To do this, we need to refer to vertex-weighted distances, where the weight of a vertex  $i$  is its population  $p_i$ . The distance  $\text{dist}_{G,p}(a,b)$  from  $a$  to  $b$  is the length  $\sum_{v \in V(P)} p_v$  of a shortest vertex-weighted path  $P$  from  $a$  to  $b$ , where  $V(P)$  denotes the set of vertices on path  $P$ . An example is given in Figure 3.

**Definition 2** (Length- $U$   $a, b$ -Separator). A subset  $C \subseteq V \setminus \{a, b\}$  of vertices is called a length- $U$   $a,b$ -separator in  $G = (V, E)$ , with respect to vertex weights  $p$ , if  $\text{dist}_{G-C,p}(a, b) > U$ .

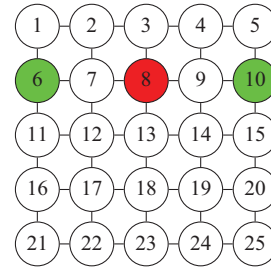
With this definition, we can write inequalities having the exact same form as the  $a,b$ -separator Inequalities (8b), except that  $C$  will now be a length- $U$   $a, b$ -separator. We write these inequalities for every ordered pair  $(a, b)$  of *distinct* (possibly adjacent) vertices and every (minimal) length- $U$   $a, b$ -separator  $C$ , which we again denote by the shorthand  $\forall (a, b, C)$ .

$$x \in \mathcal{P}_{\text{HESS}}, \tag{9a}$$

$$x_{ab} \leq \sum_{c \in C} x_{cb} \quad \forall (a, b, C), \tag{9b}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V. \tag{9c}$$

**Figure 3.** (Color online) Example of a Length- $U$   $a,b$ -Separator



Note.  $p = 1$  and  $U = 5$ , Where  $a = 6$  and  $b = 10$ , and  $C = \{8\}$

Constraints (9b) ensure that if vertex  $a$  is assigned to center  $b$ , then at least one vertex from every length- $U$   $a,b$ -separator  $C$  is assigned to  $b$ . The inequality associated with the length- $U$   $a,b$ -separator from Figure 3 is  $x_{6,10} \leq x_{8,10}$ .

Similar length-bounded cut models have been proposed recently for other problems (Salemi and Buchanan 2020; Validi and Buchanan 2020; Arslan et al. 2019, 2020).

**Remark 3.** If  $\text{dist}_{G,p}(a, b) > U$ , then  $C = \emptyset$  is a length- $U$   $a,b$ -separator and  $x_{ab} \leq 0$  is the associated length- $U$   $a,b$ -separator inequality.

Now, we define polytope  $\mathcal{P}_{\text{LCUT}}$  as follows:

$$\mathcal{P}_{\text{LCUT}} := \{x \in \mathbb{R}^{n \times n} \mid x \text{ satisfies constraints (9a)–(9b)}\}.$$

## 4. Analysis of the Formulations

We now compare the strength of the four formulations, prove their correctness, and analyze the separation problems associated with the exponentially sized models CUT and LCUT.

### 4.1. Formulation Strength

The main result of this section is the following theorem. A specific result is that the newly proposed LCUT formulation is the strongest formulation in this paper.

**Theorem 1.** For every instance of districting,

$$\mathcal{P}_{\text{LCUT}} \subseteq \mathcal{P}_{\text{CUT}} = \text{proj}_x \mathcal{P}_{\text{MCF}} \subseteq \text{proj}_x \mathcal{P}_{\text{SHIR}},$$

and there exist instances for which the inclusions are strict.

**Proof.** This follows by Lemmas 1–3, which are proven later.  $\square$

**Lemma 1.** For every instance of districting,  $\mathcal{P}_{\text{LCUT}} \subseteq \mathcal{P}_{\text{CUT}}$ , and this inclusion can be strict.

**Proof.** Because  $a, b$ -separator inequalities are length- $U$   $a, b$ -separator inequalities for every  $U$ , the inclusion  $\mathcal{P}_{\text{LCUT}} \subseteq \mathcal{P}_{\text{CUT}}$  holds. Figure 4 gives an example where  $\mathcal{P}_{\text{LCUT}} \neq \mathcal{P}_{\text{CUT}}$ . Therefore, the inclusion can be strict.  $\square$

**Lemma 2.** For every instance of districting,  $\mathcal{P}_{\text{CUT}} = \text{proj}_x \mathcal{P}_{\text{MCF}}$ .

**Proof.** ( $\supseteq$ ) Suppose that  $(\hat{x}, \hat{f})$  belongs to  $\mathcal{P}_{\text{MCF}}$ . To show that  $\hat{x}$  belongs to  $\mathcal{P}_{\text{CUT}}$ , it suffices to show that  $\hat{x}$  satisfies Constraints (8b) for an arbitrary pair of nonadjacent vertices  $a, b \in V$  and  $a, b$ -separator  $C \subseteq V \setminus \{a, b\}$ . Let  $B$  the set of vertices reachable from  $b$  in the graph  $G - C$ . Then,

$$\hat{x}_{ab} = \hat{f}^{ab}(\delta^+(b)) - \hat{f}^{ab}(\delta^-(b)) \tag{10a}$$

$$= \hat{f}^{ab}(\delta^+(B)) - \hat{f}^{ab}(\delta^-(B)) \tag{10b}$$

$$\leq \hat{f}^{ab}(\delta^+(B)) \tag{10c}$$

$$\leq \hat{f}^{ab}(\delta^-(C)) \tag{10d}$$

$$\leq \sum_{j \in C} \hat{f}^{ab}(\delta^-(j)) \tag{10e}$$

$$\leq \sum_{j \in C} \hat{x}_{jb}. \tag{10f}$$

Here, Equation (10a) holds by Constraints (5b), Equation (10b) holds by Constraints (5c), and Inequalities (10c)–(10e) hold by nonnegativity of  $\hat{f}$ . Finally, Inequality (10f) holds by Constraints (5e).

( $\subseteq$ ) Suppose that  $\hat{x}$  belongs to  $\mathcal{P}_{\text{CUT}}$ . For every pair of distinct vertices  $a, b \in V$ , we define  $\hat{f}^{ab}$  as follows. If vertices  $a$  and  $b$  are adjacent, then define  $\hat{f}_{ba}^{ab} := \hat{x}_{ab}$  and  $\hat{f}_{ij}^{ab} := 0$  for all other arcs  $(i, j) \in A \setminus \{(b, a)\}$ . If  $a$  and  $b$  are nonadjacent, consider the following maximum  $b$ - $a$  flow problem in which the fixed  $\hat{x}_{jb}$  values are used as vertex capacities.

$$\max f^{ab}(\delta^+(b)) \tag{11a}$$

$$f^{ab}(\delta^+(i)) - f^{ab}(\delta^-(i)) = 0 \quad \forall i \in V \setminus \{a, b\} \tag{11b}$$

$$f^{ab}(\delta^-(b)) = 0 \tag{11c}$$

$$f^{ab}(\delta^-(j)) \leq \hat{x}_{jb} \quad \forall j \in V \setminus \{b\} \tag{11d}$$

$$f_{ij}^{ab} \geq 0 \quad \forall (i, j) \in A. \tag{11e}$$

This problem is feasible (by the zero flow), and its objective is at most one because the sink  $a$  has capacity  $\hat{x}_{ab}$  by (11d). Let  $\hat{f}_{ij}^{ab}$  for  $(i, j) \in A$  be an optimal solution to this flow problem.

Now we are to show that  $(\hat{x}, \hat{f})$  satisfies Constraints (5b)–(5f). They are easily satisfied when the constraint quantifiers  $a$  and  $b$  are adjacent, by the simple definition of  $\hat{f}^{ab}$  in this case. When the constraint quantifiers  $a$  and  $b$  are nonadjacent, Constraints (5c)–(5f) hold by the constraints defining the flow problem. Therefore, it remains to show that  $(\hat{x}, \hat{f})$  satisfies Constraints (5b) for nonadjacent vertices  $a$  and  $b$ . By classical results of Ford and Fulkerson (1962) (see section 1.11 on vertex capacities), the flow value  $\hat{f}^{ab}(\delta^+(b))$  is equal to the weight  $\sum_{j \in C} \hat{x}_{jb}$  of a minimum-weight  $b, a$ -separator  $C$ , where vertex  $j$  has weight  $\hat{x}_{jb}$ . Then,

$$\hat{x}_{ab} \leq \sum_{j \in C} \hat{x}_{jb} = \hat{f}^{ab}(\delta^+(b)) = \hat{f}^{ab}(\delta^-(a)) \leq \hat{x}_{ab}.$$

Here, the first inequality holds by assumption that  $\hat{x}$  satisfies Constraints (8b), and the last inequality holds by the capacity of vertex  $a$  in the flow problem (11d).

Then, because  $\hat{f}^{ab}(\delta^-(b)) = 0$  by the flow problem (11c), the constraint  $\hat{f}^{ab}(\delta^+(b)) - \hat{f}^{ab}(\delta^-(b)) = \hat{x}_{ab}$  holds.  $\square$

**Lemma 3.** For every instance of districting,  $\text{proj}_x \mathcal{P}_{\text{MCF}} \subseteq \text{proj}_x \mathcal{P}_{\text{SHIR}}$ , and this can be strict.

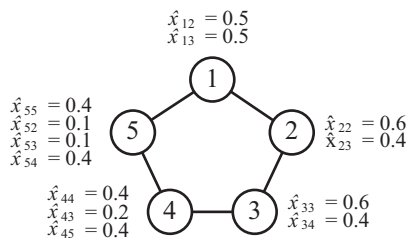
**Proof.** To prove  $\text{proj}_x \mathcal{P}_{\text{MCF}} \subseteq \text{proj}_x \mathcal{P}_{\text{SHIR}}$ , consider a point  $(\bar{x}, \bar{f})$  that belongs to  $\mathcal{P}_{\text{MCF}}$ . We construct  $\bar{f}$  such that  $(\bar{x}, \bar{f})$  belongs to  $\mathcal{P}_{\text{SHIR}}$ . For every vertex  $b \in V$  and every edge  $(i, j) \in A$ , let

$$\bar{f}_{ij}^b := \sum_{a \in V \setminus \{b\}} \hat{f}_{ij}^{ab}.$$

To show that  $(\bar{x}, \bar{f})$  satisfies Constraints (2b), consider distinct vertices  $i, b \in V$ . Then,

$$\begin{aligned} & \bar{f}^b(\delta^-(i)) - \bar{f}^b(\delta^+(i)) \\ &= \sum_{a \in V \setminus \{b\}} \hat{f}^{ab}(\delta^-(i)) - \sum_{a \in V \setminus \{b\}} \hat{f}^{ab}(\delta^+(i)) \end{aligned} \tag{12a}$$

$$\begin{aligned} &= \sum_{a \in V \setminus \{b, i\}} \left( \hat{f}^{ab}(\delta^-(i)) - \hat{f}^{ab}(\delta^+(i)) \right) \\ &+ \hat{f}^{ib}(\delta^-(i)) - \hat{f}^{ib}(\delta^+(i)) \end{aligned} \tag{12b}$$



Notes. Here,  $L = k = 2$ ,  $U = 3$ , and  $p = 1$ . Although the point  $\hat{x}$  belongs to  $\mathcal{P}_{\text{CUT}}$ , it does not belong to  $\mathcal{P}_{\text{LCUT}}$  because it violates the length-3 1, 3-separator Inequality (9b) for  $a = 1$ ,  $b = 3$ , and  $C = \{2\}$ .

$$= 0 + \hat{f}^{ib}(\delta^-(i)) - \hat{f}^{ib}(\delta^+(i)) = \bar{x}_{ib}. \quad (12c)$$

Here, Equation (12a) holds by the definition of  $\bar{f}$ , and Equations (12c) holds by Constraints (5c) and (6a).

To show that  $(\bar{x}, \bar{f})$  satisfies Constraints (2c), consider distinct vertices  $i, b \in V$ . Then,

$$\bar{f}^b(\delta^-(i)) = \sum_{a \in V \setminus \{b\}} \bar{f}^{ab}(\delta^-(i)) \leq \sum_{a \in V \setminus \{b\}} \bar{x}_{ib} = (n-1)\bar{x}_{ib}.$$

Here, the first equation holds by the definition of  $\bar{f}$ , and the inequality holds by Constraints (5e).

To show that  $(\bar{x}, \bar{f})$  satisfies Constraints (2d), consider a vertex  $j \in V$ . Then,

$$\bar{f}^j(\delta^-(j)) = \sum_{i \in V \setminus \{j\}} \bar{f}^{ij}(\delta^-(j)) = 0.$$

Here, the first equation holds by the definition of  $\bar{f}$ , and the second holds by Constraints (5d).

Figure 5 gives an example where  $\mathcal{P}_{\text{CUT}} \neq \text{proj}_x \mathcal{P}_{\text{SHIR}}$ . Because  $\mathcal{P}_{\text{CUT}} = \text{proj}_x \mathcal{P}_{\text{MCF}}$  by Lemma 2, this shows that the inclusion  $\text{proj}_x \mathcal{P}_{\text{MCF}} \subseteq \text{proj}_x \mathcal{P}_{\text{SHIR}}$  can be strict.  $\square$

### 4.2. Formulation Correctness

Here we consider the *correctness* of the formulations. By this, we mean that they allow precisely those plans that satisfy the four bright-line rules. To our

knowledge, no previous work has explicitly proven the correctness of the SHIR and CUT formulations. Although their correctness is not surprising, we feel that this step is critical to safeguard against seemingly innocuous formulations (Validi and Buchanan 2019). Appendix A provides a proof.

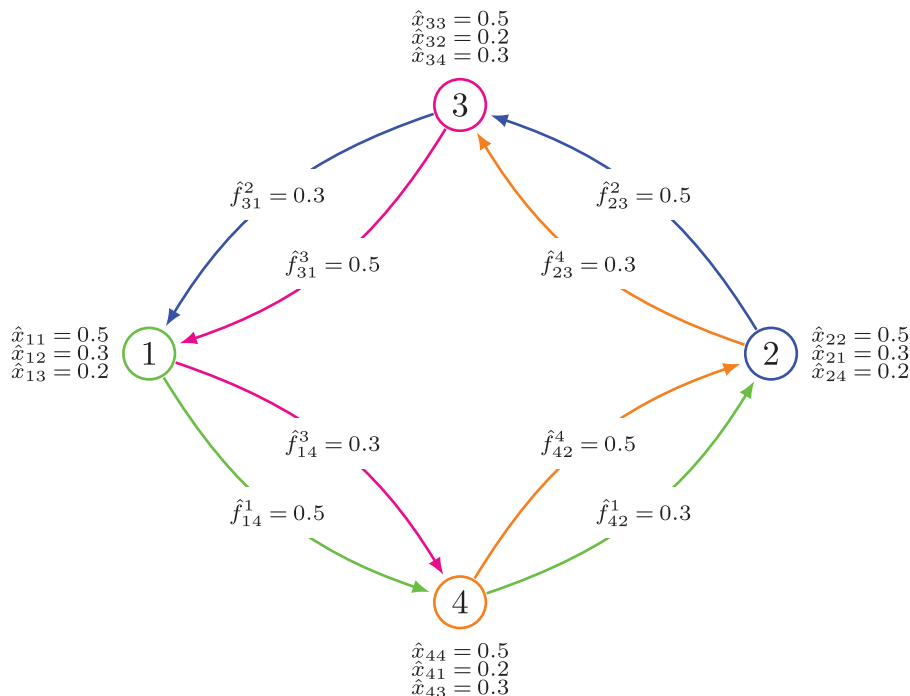
**Theorem 2.** *Formulations SHIR, MCF, CUT, and LCUT are correct.*

### 4.3. The Separation Problems

Because the models CUT and LCUT have exponential many constraints, it is important to study the associated separation problems so that inequalities can be added to the model on-the-fly as needed, instead of all up front which would render the models useless. This problem asks: given  $x^*$ , is there an inequality from the model that  $x^*$  violates? We show that this can be solved in time  $O(n^2 \log^3 n)$  for CUT, which is significantly faster than what was previously published:  $O(n^4)$  by Oehrlein and Haunert (2017). We then show that the separation problem for LCUT is NP-hard. However, we show that both separation problems can be solved in time  $O(n^2)$  when  $x^*$  is integer.

**4.3.1. Fractional Separation for CUT.** Oehrlein and Haunert propose to solve the separation problem for Inequalities (8b) as follows, when given a (fractional)

**Figure 5.** (Color online) Example Showing  $\text{proj}_x \mathcal{P}_{\text{SHIR}} \neq \mathcal{P}_{\text{CUT}}$



*Notes.* Here,  $L = U = k = 2$  and  $p = 1$ . Although the point  $(\hat{x}, \hat{f})$  belongs to  $\mathcal{P}_{\text{SHIR}}$ ,  $\hat{x}$  does not belong to  $\mathcal{P}_{\text{CUT}}$  because it violates the  $a, b$ -separator Inequality (8b) for  $a = 1, b = 2$ , and  $C = \{3, 4\}$ .

point  $\hat{x}$ . For every pair  $(a, b)$  of nonadjacent vertices, solve a minimum-weight  $a, b$ -separator problem in graph  $G$ , where vertex  $i$  has weight  $\hat{x}_{ib}$ . If this weight is less than  $\hat{x}_{ab}$ , then Inequality (8b) is violated. The minimum-weight  $a, b$ -separator problem is solved in the usual way, by a node-splitting transformation to a minimum cut problem. This procedure runs in time  $O(mn) = O(n^2)$  for a particular  $(a, b)$  pair, and in time  $O(n^4)$  overall.

**Proposition 1** (Oehrlein and Haurert 2017). *The separation problem for Constraints (8b) can be solved in time  $O(n^4)$ .*

We note that, if  $D$  is planar, the separation problem can be solved significantly faster. Namely, when given a fractional point  $\hat{x}$ , do the following for each  $b \in V$ . Start with graph  $D$  and let each vertex  $v$  have capacity  $\hat{x}_{vb}$ . Apply the linear-time reduction of Kaplan and Nussbaum (2011) to convert vertex capacities to edge capacities (while preserving planarity). Then, apply the algorithm of Lacki et al. (2012) to find the max  $b, v$ -flow value (for all  $v$ ) in time  $O(n \log^3 n)$ . This will tell us the value of a minimum-weight  $v, b$ -separator for all vertices  $v$ . Compare these flow values to  $\hat{x}_{vb}$  to see if any Inequalities (8b) are violated. Pick the most-violated one of them, say  $v = a$ , if any exist. Then apply the algorithm of Kaplan and Nussbaum (2011) to compute a max  $b, a$ -flow and from it find the associated separator  $C \subseteq V \setminus \{a, b\}$ .

**Lemma 4.** *The separation problem for Constraints (8b) for fixed  $b \in V$  can be solved in time  $O(n \log^3 n)$  when the contiguity graph is simple and planar.*

Applying Lemma 4 for each vertex  $b \in V$  gives the following proposition. We find this result quite striking given that there are  $n^2$  variables.

**Proposition 2.** *The separation problem for Constraints (8b) can be solved in time  $O(n^2 \log^3 n)$  when the contiguity graph is simple and planar.*

In some rare cases, the contiguity graph may not be planar. This is possible if the land parcels that are represented by the contiguity graph's vertices are themselves not contiguous (e.g., a census tract comprised of multiple lakes). In such situations, one may need to represent each part with its own vertex before applying the separation routines referenced here.

**4.3.2. Fractional Separation for LCUT.** We show that the separation problem for LCUT is NP-hard. Hardness persists for outerplanar graphs, which are planar graphs in which all vertices touch the outer face. The proof is provided in Appendix B.

**Theorem 3.** *The separation problem for LCUT is NP-hard, even when the graph is outerplanar and the given point  $x^*$  is known to belong to  $\mathcal{P}_{\text{CUT}}$ .*

**4.3.3. Integer Separation for CUT and LCUT.** Here we give a separation procedure for CUT and LCUT that applies when  $x^*$  is an integer. The motivation is three-fold. First, the fractional separation algorithm for CUT (Proposition 2) relies on complex algorithms for planar graphs that, to our knowledge, have never been implemented. Meanwhile, the straightforward procedure for CUT taken by Oehrlein and Haurert (2017) requires the solution of roughly  $n^2$  minimum cut problems; this price is often too expensive to justify (Fischetti et al. 2017, Salemi and Buchanan 2020, Validi and Buchanan 2020). Finally, the separation problem for LCUT is NP-hard. This motivates the following procedure that builds on one proposed by Fischetti et al. (2017).

**Algorithm 1** (IntegerSeparation( $G, p, U, x^*$ ))

```

1: for  $b \in V$  do
2:   if  $x_{bb}^* = 1$  then
3:     let  $V_b := \{i \in V \mid x_{ib}^* = 1\}$ 
4:     for every component  $G'$  of  $G[V_b]$  that does
       not contain  $b$  do
5:       let  $a$  be an arbitrary vertex of  $G'$  (e.g., one
         with the largest population)
6:       let  $C$  be the minimal  $a, b$ -separator obtained
         by Fischetti et al. (2017)
7:       if model = LCUT then
8:         for  $c \in C$  do
9:           if  $\text{dist}_{G-(C \setminus \{c\}), p}(a, b) > U$  then
10:             $C \leftarrow C \setminus \{c\}$ 
11:            add cut  $x_{ab} \leq \sum_{c \in C} x_{cb}$  to the model

```

When  $G$  is planar, Algorithm 1 runs in time  $O(n^2)$  and returns a collection of violated (minimal) separator inequalities. This time is quite modest given that the solution  $x^*$  has  $n^2$  entries. When the solution is already contiguous, the separation procedure runs in time  $O(kn)$ . Another nice property is that the cuts added on line 11 will have a combined  $O(n)$  nonzeros.

Key to our arguments is the observation that each vertex  $i$  will belong to at most  $\deg_G(i)$  many of the sets  $C$  from line 6. This follows because each minimal  $a, b$ -separator obtained from the algorithm of Fischetti et al. (2017) is a subset of the shore of  $V(G')$ , and  $i$  can belong to at most  $\deg_G(i)$  such shores (once for each of its neighbors). This implies that the total number of nonzeros coming from the round of cuts on line 11 is at most  $\sum_{i \in V} \deg_G(i) + n = 2m + n = O(n)$ . Here,  $m = O(n)$  holds for simple planar graphs by Euler's formula.

To argue for a runtime of  $O(n^2)$  when Algorithm 1 is applied to CUT, see that each vertex  $i$  can take at most one turn as the vertex  $a$  from line 5 and the fact that the algorithm of Fischetti et al. (2017) takes time  $O(m)$ , and  $m = O(n)$  because  $G$  is planar.

To argue for a runtime of  $O(n^2)$  when Algorithm 1 is applied to LCUT, it now suffices to argue that the total time spent on lines 7 through 10 (over all iterations of  $b$  and  $G'$ ) is at most  $O(n^2)$ . To see this, observe that the distance found in line 9 can be computed in time  $O(n)$  because  $G$  is planar (Henzinger et al. 1997), also recalling that the number of distance computations will be at most  $2m = O(n)$  because each vertex  $i$  will make at most  $\deg_G(i)$  appearances in the sets  $C$ . The inequalities added for LCUT will be as strong or stronger than the inequalities added for CUT. For ease of implementation, we instead use Dijkstra's shortest path algorithm, meaning that our code will take time  $O(n^2 \log n)$  instead of time  $O(n^2)$  when applied to LCUT.

## 5. Variable Fixing and a Heuristic

In initial experiments, the MIP solver had difficulties handling most tract-level instances, even when contiguity was not imposed. The large number of variables ( $n^2$ ) and the numerical instability coming from the wide range of the objective coefficients ( $w_{ij} := p_i d_{ij}^2$ ) lead to the following issues.

1. The root LP took a long time to solve with simplex; typically barrier was significantly faster.
2. If the root LP was solved with barrier, the crossover step took a very long time. The primal and dual push phases were not too costly, but the final simplex cleanup was slow—often taking ten times longer than barrier.
3. Even when an optimal basis for the root LP could be found, the MIP solver took a long time to find a good MIP-feasible solution.

To mitigate these issues, we sought to reduce the model's size by safely fixing some of the variables. By *safe*, we mean that doing so preserves an optimal solution. Intuitively, opportunities for variable fixing should be quite common. For example, if a vertex  $j$  is near the border of the state, it is likely not an optimal district center; instead, we expect the district centers to be in the state's interior. If we can rigorously argue that this is the case, we can fix the  $n$  variables  $x_{ij}$ ,  $i \in V$ , to zero. Similarly, if vertices  $i$  and  $j$  are far from each other (say, at opposite corners of a state), then we expect that we can fix  $x_{ij} = 0$ . It is also possible to safely fix some variables to one, although this occurred so infrequently that we chose not to pursue it. This is perhaps not surprising; very few variables could safely be fixed to one in the experiments of Beasley (1993) for the  $k$ -median problem when  $k$  was small.

To safely fix variables to zero, we use Lagrangian arguments. We run a heuristic to find an upper bound  $UB$  on the optimal objective value. Then, we construct a Lagrangian relaxation model that provides a lower bound  $LB_{ij}$  on the objective value when a variable  $x_{ij}$  is tentatively fixed to one. The bounds  $LB_{ij}$  can be computed almost for free while solving the Lagrangian. Now, if  $LB_{ij} > UB$ , then  $x_{ij}$  cannot equal one in an optimal solution, meaning that we can safely fix  $x_{ij} = 0$ . The Lagrangian is solved using an implementation of Shor's  $r$ -algorithm that was developed by one of the authors (Lykhovyd 2019); see also Shor (1985) and Kappel and Kuntsevich (2000). We found that the Lagrangian terminated more quickly and with a larger objective if the Lagrangian multipliers were initialized using optimal dual variables from the LP relaxation, which we obtained with barrier (no crossover). These initial multipliers can be found in the `ralg_warm` directory.

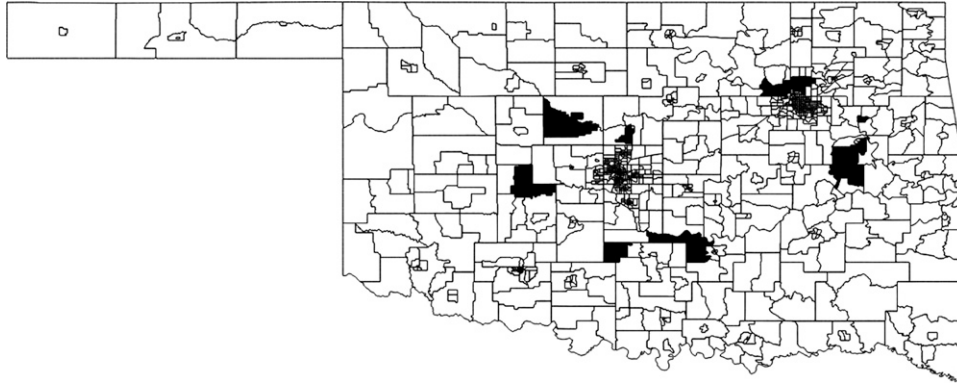
Figure 6 illustrates the power of this variable fixing procedure for Oklahoma. The heuristic runs for four seconds and the Lagrangian runs for three seconds (when given optimal LP dual variables). The total number of  $x_{ij}$  variables reduces from 1,094,116 to 12,425, a savings of 99%.

### 5.1. Heuristic

After some ad hoc testing, we settled on the following heuristic. It draws on the districting experience of Hess et al. (1965) and the  $k$ -median experience of Resende and Werneck (2004). Pseudocode for `RandomizedHeuristic()` and `LocalSearch()` is given in Algorithms 2 and 3, respectively.

1. Call `RandomizedHeuristic` for 10 iterations to find an initial set  $S$  of centers;
2. Call `LocalSearch(S)` to improve the set of centers, using a first-improvement strategy; and
3. If contiguity is required, solve a particular instance of the SHIR model (detailed later).

The descent steps in the inner loop of `RandomizedHeuristic` originate with Hess et al. (1965) (see their figure 1). They propose to find a heuristic solution by solving a restricted problem. That is, a set  $S$  of  $k$  centers is fixed (i.e., fix  $x_{jj} = 1$  for  $j \in S$ ), and the task is to assign the other vertices to them. At the time, Hess et al. heuristically solved this restricted problem by transportation techniques, but we use a MIP solver and denote the objective of this restricted problem by  $\text{obj}(S)$ . The solution to this restricted problem partitions the vertices into  $k$  districts. The best center of each district is calculated, and the restricted problem is resolved taking this new set of centers as  $S$ . Repeat until convergence. This heuristic works surprisingly well even when the initial set of centers is chosen

**Figure 6.** When Contiguity Is Imposed, 96% of the Center Variables  $x_{ij}$  Are Safely Fixed to Zero for Oklahoma at the Tract Level

Note. The nonfixed tracts are filled in black.

uniformly at random (as observed by Resende and Werneck). Following Resende and Werneck, we run this for 10 iterations and return the best set of centers that was found.

**Algorithm 2** (RandomizedHeuristic( $maxit$ ))

```

1:  $S^* \leftarrow \emptyset$ 
2: for  $i = 1, 2, \dots, maxit$  do
3:   pick a set  $S$  of  $k$  centers uniformly at random
   from  $V$ 
4:   repeat
5:     find a solution  $x^*$  to the Hess model (1) re-
   stricted to the centers  $\{v_1, v_2, \dots, v_k\} \leftarrow S$ 
6:     for  $j = 1, 2, \dots, k$  do
7:       let  $V_j$  be the vertices assigned to  $v_j$  in  $x^*$ 
8:       let  $v_j^*$  be the best center of  $V_j$ , that is, a ver-
       tex  $b \in V_j$  that minimizes  $\sum_{i \in V_j} w_{ib}$ 
9:        $S \leftarrow \{v_1^*, v_2^*, \dots, v_k^*\}$ 
10:    until convergence
11:    if  $obj(S) < obj(S^*)$  then
12:       $S^* \leftarrow S$ 
13:  return  $S^*$ 

```

However, the subsequent local search procedure used by Resende and Werneck would be too burdensome for us. In the traditional  $k$ -median problem, the problem of assigning customers to facilities is trivial once the facilities have been opened: assign each customer to its nearest open facility. This allows one to quickly evaluate whether it is beneficial to swap a center with a non-center. The same cannot be said for districting due to the population bounds and contiguity constraints. This makes the problem of assigning vertices to centers, that is, the function evaluation  $obj(S)$ , an NP-hard problem. Consequently, we find just one local minimum with LocalSearch, starting the search from the best set of centers found by RandomizedHeuristic. To further speed up local search, we only consider swapping a center  $s \in S$  with one of its neighbors  $s' \in V \setminus S$ . This reduces the number of function evaluations from  $|S|(n - |S|)$  to

roughly  $\sum_{i \in S} \deg_G(i)$  in each call of LocalSearch. For California at the tract level, this reduction is from 425,000 to roughly 300.

**Algorithm 3** (LocalSearch( $S$ ))

```

1: for  $s \in S$  do
2:   for  $s' \in N_G(s) \cap (V \setminus S)$  do
3:      $S' \leftarrow (S \setminus \{s\}) \cup \{s'\}$ 
4:     if  $obj(S') < obj(S)$  then
5:       return LocalSearch( $S'$ )
6: return  $S$ 

```

Last, if required, we solve a final MIP to find a contiguous solution. Typically, the solution identified in local search is nearly contiguous and few changes are needed to achieve contiguity. To exploit this observation, we take the current set of district centers  $S \subseteq V$  and fix the associated variables  $x_{ij}$  to one in the SHIR model. For most states, this is sufficient for the MIP solver to identify a good contiguous solution within a few seconds. However, we observed that the MIP solver still struggled to solve this restricted problem on some of the largest instances. In response, we take the following approach. For each center  $j \in S$ , do the following:

- i. Identify the vertices  $V_j$  that are assigned to  $j$  in the local search solution;
- ii. Identify the component of  $G[V_j]$  that contains  $j$  and let  $J$  be its vertex set;
- iii. For each vertex  $i \in J$ , provide  $x_{ij} = 1$  as part of a warm start solution; and
- iv. If working with a tract-level instance: for each vertex  $i$  in the *interior* of  $J$ , fix  $x_{ij} = 1$ .

Here, we define the *interior* of  $J$  to be the vertices in  $J$  whose neighbors in  $G$  also belong to  $J$ .

In practice, we found that the last two steps guide the MIP solver to a contiguous solution without sacrificing much in terms of solution quality. This tweak allowed us to find (contiguous) feasible

solutions for states like Illinois, New York, and Texas at the tract level.

### 5.2. Lagrangian-Based Variable Fixing for Hess Model

Lagrangian techniques are quite common and useful for variants of the  $k$ -median problem (Beasley 1993) and have also been used for districting purposes (Hojati 1996). Here, we propose to use Lagrangian reduced costs to safely fix many of the variables  $x_{ij}$  to zero *before* building the Hess model, thus reducing it to a more manageable size. A similar approach was taken by Briant and Naddef (2004) for the diversity management problem.

The Lagrangian relaxation model is obtained from the Hess model as follows. We relax the assignment Constraints (1b), population lower bounds (1d), and population upper bounds (1d) and penalize their violation in the objective function with (vector) multipliers  $\alpha$ ,  $\lambda$ , and  $v$ , respectively. Following Beasley (1993) and Hojati (1996), we scale the population constraints by dividing them by  $L$  and  $U$ , respectively. The optimal objective of the following Lagrangian is denoted by  $\mathcal{L}(\alpha, \lambda, v)$ :

$$\min \sum_{i \in V} \sum_{j \in V} w_{ij} x_{ij} + \sum_{i \in V} \alpha_i \left(1 - \sum_{j \in V} x_{ij}\right) + \sum_{j \in V} |\lambda_j| \left(x_{jj} - \sum_{i \in V} \frac{p_i}{L} x_{ij}\right) + \sum_{j \in V} |v_j| \left(\sum_{i \in V} \frac{p_i}{U} x_{ij} - x_{jj}\right) \quad (13a)$$

$$\sum_{j \in V} x_{jj} = k \quad (13b)$$

$$x_{ij} \leq x_{jj} \quad \forall i, j \in V \quad (13c)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V. \quad (13d)$$

There are two main differences with previous works (besides the definition of  $w$ ). First, we have population lower and upper bounds ( $L$  and  $U$ ), whereas Beasley (1993) had no analogue of  $L$  and Hojati (1996) had an equality constraint (i.e.,  $L = U$ ). Second, we take the absolute values of  $\lambda_j$  and  $v_j$  in the Lagrangian’s objective function, whereas Beasley (1993) required  $v_j$  to be nonnegative. The absolute-value approach for handling inequality constraints is simpler and more effective for Shor’s  $r$ -algorithm (Shor 1985, p. 134; cf. Mikhalevich et al. 1977, p. 126).

To simplify the Lagrangian’s objective function, we can collect like terms. For this, define

$$\hat{w}_{ij} = \begin{cases} w_{ij} - \alpha_i - |\lambda_j| \left(\frac{p_i}{L}\right) + |v_j| \left(\frac{p_i}{U}\right) & \text{if } i \neq j \\ w_{ij} - \alpha_i - |\lambda_j| \left(\frac{p_i}{L}\right) + |v_j| \left(\frac{p_i}{U}\right) + |\lambda_j| - |v_j| & \text{if } i = j \end{cases}$$

for every  $i \in V$  and  $j \in V$ . The Lagrangian’s objective

function now reduces to

$$\min \sum_{i \in V} \alpha_i + \sum_{i \in V} \sum_{j \in V} \hat{w}_{ij} x_{ij}.$$

To find an optimal solution  $x^*$  to the Lagrangian, consider the case where vertex  $j \in V$  is selected as a center. In this case, it would be optimal to assign vertex  $i$  to center  $j$  if and only if  $\hat{w}_{ij} \leq 0$ . Thus, if  $j \in V$  were selected as a center, its contribution to the objective would be

$$W_j := \hat{w}_{jj} + \sum_{i \in V \setminus \{j\}} \min \{0, \hat{w}_{ij}\},$$

otherwise, the contribution would be zero.

With these observations, the Lagrangian reduces to the following  $n$ -variable problem:

$$\min \sum_{i \in V} \alpha_i + \sum_{j \in V} W_j x_{jj} \quad (14a)$$

$$\sum_{j \in V} x_{jj} = k \quad (14b)$$

$$x_{jj} \in \{0, 1\} \quad \forall j \in V. \quad (14c)$$

This can be solved by identifying the  $k$  different vertices  $j$  with the smallest  $W_j$  values and setting their variables  $x_{jj}$  to one; set all others to zero. The objective value of this solution is equal to  $\mathcal{L}(\alpha, \lambda, v)$ , which provides a lower bound on our original MIP (whether or not contiguity is imposed).

Now, we discuss how to safely fix variables to zero in the Hess model (1) using the heuristic upper bound  $UB$  and the Lagrangian. Let  $x^*$  be an optimal solution to the reduced Lagrangian problem (14) with objective value  $\mathcal{L}(\alpha, \lambda, v)$ . The associated set of centers is  $S := \{v \in V \mid x_{vv}^* = 1\}$ . If we were to tentatively fix  $x_{ij} = 1$ , the resulting Lagrangian bound  $\mathcal{L}_{ij}(\alpha, \lambda, v)$  would be

$$\mathcal{L}_{ij}(\alpha, \lambda, v) = \begin{cases} \mathcal{L}(\alpha, \lambda, v) & \text{if } j \in S, i = j \\ \mathcal{L}(\alpha, \lambda, v) + \max \{0, \hat{w}_{ij}\} & \text{if } j \in S, i \neq j \\ \mathcal{L}(\alpha, \lambda, v) - \max_{v \in S} \{W_v\} + W_j & \text{if } j \in V \setminus S, i = j \\ \mathcal{L}(\alpha, \lambda, v) - \max_{v \in S} \{W_v\} + W_j + \max \{0, \hat{w}_{ij}\} & \text{if } j \in V \setminus S, i \neq j. \end{cases}$$

This value can be used to update the lower bound  $LB_{ij} \leftarrow \max \{\mathcal{L}_{ij}(\alpha, \lambda, v), LB_{ij}\}$  on the objective value that would result from fixing  $x_{ij} = 1$ . Observe that  $\mathcal{L}(\alpha, \lambda, v)$  and  $\max \{W_v \mid v \in S\}$  do not depend on  $i$  nor  $j$ , meaning that they can be precomputed, stored, and then used to update all of the  $LB_{ij}$  values in time  $\Theta(n^2)$ .

The outer problem for the Lagrangian (for finding the best Lagrange multipliers) is controlled by Shor’s  $r$ -algorithm. When it terminates with the final values  $LB_{ij}$ , we fix  $x_{ij} = 0$  if  $LB_{ij} > UB$ .

### 5.3. Lagrangian-Based Variable Fixing for Contiguity Models

We can fix even more variables  $x_{ij}$  to zero by exploiting contiguity. Ideally, we would add constraints to (13) requiring that the vertices  $V_j(x)$  assigned to  $j$  induce a connected subgraph. In this case, the Lagrangian - relaxation model could be solved by redefining  $W_j$  as the weight of a minimum-weight connected subgraph (MWCS) rooted at  $j$  (where the weight of vertex  $i$  is  $\hat{w}_{ij}$ ) and again solving Problem (14). One issue is that the rooted MWCS problem is NP-hard. Although this problem has been studied frequently lately (Álvarez-Miranda et al. 2013a, b; Álvarez-Miranda and Sinnl 2017; Rehfeldt et al. 2019), and several research codes perform quite well on benchmark instances (Fischetti et al. 2017, Gamrath et al. 2017, Rehfeldt and Koch 2019), we choose not to use them given how frequently we would need to solve the rooted MWCS problem. Also, preliminary experiments with solving the rooted MWCS problem exactly showed little to no improvement—in terms of the number of variables fixed—over the simple and very quick procedure that was settled on.

We consider a relaxed form of contiguity for which all  $LB_{ij}$  values can be updated in  $O(n^2)$  time. This relaxed form of contiguity only enforces that an  $i, j$ -path exists within  $G[V_j(x)]$  if  $x_{ij}$  is tentatively fixed to one. For this, define the weights  $q_u^j$  as below, representing the “extra” cost to use vertex  $u$  in an  $i, j$ -path in district  $V_j(x)$ . Observe that the “cost”  $\hat{w}_{ij}$  of vertex  $u$  has already been accounted for in  $W_j$  if  $u = j$  or  $\hat{w}_{ij} \leq 0$ , giving an extra cost of zero.

$$q_u^j = \begin{cases} 0 & \text{if } u = j \text{ or } \hat{w}_{ij} \leq 0 \\ \hat{w}_{ij} & \text{otherwise.} \end{cases}$$

Again, let  $\mathcal{L}(\alpha, \lambda, v)$  be the objective value of the Lagrangian relaxation model (14) and let  $S$  be the associated set of centers. Then, a lower bound on the *contiguity-constrained* Lagrangian relaxation model—when  $x_{ij}$  is tentatively fixed to one—is as follows:

$$\tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v) = \begin{cases} \mathcal{L}(\alpha, \lambda, v) + \text{dist}_{G,q^j}(i, j) & \text{if } j \in S \\ \mathcal{L}(\alpha, \lambda, v) - \max_{v \in S} \{W_v\} & \text{if } j \notin S. \\ +W_j + \text{dist}_{G,q^j}(i, j) \end{cases}$$

Here,  $\text{dist}_{G,q^j}(i, j)$  is the vertex-weighted distance from  $i$  to  $j$ , where vertex  $u$  has weight  $q_u^j$ . These distances  $\text{dist}_{G,q^j}(\cdot, j)$  can be computed in time  $O(n)$  with a planarity-exploiting single-source shortest path algorithm (Henzinger et al. 1997). Thus, all updates  $LB_{ij} \leftarrow \max\{LB_{ij}, \tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v)\}$  can be computed in time  $O(n^2)$ . However, for ease of implementation, we use Dijkstra’s algorithm, so our code takes time  $O(n^2 \log n)$  to update the  $LB_{ij}$  values. As before, we fix  $x_{ij} = 0$  if  $LB_{ij} > UB$ . Theorem 4 ensures that this is safe.

**Theorem 4.** *For every set of Lagrange multipliers  $(\alpha, \lambda, v)$  we have  $z_{ij}^* \geq \tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v)$ , where  $z_{ij}^*$  is the optimal objective of the Hess model when  $x_{ij}$  is fixed to one and contiguity is imposed.*

**Proof.** Consider a set of Lagrange multipliers  $(\alpha, \lambda, v)$ , the corresponding  $\hat{w}$  and  $W$ , and an optimal set of centers  $S$  for the Lagrangian (14). Then, let  $x^*$  be an optimal solution to the Hess model (1) when  $x_{ij}$  is fixed to one and contiguity is imposed, and let  $S^* := \{v \in V \mid x_{vv}^* = 1\}$  be the associated set of centers. Observe that subset of vertices assigned to  $j$  induces a connected subgraph, so there is a path  $P$  from  $i$  to  $j$  whose vertices  $u \in V(P)$  satisfy  $x_{uj}^* = 1$ .

**Claim 1.** We have  $z_{ij}^* \geq \sum_{u \in V} \alpha_u + \sum_{v \in S^*} W_v + \text{dist}_{G,q^j}(i, j)$ .

**Proof.** Using the notations  $(a)^+ := \max\{0, a\}$  and  $(a)^- := \min\{0, a\}$ , observe that

$$z_{ij}^* = \sum_{u \in V} \sum_{v \in V} \hat{w}_{uv} x_{uv}^* \tag{15a}$$

$$\geq \sum_{u \in V} \sum_{v \in V} \hat{w}_{uv} x_{uv}^* + \sum_{u \in V} \alpha_u \left(1 - \sum_{v \in V} x_{uv}^*\right) + \sum_{v \in V} |\lambda_v| \left(x_{vv}^* - \sum_{u \in V} \frac{p_u}{L} x_{uv}^*\right) + \sum_{v \in V} |v_v| \left(\sum_{u \in V} \frac{p_u}{U} x_{uv}^* - x_{vv}^*\right) \tag{15b}$$

$$= \sum_{u \in V} \alpha_u + \sum_{u \in V} \sum_{v \in V} \hat{w}_{uv} x_{uv}^* \tag{15c}$$

$$\geq \sum_{u \in V} \alpha_u + \sum_{v \in S^*} W_v + \text{dist}_{G,q^j}(i, j). \tag{15d}$$

Here, Inequality (15b) holds by the feasibility of  $x^*$  for the Hess model. Equality (15c) holds by the definition of  $\hat{w}$ . Finally, Inequality (15d) holds by Inequalities (16), which are shown here:

$$\sum_{u \in V} \sum_{v \in V} \hat{w}_{uv} x_{uv}^* = \sum_{v \in V} \hat{w}_{vv} x_{vv}^* + \sum_{v \in V} \sum_{u \in V \setminus \{v\}} (\hat{w}_{uv})^- x_{uv}^* + \sum_{v \in V} \sum_{u \in V \setminus \{v\}} (\hat{w}_{uv})^+ x_{uv}^* \tag{16a}$$

$$\geq \sum_{v \in V} \hat{w}_{vv} x_{vv}^* + \sum_{v \in V} \sum_{u \in V \setminus \{v\}} (\hat{w}_{uv})^- x_{uv}^* + \sum_{u \in V(P) \setminus \{j\}} (\hat{w}_{uj})^+ x_{uj}^* \tag{16b}$$

$$\text{(by } x_{uv}^* \leq x_{vv}^*) \geq \sum_{v \in V} \hat{w}_{vv} x_{vv}^* + \sum_{v \in V} \sum_{u \in V \setminus \{v\}} (\hat{w}_{uv})^- x_{uv}^* + \sum_{u \in V(P) \setminus \{j\}} (\hat{w}_{uj})^+ x_{uj}^* \tag{16c}$$

$$= \sum_{v \in V} \left( \hat{w}_{vv} + \sum_{u \in V \setminus \{v\}} (\hat{w}_{uv})^- \right) x_{vv}^* + \sum_{u \in V(P) \setminus \{j\}} (\hat{w}_{uj})^+ x_{uj}^* \tag{16d}$$

$$\text{(by } W_v \text{ def.)} = \sum_{v \in V} W_v x_{vv}^* + \sum_{u \in V(P) \setminus \{j\}} (\hat{w}_{uj})^+ x_{uj}^* \tag{16e}$$

$$\text{(by } q_u^j \text{ def.)} = \sum_{v \in S^*} W_v + \sum_{u \in V(P)} q_u^j \tag{16f}$$

$$\text{(by } \text{dist}_{G,q^j}(i, j) \text{ def.)} \geq \sum_{v \in S^*} W_v + \text{dist}_{G,q^j}(i, j). \quad \square \tag{16g}$$

With Claim 1 established, we turn to the theorem. In the first case, suppose that  $j \in S$ . Then,

$$\begin{aligned} z_{ij}^* &\geq \sum_{u \in V} \alpha_u + \sum_{v \in S^*} W_v + \text{dist}_{G,q^l}(i,j) \\ &\geq \sum_{u \in V} \alpha_u + \sum_{v \in S} W_v + \text{dist}_{G,q^l}(i,j) \\ &= \mathcal{L}(\alpha, \lambda, v) + \text{dist}_{G,q^l}(i,j) = \tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v), \end{aligned}$$

where the inequalities hold by Claim 1 and by the optimality of  $S$  for the Lagrangian (14), and the equalities hold by the definitions of  $\mathcal{L}(\alpha, \lambda, v)$  and  $\tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v)$ .

In the other case, where  $j \notin S$ , there exists a vertex  $s$  that belongs to  $S$  but not to  $S^*$ . (Otherwise,  $S \subseteq S^*$  and  $j \in S^* \setminus S$ , implying the contradiction  $k = |S| < |S^*| = k$ ). Then,

$$\begin{aligned} \sum_{v \in S^*} W_v &= \sum_{v \in (S^* \cup \{s\}) \setminus \{j\}} W_v - W_s + W_j \\ &\geq \sum_{v \in S} W_v - W_s + W_j \\ &\geq \sum_{v \in S} W_v - \max_{v \in S} \{W_v\} + W_j, \end{aligned}$$

which, along with Claim 1, implies that

$$\begin{aligned} z_{ij}^* &\geq \sum_{u \in V} \alpha_u + \sum_{v \in S^*} W_v + \text{dist}_{G,q^l}(i,j) \\ &\geq \sum_{u \in V} \alpha_u + \sum_{v \in S} W_v - \max_{v \in S} \{W_v\} + W_j + \text{dist}_{G,q^l}(i,j) \\ &= \mathcal{L}(\alpha, \lambda, v) - \max_{v \in S} \{W_v\} + W_j + \text{dist}_{G,q^l}(i,j) = \tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v), \end{aligned}$$

where the equalities hold by the definitions of  $\mathcal{L}(\alpha, \lambda, v)$  and  $\tilde{\mathcal{L}}_{ij}(\alpha, \lambda, v)$ .  $\square$

## 6. Computational Experiments

In this section, we conduct an extensive computational study. Broadly speaking, the aim is to answer two questions. (i) Which model for imposing contiguity is the fastest in practice? (ii) Is this fastest model able to solve the large-scale instances encountered in practice?

Most experiments were performed on a machine with Intel Xeon E3-1270 v6 “Kaby Lake” 3.80-GHz CPU with eight cores and 32-GB RAM. The MIP solver is Gurobi Optimizer 8.1.1. When solving the MIPs, default settings are used with the following exceptions: eight threads maximum, 10-GB RAM maximum, concurrent method for the LP relaxation, and zero MIP gap tolerance. We invoke the LazyConstraints parameter when solving the CUT and LCUT models. In the CUT and LCUT implementations, only integer points are separated, using Algorithm 1. This choice was informed by the authors’ previous experience (Salemi and Buchanan 2020, Validi and Buchanan 2020) and the experience of other researchers (Fischetti et al. 2017, Arslan et al. 2019).

Recall that to initialize the Lagrangian procedure we use optimal dual multipliers from the LP

relaxation of the Hess model. In this case, the barrier method is used to solve the LP. For four states with the most census tracts (Florida, New York, Texas, and California), the size of this LP model exceeds available memory, in which case we solve the LP using a different machine that has dual Intel Xeon E5-2620 “Sandy Bridge” hex core 2.0-GHz CPUs and 256-GB RAM.

### 6.1. Data Preparation

To compare the computational performance of the models, we needed test instances having the following input data: contiguity graph  $G = (V, E)$ , population vector  $p$ , bounds  $L$  and  $U$ , number of districts  $k$ , and the distance matrix  $d$ . These data are not directly provided by the Census (particularly the graph and distance matrix), and optimization researchers who have studied districting in the past did not make their data public. This is a significant barrier to entry for those who want to apply their methods to redistricting problems. Moreover, previous studies have focused on one or two states, making it impossible to know how well their methods would perform on another state or if the data were different (say, after a new Census).

For these reasons, we generate the requisite data ourselves for all 50 states using the 2010 Census numbers. So that future researchers can use the same data and also for purposes of transparency, we post the complete data set online at <https://lykhovyd.com/files/public/districting/>. The GitHub repository includes scripts used to convert the raw data from the Bureau of the Census (2012) into formats convenient for our use.

We generate the input data at two different levels: county and census tract. There is an average of 62 counties per state and a maximum of 254 for Texas. Meanwhile, there is an average of 1,461 census tracts per state and a maximum of 8,057 in the case of California. The county-level instances are small enough that we can apply each of the contiguity formulations. This allows us to compare them and discern which of them has the best chance to handle large instances. Moreover, several states require congressional redistricting plans to split the fewest number of counties, and states such as Iowa and West Virginia split zero counties in their 2013 maps. This makes county-level instances practical in some cases. However, most states cannot redistrict at the county level. For example, Dallas County in Texas had a population of 2,368,139 after the 2010 Census, meaning that it needed to be split into (at least) four congressional districts in order to satisfy the rigid population bounds. Census tracts are designed to be relatively homogeneous and typically have between 2,500 and 8,000 people in them, which provides enough granularity to satisfy the population bounds. The tract-level

instances are sufficiently large and challenging that roughly half of them can be solved by our techniques, allowing us to show their computational limits. Note, however, that no state has a law requiring the indivisibility of tracts, and many states perform districting at the block level, where  $n$  approaches one million. The exact techniques considered in this paper cannot handle such instances, so we do not generate block-level data.

The data are constructed for each state separately, using a suitable map projection from the EPSG data set (IOGP 2019). The centroid of each county or tract is taken as its center, and Euclidean distances are measured between centroids. If two counties or tracts share a nontrivial border, we connect them by an edge in  $G$ . For some states, the graph  $G$  is disconnected (e.g., because of islands off a state's coast). In this case, we make it connected by adding a minimum-weight subset of (currently missing) edges via a straightforward extension of Kruskal's algorithm. We take the weight of an edge  $\{i, j\}$  to be the distance between (the centroids)  $i$  and  $j$ . In this way, a *single* island tract will be made adjacent to its nearest tract from the coast. However, if there is a tightly packed *cluster* of islands sufficiently far from the coast, only one edge will be added connecting the islands to the coast; the other added edges will be internal to the island cluster.

Although rare, some tracts are themselves non-contiguous. For example, the tract in Massachusetts with GEOID 25023990003 consists of three disconnected pieces, each a body of water with zero population. In these cases, a district that is connected in the contiguity graph may not be contiguous on the map. In fact, this happened in our experiments. To address this issue, one could adjust the contiguity graph by creating a different node for each of the tract's pieces, or by merging the disconnected pieces into neighboring tracts (Duchin 2020). However, as our intent in this paper is to compare the performance of the contiguity formulations on realistic instances—and not to create good plans—we opt to keep the contiguity graphs as is. In the future, this might not be a problem; the criteria specified for the 2020 Census state that “Census tracts must comprise a reasonably compact and contiguous land area” (Bureau of the Census 2018).

The other parameters are set as follows, where the number  $k$  of congressional districts is known and set by reapportionment. We compute the ideal population  $\bar{p} := 1/k \sum_{i \in V} p_i$  of a district and allow a 1% deviation, setting  $\hat{L} := 0.995(\bar{p})$  and  $\hat{U} := 1.005(\bar{p})$ , as was suggested in a redistricting competition held by reformers in Ohio (Altman and McDonald 2018). We then round the population bounds (in the appropriate direction) to an integer, that is,  $L := \lceil \hat{L} \rceil$  and  $U := \lfloor \hat{U} \rfloor$ .

## 6.2. Experiments with Heuristic and Variable Fixing

Table 1 reports our experience with the heuristic and variable fixing procedures from Section 5. For space considerations, only tract-level results are reported here in the paper. Tract-level results are also more interesting because this is where these procedures are most needed.

Reported under the Lagrangian columns are the lower bound obtained from the Lagrangian and the time in seconds spent by Shor's  $r$ -algorithm. As noted in Section 5, we initialize Shor's  $r$ -algorithm with optimal dual multipliers from the LP relaxation of the Hess model. The time to solve this LP can be substantial—five days for California—so we precompute the LP dual multipliers and store them in the `ralg_warm` directory for reuse; the time to solve this LP is not included in the time given in Table 1. We see that if Shor's  $r$ -algorithm is limited to (at most) 100 iterations, it terminates in less than one minute for most instances. The last six columns report the upper bound obtained via the heuristic (without and with contiguity constraints), the time spent by the heuristic, and the percentage of the variables  $x_{ij}$  that are eventually fixed.

Generally speaking, the objective values of the heuristic solutions (contiguous versus not) are similar. Therefore, the price of contiguity appears small. (This is confirmed for *optimal* solutions later.) Recall that in our heuristic we solve a series of restricted MIPs. For speed considerations, we do not force the MIP solver to prove optimality for these restricted MIPs. This explains the perhaps counterintuitive observation that, when contiguity is enforced, the heuristic's objective value sometimes *improves* (e.g., for New Hampshire, Idaho, Oregon, and Oklahoma).

Another observation is that the variable fixing procedure is quite powerful, sometimes allowing us to fix approximately 100% of the variables. Although this is most pronounced on the smaller instances (e.g., Rhode Island, New Hampshire, Idaho, Hawaii, and Maine), some large instances are also amenable to fixing. Roughly 97% of the variables can be fixed for Indiana, which has  $n = 1,511$  tracts.

Finally, we observe that it can be quite helpful to exploit contiguity in the variable fixing procedure (Section 5.3). For example, consider the case of Tennessee. Here, the heuristic's objective value does not change when contiguity is imposed, but the fixings increase from 8% to 44%. In another example, Colorado sees the fixings increase from 40% to 53% despite a degradation in objective value.

## 6.3. County-Level Results

Most county-level instances are infeasible. For example, consider Texas. Dallas County had a population

**Table 1.** Heuristic and Lagrangian Results

State	$n$	$k$	Lagrangian		Without contiguity			With contiguity		
			LB	Time	UB	Time	Fixed	UB	Time	Fixed
RI	244	2	228.33	0.09	228.42	0.21	100	228.42	0.23	100
NH	295	2	2,687.72	0.09	2,689.07	0.18	100	2,688.28	0.30	100
ID	298	2	53,913.93	0.22	53,930.05	0.15	100	53,916.36	0.20	100
HI	351	2	13,990.50	0.22	13,991.27	0.17	100	13,991.95	0.29	100
ME	358	2	7,713.30	0.30	7,716.04	0.16	100	7,716.04	0.22	100
WV	484	3	10,789.79	0.52	11,801.80	0.66	48	11,834.84	0.87	58
NM	499	3	31,575.28	0.53	31,598.10	1.72	95	31,608.19	1.91	95
NE	532	3	21,975.38	0.59	21,983.42	1.28	99	21,983.42	1.45	99
UT	588	4	22,029.31	0.81	22,035.62	2.03	97	22,037.00	2.50	97
MS	664	4	16,295.54	0.98	16,303.82	1.81	100	16,308.89	2.05	100
AR	686	4	15,490.50	1.02	15,565.41	3.85	96	15,569.97	4.09	97
NV	687	4	11,367.90	1.32	12,443.84	5.96	35	12,497.05	6.14	43
KS	770	4	22,654.34	1.43	22,786.63	3.08	82	22,786.69	3.87	85
IA	825	4	18,164.68	1.72	18,202.13	2.77	96	18,206.43	3.37	97
CT	833	5	1,420.41	1.68	1,422.13	4.50	99	1,422.13	4.74	99
OR	834	5	26,742.01	1.87	26,750.23	9.55	98	26,750.15	10.36	98
OK	1,046	5	19,106.02	3.34	19,132.60	3.60	99	19,131.99	4.35	99
SC	1,103	7	8,356.08	3.28	9,293.98	13.48	18	9,294.45	14.94	40
KY	1,115	6	14,793.25	2.90	14,833.72	10.25	95	14,835.16	11.27	96
LA	1,148	6	13,805.17	3.61	14,829.56	12.52	32	14,830.45	13.92	48
AL	1,181	7	13,058.24	3.86	13,510.26	17.54	49	13,510.26	18.08	68
CO	1,249	7	19,715.16	4.27	19,916.43	15.99	40	19,918.36	17.50	53
MN	1,338	8	24,207.22	4.86	24,220.52	184.95	89	24,224.19	217.11	87
MO	1,393	8	18,830.09	5.47	21,214.38	29.24	16	21,222.79	31.44	44
MD	1,406	8	5,079.56	5.56	5,084.47	26.79	94	5,084.53	29.75	95
WI	1,409	8	16,272.83	5.34	16,340.39	28.70	79	16,340.81	32.23	84
WA	1,458	10	12,376.66	5.70	12,604.21	47.42	53	12,609.09	71.92	61
MA	1,478	9	2,558.82	6.09	2,626.48	50.55	41	2,627.03	53.75	57
TN	1,497	9	10,783.27	6.05	13,092.08	48.49	8	13,092.08	51.55	44
IN	1,511	9	11,061.11	6.46	11,084.81	28.10	97	11,085.37	32.27	97
AZ	1,526	9	29,479.16	6.71	30,219.11	81.92	22	30,220.39	88.56	33
VA	1,907	11	12,836.67	11.22	13,814.08	379.21	20	13,815.25	391.63	49
GA	1,969	14	15,836.59	11.43	15,955.65	243.27	67	15,955.65	242.99	79
NJ	2,010	12	2,250.25	12.09	2,291.51	213.81	52	2,291.51	216.89	70
NC	2,195	13	14,416.66	14.29	14,679.00	331.61	52	14,680.02	350.92	77
MI	2,813	14	24,569.50	24.79	24,685.68	775.69	56	24,686.55	788.41	71
OH	2,952	16	11,520.54	26.59	11,908.25	1,936.37	25	11,911.64	2,056.53	65
IL	3,123	18	15,815.07	30.26	16,067.49	14,316.10	40	16,091.05	14,403.97	55
PA	3,218	18	11,424.45	32.59	11,799.41	6,674.80	38	11,806.00	6,721.78	66
FL	4,245	27	14,766.11	58.53	15,222.31	7,668.40	38	15,225.47	7,811.97	65
NY	4,919	27	15,405.98	77.60	16,700.05	13,848.68	7	16,700.81	15,120.01	45
TX	5,265	36	57,770.13	105.63	72,846.42	14,691.58	1	72,860.21	15,657.41	14
CA	8,057	53	26,238.24	218.99	27,603.66	22,122.05	28	—	—	—

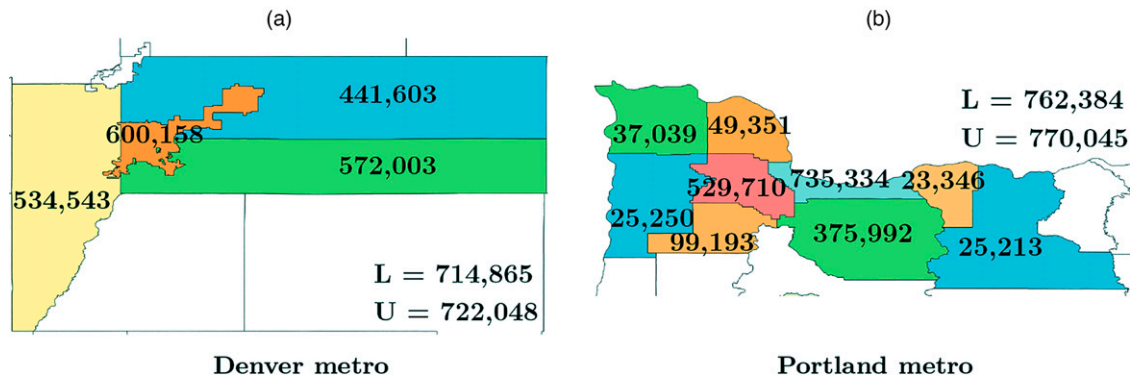
*Note.* We report the number of tracts ( $n$ ), the number of districts ( $k$ ), the Lagrangian objective (LB), and time in seconds, as well as the heuristic objective (UB), time, and the percentage of the variables fixed to zero (fixed) when contiguity is (not) imposed—rounded to the nearest percent.

of  $p_v = 2,368,139$ , which far exceeds the population limit  $U = 701,980$  that we impose. Thus, Texas is obviously infeasible at the county level. This type of overt infeasibility where the population of a single vertex exceeds  $U$  is exhibited by 27 states. These, as well as the seven trivial instances where  $k = 1$ , are uninteresting and are excluded from our county-level experiments (Table 2).

This leaves 16 county-level instances. It turns out that each of them is MIP-feasible under the Hess model, although 4 of them become infeasible when

contiguity is imposed (Colorado, New Hampshire, Oregon, and South Carolina), leaving 12 contiguity-feasible instances. The reasons why Colorado and Oregon are infeasible can be understood by inspecting the submaps depicted in Figure 7. For four instances, the optimal solution that is returned by the Hess model under the moment-of-inertia (MOI) objective happens to be contiguous (Iowa, Idaho, Kansas, Mississippi). In the other eight cases, the Hess solution that is found is not contiguous. Furthermore, when contiguity is explicitly imposed, the optimal objective

**Figure 7.** (Color online) Explaining County-Level Infeasibility



Notes. (a) Colorado is infeasible at the county level because there is no feasible district containing Denver County (population 600,158). (b) Oregon is infeasible at the county level because there is no feasible district containing Multnomah County (population 735,334).

value increases, implying that for these states no optimal solution to the Hess model is contiguous (Figure 8).

Table 3 reports the optimal objective values and solve times (in seconds) for the 16 remaining county-level instances. The reported times include all operations (heuristic, Lagrangian fixing, model build, model solve) excluding read time and the time to get the initial Lagrange multipliers (see Section 5). More details can be found in the results directory in the GitHub repository. As the table shows, the Hess model is relatively easy for the MIP solver at the county level, with each being solved to optimality in under 10 seconds.

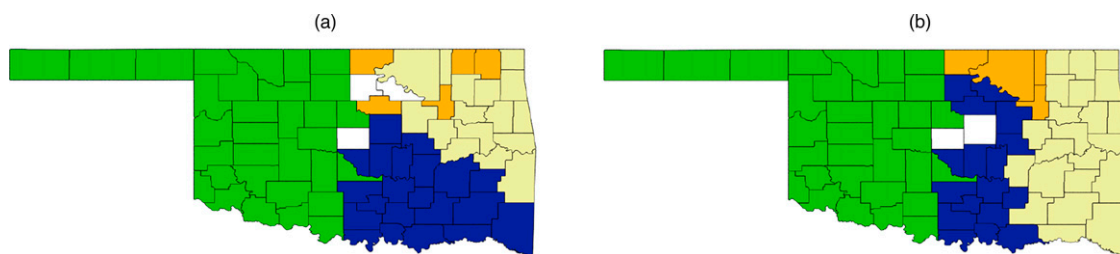
For the contiguity models, one observes that the MCF model is the slowest (attributable to its large size), whereas the others are reasonably competitive with each other. In some cases, LCUT edges out the competition. For example, LCUT edges out SHIR for Alabama (19 versus 64 seconds). We attribute this time difference to LCUT being smaller and more nimble than SHIR. Meanwhile, LCUT edges out CUT for Colorado (2 seconds versus time limit (TL)). In fact, we find that LCUT proves infeasibility of Colorado at

the root node of the branch-and-bound tree. To illustrate, recall Figure 7. Denver County cannot be paired with any adjacent counties without exceeding the population upper bound. Thus, the special case of the length- $U$   $a,b$ -separator inequalities from Remark 3 enforces that Denver County cannot be assigned to other counties, nor can other counties be assigned to it. Therefore, Denver County must be in its own district. However, this conflicts with the population lower bound, proving infeasibility. We also observe that SHIR proves the infeasibility of Colorado at the root node—with the help of the MIP solver’s presolve and

**Table 2.** Initial Classification of the 50 County-Level Instances

Class	No.	States
Overt infeasibility	27	AZ, CA, CT, FL, GA, HI, IL, IN, KY, MA, MC, MI, MN, MO, NC, NJ, NV, NY, OH, PA, RI, TN, TX, UT, VA, WA, WI
Trivial ( $k = 1$ )	7	AK, DE, MT, ND, SD, VT, WY
Remaining	16	AL, AR, CO, IA, ID, KS, LA, ME, MS, NE, NH, NM, OK, OR, SC, WV

**Figure 8.** (Color online) Optimal County-Level Solutions for Oklahoma



Notes. (a) Optimal solution without contiguity. (b) Optimal solution with contiguity.

**Table 3.** Times and Objective Values for Solving the Resulting MIPs at the County Level

State	$n$	$k$	Without contiguity		With contiguity				
			Objective	Hess	Objective	MCF	SHIR	CUT	LCUT
NH	10	2	3,641.24	0.03	$\infty$	0.23	0.12	0.12	0.08
ME	16	2	10,789.07	0.03	19,093.26	2.42	0.78	0.72	0.66
NM	33	3	32,847.05	0.07	32,944.25	0.63	0.11	0.10	0.10
OR	36	5	31,424.49	0.40	$\infty$	55.80	0.59	3.08	0.32
ID	44	2	61,232.58	0.06	61,232.58	2.79	0.14	0.09	0.09
SC	46	7	12,479.94	8.67	$\infty$	TL	TL	TL	TL
WV	55	3	11,844.76	0.82	12,000.62	62.16	1.68	0.86	0.84
LA	64	6	17,239.16	2.14	17,272.65	66.87	4.08	2.22	2.23
CO	64	7	35,466.64	2.46	$\infty$	215.25	2.74	TL	1.94
AL	67	7	15,519.54	3.83	16,627.25	457.52	64.29	20.94	19.41
AR	75	4	16,525.07	0.53	16,543.15	29.04	1.62	0.96	0.96
OK	77	5	21,527.57	1.01	21,756.50	101.64	5.06	1.64	1.97
MS	82	4	16,142.04	0.41	16,142.04	8.31	0.54	0.49	0.49
NE	93	3	22,112.00	0.28	22,193.01	13.33	0.57	0.45	0.47
IA	99	4	17,748.05	0.85	17,748.05	29.23	1.47	0.87	0.87
KS	105	4	23,736.89	1.34	23,736.89	48.98	1.78	1.10	1.11

Note. Infeasibilities are denoted by an objective value of  $\infty$ , whereas TL denotes that the one-hour time limit was reached.

cuts. When these features are turned off, branching is required to solve the SHIR model.

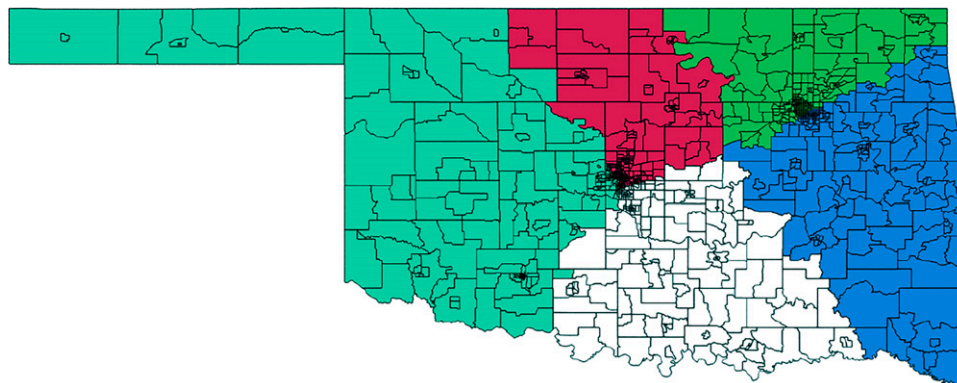
Surprisingly, South Carolina (with only 46 counties!) was left unsolved by all contiguity models after a one-hour time limit. Digging deeper, we find that each of its counties can be placed in a suitable district (satisfying contiguity and population balance) but that these districts cannot be pieced into a full districting plan. This provides a partial explanation why the contiguity models have more trouble proving the infeasibility of this instance than, say, Colorado. Another reason why the models might struggle is a sort of symmetry. To illustrate, observe that a partition  $(V_1, V_2, \dots, V_k)$  of the vertices can be represented in many ways in the formulation  $(|V_1||V_2| \dots |V_k|)$  to be exact). Normally, we can distinguish between these solutions by their objective values (which differ based on the choice of centers). If any such feasible solution is discovered, then the other representations (with inferior objectives) would be pruned by bound. However, when the instance is infeasible there is no incumbent

solution that can be used for pruning, unleashing the combinatorial explosion. If an instance is suspected to be infeasible, we can attempt to prove this by imposing a canonical center for a district, say, with the largest population, by fixing  $x_{ij} = 0$  whenever  $p_i > p_j$ . In this way, we can prove the infeasibility of South Carolina with LCUT in under three seconds and safely report the objective value in Table 3 as  $\infty$ . Applegate (2019) and Buchanan (2019) confirm the infeasibility of South Carolina using different methods.

#### 6.4. Tract-Level Results

Now we report tract-level results for the Hess, SHIR, and CUT formulations. The formulation MCF is omitted because it is too large to handle tract-level instances, and LCUT is omitted because it performs nearly the same as CUT on *tract-level* instances. This is because, on tract-level instances, the minimal  $a,b$ -separator obtained in Algorithm 1 is typically already a minimal length- $U$   $a,b$ -separator, in which case the CUT and LCUT callbacks add the same violated

**Figure 9.** (Color online) Tract-Level Solution for Oklahoma That Is Optimal for the Hess and CUT Models



**Table 4.** Results for Solving the Resulting MIPs at the Tract Level

State	Hess				SHIR			CUT		
	LB	UB	Time	Contiguous	LB	UB	Time	LB	UB	Time
RI	228.42	228.42	0.00	Yes	228.42	228.42	0.02	228.42	228.42	0.00
NH	2,688.28	2,688.28	0.03	Yes	2,688.28	2,688.28	0.13	2,688.28	2,688.28	0.01
ID	53,916.36	53,916.36	0.01	Yes	53,916.36	53,916.36	0.02	53,916.36	53,916.36	0.00
HI	13,991.27	13,991.27	0.01	No	13,991.63	13,991.63	0.15	13,991.63	13,991.63	0.01
ME	7,716.04	7,716.04	0.00	—	7,716.04	7,716.04 <sup>b</sup>	0.02	7,716.04	7,716.04 <sup>b</sup>	0.00
WV	11,026.27	11,801.80	TL	—	10,937.99	11,834.84	TL	11,053.92	11,834.84	TL
NM	31,598.10	31,598.10	24.79	No	31,603.15	31,603.15	216.76	31,603.15	31,603.15	44.78
NE	21,983.14	21,983.14	0.22	No	21,983.42	21,983.42	2.40	21,983.42	21,983.42	0.16
UT	22,035.37	22,035.37	9.57	No	22,037.00	22,037.00	44.09	22,037.00	22,037.00	16.85
MS	16,303.82	16,303.82	0.17	No	16,305.45	16,305.45	1.57	16,305.45	16,305.45	0.19
AR	15,563.52	15,563.52	17.72	No	15,569.97	15,569.97	119.47	15,569.97	15,569.97	14.99
NV	11,383.34	12,443.84	TL	—	11,368.70	12,497.05	TL	11,370.55	12,497.05	TL
KS	22,784.49	22,784.49	1,310.10	No	22,784.53	22,786.69	TL	22,786.69	22,786.69	660.25
IA	18,172.01	18,172.01	12.27	No	18,176.37	18,176.37	276.96	18,176.37	18,176.37	39.99
CT	1,422.13	1,422.13	8.39	No	1,422.13	1,422.13	32.21	1,422.13	1,422.13	6.96
OR	26,749.59	26,749.59	81.37	No	26,749.74	26,749.74	733.67	26,749.74	26,749.74	83.74
OK	19,107.72	19,107.72	2.09	Yes	19,107.72	19,107.72	17.94	19,107.87	19,107.87 <sup>a</sup>	1.65
SC	8,357.73	9,293.98	TL	—	—	9,294.45	LPNS	8,360.04	9,294.45	TL
KY	14,833.72	14,833.72	244.93	No	14,835.16	14,835.16	2,272.44	14,835.16	14,835.16	2,354.22
LA	13,909.48	14,829.56	TL	—	—	14,830.45 <sup>b</sup>	LPNS	13,909.56	14,830.45 <sup>b</sup>	TL
AL	13,073.21	13,510.26	TL	—	—	13,510.26	LPNS	13,080.87	13,510.26	TL
CO	—	19,916.43	LPNS	—	—	19,918.36	LPNS	19,715.23	19,918.36	TL
MN	24,217.31	24,220.16	TL	—	24,210.23	24,223.84	TL	24,217.35	24,224.19	TL
MO	—	21,214.38	LPNS	—	—	21,222.79	LPNS	—	21,222.79	LPNS
MD	5,082.38	5,084.32	TL	—	5,080.71	5,084.53	TL	5,083.39	5,084.53	TL
WI	16,272.94	16,311.75	TL	—	—	16,340.81	LPNS	16,273.67	16,340.81	TL
WA	—	12,604.21	LPNS	—	—	12,609.09	LPNS	—	12,609.09	LPNS
MA	—	2,626.48	LPNS	—	—	2,627.03 <sup>b</sup>	LPNS	—	2,627.03 <sup>b</sup>	LPNS
TN	10,783.27	13,092.08	TL	—	—	13,092.08	LPNS	10,786.16	13,092.08	TL
IN	11,078.13	11,081.76	TL	—	11,075.06	11,081.76	TL	11,078.06	11,081.76	TL
AZ	—	30,219.11	LPNS	—	—	30,220.39	LPNS	—	30,220.39	LPNS
VA	—	13,814.08	LPNS	—	—	13,815.25	LPNS	12,836.85	13,815.25	TL
GA	15,836.60	15,955.65	TL	—	—	15,955.65	LPNS	—	15,955.65	LPNS
NJ	—	2,291.51	LPNS	—	—	2,291.51	LPNS	—	2,291.51	LPNS
NC	—	14,679.00	LPNS	—	—	14,680.02	LPNS	14,417.50	14,680.02	TL
MI	—	24,685.68	LPNS	—	—	24,686.55	LPNS	—	24,686.55	LPNS
OH	—	11,908.25	LPNS	—	—	11,911.64	LPNS	—	11,911.64	LPNS
IL	—	16,067.49	LPNS	—	—	16,091.05	LPNS	—	16,091.05	LPNS
PA	—	11,799.41	LPNS	—	—	11,806.00	LPNS	—	11,806.00	LPNS
FL	—	15,222.31	LPNS	—	—	15,225.47	LPNS	—	15,225.47	LPNS
NY	—	16,700.05	LPNS	—	—	16,700.81	LPNS	—	16,700.81	LPNS
TX	—	72,846.42	LPNS	—	—	72,860.21	LPNS	—	72,860.21	LPNS
CA	—	27,603.66	LPNS	—	—	—	LPNS	—	—	LPNS

*Notes.* For each formulation, we report the final lower and upper bounds at termination within a 3,600-second time limit, as reported by Gurobi. When the LP relaxation does not solve within the time limit, the time is reported as LPNS.

<sup>a</sup>The objective values obtained for the CUT and SHIR models are inconsistent with each other. Suspecting this inconsistency was because of a numerical issue, we reran both models with the NumericFocus parameter set to three, which is an action recommended by Gurobi for numerically unstable instances (Gurobi 2017). With this change, both models reported an objective value of 19,107.72.

<sup>b</sup>The best solution that was found is connected in the contiguity graph but not on the map (see Section 6.1). This is because of the noncontiguous tracts in Massachusetts (25023990003), Louisiana (22075050100), and Maine (multiple).

inequality. As before, we do not consider the seven trivial instances (where  $k = 1$ ) in our experiments. The remaining 43 tract-level instances are feasible. Figure 9 gives an optimal solution for Oklahoma.

Table 4 gives the results under a one-hour time limit. In every case, we apply the heuristic from Section 5 and provide the resulting solution to Gurobi as a MIP start. We also use the associated upper bound in the

Lagrangian-based procedure discussed in Section 5 to safely fix some variables to zero. As expected, the formulations solve most quickly when the number of tracts is small, and they generally struggle more and more as  $n$  increases. They all solve instances as big as Kentucky ( $n = 1,115$ ), but solve none of the larger instances within a one-hour time limit. Somewhat surprisingly, the MIP solver even struggles with West

**Table 5.** Summary of Tract-Level MIP Results with a One-Hour Time Limit

Status	Hess	SHIR	CUT
No. MIPs solved	16	15	16
No. LPs solved (but not MIP)	11	6	13
No. LPs not solved	16	22	14

Virginia ( $n = 484$ ) and Nevada ( $n = 687$ ) whether or not contiguity is imposed. One explanation for this is that the Lagrangian-based reduced cost fixing is less effective on these instances (recall Table 1).

In some cases, Hess is noticeably faster than SHIR and CUT; for example, the times for Kentucky are 245, 2,272, and 2,354 seconds, respectively. There are also cases where CUT is noticeably faster than Hess and SHIR; for example, the times for Kansas are 660, 1,310, and >3,600 seconds, respectively. Meanwhile, SHIR is consistently the slowest, performing worse than Hess in all cases, and worse than CUT in all but one case (Kentucky, by 82 seconds). The dominance of Hess over SHIR matches the experience of many researchers over the years who have observed that contiguity constraints tend to make problems more difficult. However, if using an alternative model for enforcing contiguity (CUT), this no longer seems to be true. In fact, the CUT and Hess models seem to have very similar performance. They solve the same 16 instances within a one-hour time limit, and their running times on these instances are on par with each other.

Another observation is the LP relaxations are often quite difficult to solve. Half of the SHIR LP relaxations could not be solved within the one-hour time limit, and one third of the CUT LP relaxations could not be solved. This is summarized in Table 5.

Over the course of this project, the authors discovered five other instances that could be solved if the MIP solver was given more time. Results for these five instances are given in Table 6. Interestingly, the MIP solver could not solve the Hess model for Minnesota when it is told to obey a one-hour time limit; however, when no time limit is imposed, it solves in 2,205 seconds. The MIP solver appears to behave differently when told to obey a one-hour time limit.

**Table 6.** Tract-Level MIPs That Were Solved with More Time

State	$n$	$k$	Hess		CUT	
			Objective	Time	Objective	Time
WV	484	3	11,801.26	27,056.92	11,834.84	28,984.58
MN	1,338	8	24,220.16	2,204.75	24,221.70	92,312.17
MD	1,406	8	5,084.32	36,553.32	5,084.44	75,539.58
WI	1,409	8	16,311.75	173,216.16	16,312.72	293,674.57
IN	1,511	9	11,081.72	10,570.75	11,081.72	13,435.36

## 7. Conclusion

Many researchers have stated that contiguity constraints make districting problems particularly difficult to solve. However, for the classical districting problem of Hess et al. (1965), the experiments conducted in this paper suggest otherwise; the exact same 21 tract-level instances are solved to optimality whether or not contiguity is imposed. One explanation for this phenomenon is that the compactness objective proposed by Hess et al. leads to *nearly* contiguous districting plans; a slight nudge is all that is needed to achieve “full” contiguity. Formulations based on graph cuts, like the CUT formulation of Oehrlein and Haurert (2017) and the newly proposed LCUT formulation, are well suited to this task. Few inequalities are needed to prove optimality. Meanwhile, flow-based formulations can also perform well on smaller instances, but their size becomes a problem sooner as their LP relaxations are larger. The practical difficulty of handling contiguity constraints under other objective functions (e.g., those unrelated to compactness) is less clear and is left for future work.

The biggest bottleneck is solving the root LP relaxations. The bedrock of the formulations considered in this paper—the Hess formulation—grows quadratically with the number  $n$  of vertices. With Lagrangian reduced-cost fixing, many of these variables can be avoided, allowing half of the tract-level instances to be solved. With small tweaks to the implementation, other instances may be within reach, but entirely new ideas may be needed to solve the largest instances like Texas ( $n = 5,265$ ) and California ( $n = 8,057$ ).

We make no claims that the maps generated in our computational experiments are good or even legal. For example, our implementation does not explicitly consider the Voting Rights Act, political subdivisions, partisan makeup of the districts, or any laws that vary by state. We consider one measure of compactness (the most prominent one in OR models), but there are many others in the literature. We hope that the work conducted in this paper (including our publicly available source code and test instances) will provide a basis for further districting research.

## Acknowledgments

The authors thank Jose Walteros for many helpful comments.

## Appendix A. Formulation Correctness

Here we prove the correctness of the contiguity formulations.

**Definition A.1.** A given  $\hat{x} \in \{0,1\}^{n \times n}$  is consistent if  $\hat{x}_{ij} \leq \hat{x}_{ji}$  for all  $i, j \in V$ .

In the context of the  $x$  variables, a districting plan is represented as follows. If a given  $\hat{x} \in \{0,1\}^{n \times n}$  is consistent,

the associated set of district “roots” is given by

$$R(\hat{x}) := \{b \in V \mid \hat{x}_{bb} = 1\},$$

and the district rooted at  $b \in R(\hat{x})$  is given by

$$V_b(\hat{x}) := \{a \in V \mid \hat{x}_{ab} = 1\}.$$

Supposing that  $\hat{x}$  is consistent, the four bright-line rules can be expressed mathematically as follows:

1. Each vertex  $i \in V$  belongs to exactly one of the sets  $V_r(\hat{x})$ ,  $r \in R(\hat{x})$ ;
2. We have  $|R(\hat{x})| = k$ ;
3. For every  $r \in R(\hat{x})$ ,  $L \leq \sum_{i \in V_r(\hat{x})} p_i \leq U$ ; and
4. For every  $r \in R(\hat{x})$ ,  $V_r(\hat{x})$  induces a connected subgraph in  $G$ .

We restate the correctness theorem for completeness.

**Theorem 2.** *Formulations SHIR, MCF, CUT, and LCUT are correct.*

**Proof.** We are to show that, for each formulation  $F$ ,  $\hat{x} \in \{0,1\}^{n \times n}$  is consistent and satisfies the four bright-line rules if and only if  $\hat{x} \in \text{proj}_x F$ . By Theorem 1, it suffices to show the two claims:

1. If  $\hat{x} \in \{0,1\}^{n \times n}$  is consistent and satisfies the four bright-line rules, then  $\hat{x} \in \mathcal{P}_{\text{LCUT}}$ ; and
2. If  $(\hat{x}, \hat{f}) \in \mathcal{P}_{\text{SHIR}}$  and  $\hat{x} \in \{0,1\}^{n \times n}$ , then  $\hat{x}$  is consistent and satisfies the four bright-line rules.

To prove the first claim, suppose that  $\hat{x}$  satisfies the four bright-line rules and is consistent. By the first three rules,  $\hat{x}$  satisfies Constraints (9a) from the Hess model. Therefore, all that remains is to show that  $\hat{x}$  satisfies the length- $U$   $a,b$ -separator Inequalities (9b). Therefore, consider vertices  $a$  and  $b$  and a length- $U$   $a,b$ -separator  $C \subseteq V \setminus \{a, b\}$ . If  $\hat{x}_{ab} = 0$ , then Constraint (9b) is trivially satisfied, so suppose  $\hat{x}_{ab} = 1$ . This implies that  $a \in V_b(\hat{x})$ . By the fourth rule,  $V_b(\hat{x})$  induces a connected subgraph in  $G$ , implying that there exists a path  $P$  from  $a$  to  $b$  in  $G[V_b(\hat{x})]$ . Moreover, this path  $P$  has length at most  $U$ , because  $V(P)$  is a subset of  $V_b(\hat{x})$  and because  $p(V_b(\hat{x})) \leq U$  by the third rule. By the definition of a length- $U$   $a,b$ -separator, there is at least one vertex  $c \in V_b(\hat{x})$  that belongs to both  $C$  and  $V(P)$  and so Constraint (9b) is satisfied as

$$\hat{x}_{ab} = 1 = \hat{x}_{cb} \leq \sum_{j \in C} \hat{x}_{jb}.$$

To prove the second claim, suppose that  $(\hat{x}, \hat{f}) \in \mathcal{P}_{\text{SHIR}}$  and  $\hat{x} \in \{0,1\}^{n \times n}$ . This implies  $\hat{x}$  is consistent. By the correctness of the Hess formulation, the districting plan  $V_b(\hat{x}), b \in R(\hat{x})$ , satisfies the first three bright-line rules. Therefore, it suffices to show that the fourth holds, that is, that each vertex subset  $V_b(\hat{x})$  induces a connected subgraph  $G_b := G[V_b(\hat{x})]$  in  $G$ . For contradiction purposes, suppose that some  $V_b(\hat{x})$  induces at least two connected components in  $G$ , and let  $S$  be the vertex set of a component of  $G_b$  that does not contain vertex  $b$ . Let  $N(S)$  be the open neighborhood of set  $S$ , that is,

$$N(S) := \left( \bigcup_{v \in S} N(v) \right) \setminus S.$$

Then,

$$1 \leq |S| = \sum_{j \in S} \hat{x}_{jb} = \sum_{j \in S} \left( \hat{f}^b(\delta^-(j)) - \hat{f}^b(\delta^+(j)) \right) \quad (\text{A.1a})$$

$$= \sum_{j \in S} \left( \sum_{i \in N(j)} \hat{f}_{ij}^b - \hat{f}_{ji}^b \right) \quad (\text{A.1b})$$

$$= \sum_{j \in S} \sum_{i \in N(j) \cap S} \left( \hat{f}_{ij}^b - \hat{f}_{ji}^b \right) + \sum_{j \in S} \sum_{i \in N(j) \cap (V \setminus S)} \left( \hat{f}_{ij}^b - \hat{f}_{ji}^b \right) \quad (\text{A.1c})$$

$$= 0 + \left( \sum_{j \in S} \sum_{i \in N(j) \cap (V \setminus S)} \hat{f}_{ij}^b - \sum_{j \in S} \sum_{i \in N(j) \cap (V \setminus S)} \hat{f}_{ji}^b \right) \quad (\text{A.1d})$$

$$= 0 + \left( \sum_{j \in S} \sum_{i \in N(j) \cap (V \setminus S)} \hat{f}_{ij}^b - 0 \right) \quad (\text{A.1e})$$

$$\leq 0 + \left( \sum_{i \in N(S)} \hat{f}^b(\delta^+(i)) - 0 \right) \quad (\text{A.1f})$$

$$= \sum_{i \in N(S)} \hat{f}^b(\delta^-(i)) - \sum_{i \in N(S)} \hat{x}_{ib} \quad (\text{A.1g})$$

$$= \sum_{i \in N(S)} \hat{f}^b(\delta^-(i)) - 0 \leq 0. \quad (\text{A.1h})$$

Equation (A.1a) holds by Constraints (2b). Equation (A.1d) holds because, in the left sum, each flow variable  $\hat{f}_{ij}^b$  with  $i, j \in S$  appears once with a positive coefficient and once with a negative coefficient, so they cancel each other. Equation (A.1e) holds by Constraints (2c) and (2e), because  $\hat{x}_{ib} = 0$  for every vertex  $i \in N(S)$ . Inequality (A.1f) holds by Constraints (2e). Equation (A.1g) holds by Constraints (2b). The equation in Line (A.1h) holds because  $\hat{x}_{ib} = 0$  for every vertex  $i \in N(S)$ . The inequality in Line (A.1h) then holds by Constraints (2c). This results in the contradiction  $1 \leq 0$ .  $\square$

## Appendix B. NP-Hardness of Fractional Separation for LCUT

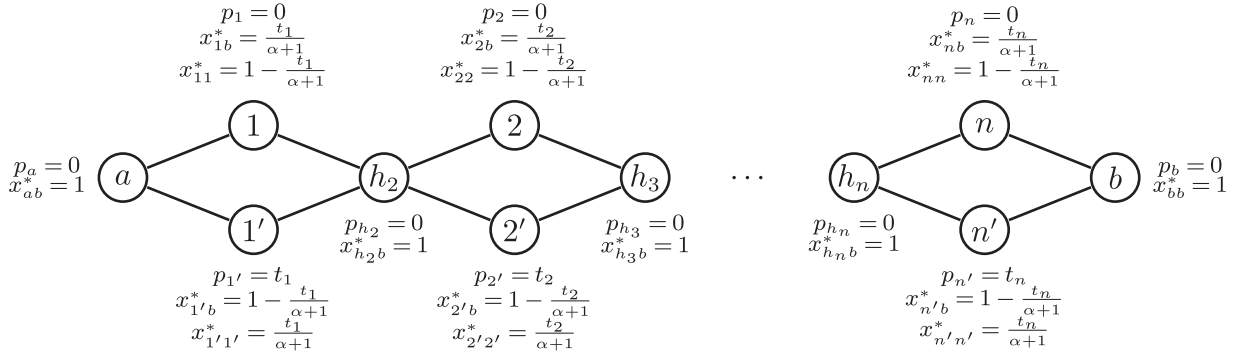
To prove Theorem 3, we require the following lemma. It refers to the PARTITION problem, in which positive integers  $t_1, t_2, \dots, t_n$  are given as input, and the task is to determine whether there exists  $T \subseteq [n]$  with  $\sum_{i \in T} t_i = \sum_{i \in [n] \setminus T} t_i$ .

**Lemma B.1.** *PARTITION remains NP-hard when two of the integers in the input equal  $(3/8) \sum_{i=1}^n t_i$ .*

**Proof.** The reduction is from an arbitrary instance of PARTITION with positive integers  $s_1, s_2, \dots, s_q$ . Let  $\sigma = 1/2 \sum_{i=1}^q s_i$  be the associated target value, and let  $n := q + 2$ . We construct an equivalent instance of partition  $t_1, t_2, \dots, t_n$ , where  $t_i := s_i$  for every  $i \in [q]$ , and the last two integers  $t_{q+1}$  and  $t_{q+2}$  are each set to  $3\sigma$ . The sum  $\sum_{i=1}^n t_i$  equals  $8\sigma$ , meaning that the new target value is  $4\sigma$ . This implies that  $t_{q+1}$  and  $t_{q+2}$  cannot be on the same side of the partition, in which case the new instance of PARTITION is equivalent to the original one. Because the reduction runs in polynomial time and because  $t_{q+1} = t_{q+2} = 3\sigma = (3/8) \sum_{i=1}^n t_i$ , this proves the lemma.  $\square$

**Proof of Theorem 3.** We show that it is NP-complete to determine whether a given point  $x^*$  lies outside of  $\mathcal{P}_{\text{LCUT}}$ .

**Figure B.1.** Graph  $G = (V, E)$ , Population Vector  $p$ , and Nonzeros of  $x^*$



This problem belongs to NP because a violated inequality from  $\mathcal{P}_{\text{LCUT}}$  is a suitable witness. The hardness reduction is from a PARTITION instance from the special class described in Lemma 5. Without loss, suppose that  $t_1 = t_2 = \frac{3}{4}\alpha$ , where  $\alpha := 1/2 \sum_{i=1}^n t_i$  is the target value, and that  $\alpha \geq 17$  (otherwise it is solvable in polynomial time by dynamic programming). Figure B.1 details the construction of the graph  $G = (V, E)$ , the population vector  $p$ , and the nonzeros of the point  $x^*$ . Also, set  $L := 0$ ,  $U := \alpha - 1$ , and  $k := n + 1$ . The graph  $G$  has  $3n + 1$  vertices and is outerplanar.

First, we show that  $x^*$  belongs to  $\mathcal{P}_{\text{CUT}}$ . For every vertex  $i \in V$ ,  $\sum_{j \in V} x_{ij}^* = 1$ , so the assignment Constraints (1b) are satisfied. Furthermore,  $\sum_{j \in V} x_{ij}^* = n + 1 = k$ , so Constraint (1c) is satisfied. The population Constraints (1d) are clearly satisfied for  $a$ , the  $h_i$  vertices, and the top-most vertices, so consider a vertex of the type  $v'$  on the bottom, and see that

$$Lx_{v'v'}^* = 0 \left( \frac{t_v}{\alpha + 1} \right) \leq \sum_{i \in V} p_i x_{iv'}^* = t_v \left( \frac{t_v}{\alpha + 1} \right) \leq (\alpha - 1) \left( \frac{t_v}{\alpha + 1} \right) = Ux_{v'v'}^*,$$

where the last inequality holds by  $t_v \leq \frac{3}{4}\alpha$  and  $\alpha \geq 4$ . For vertex  $b$ , we have

$$\begin{aligned} Lx_{bb}^* &= 0(1) \leq \sum_{i \in V} p_i x_{ib}^* = \sum_{i=1}^n p_{i'} x_{i'b}^* \\ &= \sum_{i=1}^n t_i \left( 1 - \frac{t_i}{\alpha + 1} \right) \\ &= \sum_{i=1}^n t_i - \left( \frac{1}{\alpha + 1} \right) \sum_{i=1}^n t_i^2 \\ &< \sum_{i=1}^n t_i - \left( \frac{1}{\alpha + 1} \right) (t_1^2 + t_2^2) \\ &= 2\alpha - \left( \frac{1}{\alpha + 1} \right) \left( \frac{9}{8} \alpha^2 \right) \\ &\leq (\alpha - 1)(1) = Ux_{bb}^*. \end{aligned}$$

The last inequality holds because  $\alpha \geq 17$ . It is clear that  $x^*$  satisfies the coupling Constraints (1e). The  $i, j$ -separator Inequalities (8b) obviously hold for most variables  $x_{ij}^*$  because most of them are zero or have  $i = j$ . The only nontrivial case is when  $j = b$ . In this case, every minimal  $i, b$ -separator is of the form  $C = \{v, v'\}$  or  $C = \{h_v\}$ , and both have  $x_{ib}^* \leq 1 = \sum_{c \in C} x_{cb}^*$ . Therefore,  $x^*$  satisfies all Inequalities (8b) and thus satisfies all constraints defining  $\mathcal{P}_{\text{CUT}}$ .

Now, to show NP-hardness, we must demonstrate that  $(t_1, t_2, \dots, t_n)$  is a “yes” instance of PARTITION if and only if there is an inequality defining  $\mathcal{P}_{\text{LCUT}}$  that  $x^*$  violates.

( $\Rightarrow$ ) Suppose that  $(t_1, t_2, \dots, t_n)$  is a “yes” instance of PARTITION, meaning that there is a subset  $C \subset [n]$  such that  $\sum_{i \in C} t_i = \sum_{i \in [n] \setminus C} t_i$ . We argue that  $C$  is a length- $U$   $a, b$ -separator in  $G$  and that  $x^*$  violates the associated inequality. First, see that if a vertex  $i \in C$  is removed from  $G$ , then any  $a, b$ -path must cross its copy  $i'$ , which will contribute  $p_{i'}$  toward the length of the path. Thus, if  $C$  is removed, then every  $a, b$ -path will have length at least  $\sum_{i \in C} p_{i'} = \sum_{i \in C} t_i = \alpha = U + 1$ . That is,  $C$  is a length- $U$   $a, b$ -separator. Then,  $x^*$  lies outside of  $\mathcal{P}_{\text{LCUT}}$  because

$$x_{ab}^* = 1 > \frac{\alpha}{\alpha + 1} = \frac{1}{\alpha + 1} \sum_{i \in C} t_i = \sum_{i \in C} \frac{t_i}{\alpha + 1} = \sum_{i \in C} x_{ib}^*.$$

( $\Leftarrow$ ) For the other direction, suppose that there is an inequality defining  $\mathcal{P}_{\text{LCUT}}$  that  $x^*$  violates. As we have shown,  $x^*$  belongs to  $\mathcal{P}_{\text{CUT}}$  (and thus  $\mathcal{P}_{\text{HESS}}$ ), so any violated inequality must take the form  $x_{ij} \leq \sum_{c \in C} x_{cj}$  where  $i \neq j$ . Observe that  $C$  must be a length- $U$   $i, j$ -separator and not just an  $ij$ -separator (by  $x^* \in \mathcal{P}_{\text{CUT}}$ ), so there exists an  $ij$ -path in  $G - C$ , but its length is more than  $U$ . Without loss of generality, we suppose that  $C$  is a minimal length- $U$   $ij$ -separator.

**Claim B.1.** We have  $j = b$ .

**Proof.** The claim follows because if  $j \neq b$ , then  $x_{ij}^* = 0 \leq \sum_{c \in C} x_{cj}^*$ . ■

**Claim B.2.** Separator  $C$  contains no vertices from the middle row  $\{a, h_2, h_3, \dots, h_n, b\}$ .

**Proof.** If  $C$  contains a vertex  $v$  from the middle row, then  $x_{ib}^* \leq 1 = x_{vb}^* \leq \sum_{c \in C} x_{cb}^*$ , and  $x^*$  satisfies the length- $U$   $i, b$ -separator inequality for  $C$ . ■

**Claim B.3.** Separator  $C$  contains no vertices from the bottom row  $\{1', 2', \dots, n'\}$ .

**Proof.** For contradiction purposes, suppose that  $C$  contains a vertex  $v'$  from the bottom row. Let  $C' := C \setminus \{v'\}$  and consider a shortest  $i, b$ -path  $P'$  in  $G - C'$ . This path  $P'$  cannot cross  $v'$ , for if it did then replacing  $v'$  by its upper counterpart  $v$  gives a shorter path  $P$  of length

$$\text{length}(P) = \text{length}(P') + p_v - p_{v'} = \text{length}(P') + 0 - t_v < \text{length}(P').$$

Thus,  $P'$  also belongs to  $G - C$ , implying that  $\text{dist}_{G-C,p}(i, b) = \text{dist}_{G-C,p}(i, b)$ . Recalling that  $\text{dist}_{G-C,p}(i, b) > U$  as  $C$  is a length- $U$   $i, b$ -separator, we see  $\text{dist}_{G-C,p}(i, b) = \text{dist}_{G-C,p}(i, b) > U$ . Consequently,  $C'$  is also a length- $U$   $i, b$ -separator, contradicting the minimality of  $C$ . ■

**Claim B.4.** The inequalities  $\alpha \leq p_i + \sum_{c \in C} t_c$  and  $(\alpha + 1) \sum_{c \in C} x_{cb}^* \geq (\alpha - p_i)$  hold.

**Proof.** By Claims B.2 and B.3,  $C \subseteq [n]$ . By minimality of  $C$ ,  $i$  must be somewhere to the left of the  $C$  vertices in Figure B.1. Therefore, every  $i, b$ -path must cross  $i$  and the lower counterparts  $C'$  of  $C$  and thus has length at least  $p_i + p(C')$ . Meanwhile, a path  $P$  of the same length can be constructed by moving from left to right, opting for  $v \in [n]$  when it is available ( $v \notin C$ ) and taking  $v'$  otherwise. Thus,  $P$  is a shortest  $i, b$ -path in  $G - C$ , and

$$\begin{aligned} \alpha &= U + 1 \leq \text{dist}_{G-C,p}(i, b) = p_i + p(C') \\ &= p_i + \sum_{c \in C} t_c = p_i + (\alpha + 1) \sum_{c \in C} x_{cb}^*. \quad \blacksquare \end{aligned}$$

**Claim B.5.** Vertex  $i$  cannot belong to the top row  $\{1, 2, \dots, n\}$ .

**Proof.** If  $i \in \{1, 2, \dots, n\}$ , we arrive at the following contradiction.

$$(3/4)\alpha \geq t_i = (\alpha + 1)x_{ib}^* > (\alpha + 1) \sum_{c \in C} x_{cb}^* \geq \alpha > (3/4)\alpha.$$

The first inequality holds by special class of PARTITION instances that we reduce from. The second inequality holds by the assumption. The third inequality holds by Claim B.4 and  $p_i = 0$ . ■

**Claim B.6.** We have that  $(t_1, t_2, \dots, t_n)$  is a “yes” instance of PARTITION.

**Proof.** By Claim B.5, either  $i \in \{a, h_2, \dots, h_n\}$  or  $i \in \{1', 2', \dots, n'\}$ . In the former case,

$$1 = x_{ib}^* > \sum_{c \in C} x_{cb}^* = \sum_{c \in C} \frac{t_c}{\alpha + 1}, \quad (\text{B.1})$$

implying that  $\alpha + 1 > \sum_{c \in C} t_c$  and so

$$\alpha + 1 > \sum_{c \in C} t_c = \sum_{c \in C} t_c + p_i \geq \alpha, \quad (\text{B.2})$$

where the last inequality holds by Claim B.4. Because all terms of (B.2) are integers, it follows that  $\sum_{c \in C} t_c = \alpha$ . Thus,  $C$  is a solution to the partition instance. In the latter case,  $i$  belongs to the bottom row  $\{1', 2', \dots, n'\}$ . Let  $j \in \{1, 2, \dots, n\}$  be its upper counterpart. Then,

$$1 - \frac{t_j}{\alpha + 1} = x_{ib}^* > \sum_{c \in C} x_{cb}^* = \sum_{c \in C} \frac{t_c}{\alpha + 1}. \quad (\text{B.3})$$

implying that  $\alpha + 1 - t_j > \sum_{c \in C} t_c$  and so

$$\alpha + 1 > \sum_{c \in C} t_c + t_j = \sum_{c \in C} t_c + p_i \geq \alpha, \quad (\text{B.4})$$

where the last inequality holds by Claim B.4. Because all terms of (B.4) are integers, it follows that  $\sum_{c \in C} t_c + t_j = \alpha$ . Thus,  $C \cup \{j\}$  is a solution to the partition instance. ■

## References

Adler WT, Wang SS-H (2019) Response to Cho and Liu, “Sampling from complicated and unknown distributions: Monte Carlo and Markov chain Monte Carlo methods for redistricting. *Phys. A* 516:591–593.

Ahn N, Park S (2015) An optimization algorithm for the minimum  $k$ -connected  $m$ -dominating set problem in wireless sensor networks. *Wireless Networks* 21(3):783–792.

Altman M (1997) The computational complexity of automated redistricting: Is automation the answer? *Rutgers Comput. Technol. Law J.* 23:81.

Altman M (1998) Traditional districting principles: Judicial myths vs. reality. *Soc. Sci. History* 22(2):159–200.

Altman M, McDonald M (2010) The promise and perils of computers in redistricting. *Duke J. Constitutional Law Public Policy* 5: 69–111.

Altman M, McDonald MP (2011) BARD: Better automated redistricting. *J. Statist. Software* 42(4):1–28.

Altman M, McDonald M (2018) Redistricting by formula: An Ohio reform experiment. *Amer. Political Res.* 46(1):103–131.

Álvarez-Miranda E, Sinnl M (2017) A relax-and-cut framework for large-scale maximum weight connected subgraph problems. *Comput. Oper. Res.* 87:63–82.

Álvarez-Miranda E, Ljubić I, Mutzel P (2013a) The maximum weight connected subgraph problem. *Facets of Combinatorial Optimization* (Springer, Berlin), 245–270.

Álvarez-Miranda E, Ljubić I, Mutzel P (2013b) The rooted maximum node-weight connected subgraph problem. *Proc. Internat. Conf. AI OR Techniques Constraint Programming Combinatorial Optim. Problems* (Springer, Berlin), 300–315.

Applegate D (2019) South Carolina districting, July 16.

Arrington TS (2010) Redistricting in the US: A review of scholarship and plan for future research. *Forum* 8(2). <https://www.degruyter.com/document/doi/10.2202/1540-8884.1351/html>.

Arslan O, Jabali O, Laporte G (2020) A flexible, natural formulation for the network design problem with vulnerability constraints. *INFORMS J. Comput.* 32(1):120–134.

Arslan O, Karşan OE, Mahjoub AR, Yaman H (2019) A branch-and-cut algorithm for the alternative fuel refueling station location problem with routing. *Transportation Sci.* 53(4):1107–1125.

Beasley JE (1993) Lagrangean heuristics for location problems. *Eur. J. Oper. Res.* 65(3):383–399.

Boehm E (2018) Gerrymandering is out of control. *Reason*, March 27, <https://reason.com/archives/2018/03/27/gerrymandering-is-out-of-control>.

Bozkaya B, Erkut E, Laporte G (2003) A tabu search heuristic and adaptive memory procedure for political districting. *Eur. J. Oper. Res.* 144(1):12–26.

Briant O, Naddef D (2004) The optimal diversity management problem. *Oper. Res.* 52(4):515–526.

Buchanan A (2019) Districting, July 29.

Buchanan A, Sung JS, Butenko S, Pasiliao EL (2015) An integer programming approach for fault-tolerant connected dominating sets. *INFORMS J. Comput.* 27(1):178–188.

Bullock CS III (2010) *Redistricting: The Most Political Activity in America* (Rowman & Littlefield Publishers).

Bureau of the Census (2012) TIGER/Line shapefiles. Accessed January 23, 2019, [https://www2.census.gov/geo/tiger/TIGER2010DP1/Profile-County\\_Tract.zip](https://www2.census.gov/geo/tiger/TIGER2010DP1/Profile-County_Tract.zip).

Bureau of the Census (2018) Census tracts for the 2020 census: Final criteria. *Federal Register* 83(219):56277–56284.

Caro F, Shirabe T, Guignard M, Weintraub A (2004) School redistricting: Embedding GIS tools with integer programming. *J. Oper. Res. Soc.* 55(8):836–849.

- Carvajal R, Constantino M, Goycoolea M, Vielma JP, Weintraub A (2013) Imposing connectivity constraints in forest planning models. *Oper. Res.* 61(4):824–836.
- Cho WKT, Liu YY (2018) Sampling from complicated and unknown distributions: Monte Carlo and Markov chain Monte Carlo methods for redistricting. *Phys. A* 506:170–178.
- Cohen-Addad V, Klein PN, Young NE (2018) Balanced centroidal power diagrams for redistricting. *Proc. 26th ACM SIGSPATIAL Internat. Conf. Adv. Geographic Inform. Systems (ACM)*, 389–396.
- Daskin MS, Tucker EL (2018) The trade-off between the median and range of assigned demand in facility location models. *Internat. J. Production Res.* 56(1-2):97–119.
- DeFord D, Duchin M, Solomon J (2021) Recombination: A family of Markov chains for redistricting. *Harvard Data Sci. Rev.* 3(1). <https://hdsr.mitpress.mit.edu/pub/1ds8ptxu/release/4/readingCollection=a133a0a2>.
- Di Cortona PG, Manzi C, Pennisi A, Ricca F, Simeone B (1999) *Evaluation and Optimization of Electoral Systems*, vol. 1 of Monographs on Discrete Mathematics and Applications (SIAM, Philadelphia).
- Drexler A, Haase K (1999) Fast approximation methods for sales force deployment. *Management Sci.* 45(10):1307–1323.
- Duchin M (2018) Political geometry: voting districts, “compactness,” and ideas about fairness. MAA-AMS-SIAM Gerald and Judith Porter Public Lecture given at the 2018 Joint Mathematics Meetings. Video of talk available at <https://www.youtube.com/watch?v=VddLOeov7QY>.
- Duchin M (2020) Dealing with non-contiguous tracts, January 8.
- Duchin M, Gladkova T, Henninger-Voss E, Ben Klingensmith HN, Wheelen H (2019) Locating the representational baseline: Republicans in Massachusetts. *Election Law J. Rules Political Policy* 18(4):388–401.
- Duque JC, Church RL, Middleton RS (2011) The  $p$ -regions problem. *Geographic Anal.* 43(1):104–126.
- Fifield B, Higgins M, Imai K, Tarr A (2015) A new automated redistricting simulator using Markov chain Monte Carlo. Working paper, Princeton University, Princeton, NJ.
- Fischetti M, Leitner M, Ljubić I, Luipersbeck M, Monaci M, Resch M, Salvagnin D, et al. (2017) Thinning out Steiner trees: A node-based model for uniform edge costs. *Math. Programming Comput.* 9(2):203–229.
- Ford LR Jr, Fulkerson DR (1962) *Flows in Networks* (Princeton University Press, Princeton, NJ).
- Gamrath G, Koch T, Maher SJ, Rehfeldt D, Shinano Y (2017) SCIP-Jack: A solver for STP and variants with parallelization extensions. *Math. Programming Comput.* 9(2):231–296.
- Garfinkel RS, Nemhauser GL (1970) Optimal political districting by implicit enumeration techniques. *Management Sci.* 16(8):B–495.
- Gentry S, Chow E, Massie A, Segev D (2015) Gerrymandering for justice: Redistricting US liver allocation. *Interfaces* 45(5):462–480.
- Goderbauer S, Winandy J (2018) Political districting problem: Literature review and discussion with regard to federal elections in Germany. Working paper, RWTH Aachen University, Aachen, Germany.
- Gopalan R, Kimbrough SO, Murphy FH, Quintus N (2013) The Philadelphia districting contest: Designing territories for city council based upon the 2010 census. *Interfaces* 43(5):477–489.
- Grofman B (1985) Criteria for districting: A social science perspective. *UCLA Law Rev.* 33:77.
- Guo D, Jin H (2011) iRedistrict: Geovisual analytics for redistricting optimization. *J. Visual Language Comput.* 22(4):279–289.
- Gurobi (2017) Gurobi guidelines for numerical issues. Accessed January 9, 2020, <http://files.gurobi.com/Numerics.pdf>.
- Gutiérrez-Andrade MA, Rincón-García EA, de-los Cobos-Silva SG, Lara-Velázquez P, Mora-Gutiérrez BA, Ponsich A (2019) Simulated annealing and artificial bee colony for the redistricting process in Mexico. *INFORMS J. Appl. Analytics* 49(3):189–200.
- Haase K, Müller S (2014) Upper and lower bounds for the sales force deployment problem with explicit contiguity constraints. *Eur. J. Oper. Res.* 237(2):677–689.
- Hebert GJ, Vandenberg ME, Smith P (2010) *The Realist’s Guide to Redistricting: Avoiding the Legal Pitfalls* (American Bar Association).
- Henzinger MR, Klein P, Rao S, Subramanian S (1997) Faster shortest-path algorithms for planar graphs. *J. Comput. System Sci.* 55(1):3–23.
- Hess SW, Weaver JB, Siegfeldt HJ, Whelan JN, Zitlau PA (1965) Nonpartisan political redistricting by computer. *Oper. Res.* 13(6):998–1006.
- Hojati M (1996) Optimal political districting. *Comput. Oper. Res.* 23(12):1147–1161.
- IOGP (2019) EPSG data set. Accessed January 23, 2019, <http://www.epsg.org/>.
- Kaplan H, Nussbaum Y (2011) Maximum flow in directed planar graphs with vertex capacities. *Algorithmica* 61(1):174–189.
- Kappel F, Kuntsevich AV (2000) An implementation of Shor’s  $r$ -algorithm. *Comput. Optim. Appl.* 15(2):193–205.
- Kim MJ (2019) Give-and-take heuristic model to political redistricting problems. *Spatial Inform. Res.* 27:539–552.
- Kim M, Xiao N (2017) Contiguity-based optimization models for political redistricting problems. *Internat. J. Appl. Geospatial Res.* 8(4):1–18.
- King DM, Jacobson SH, Sewell EC (2015) Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning. *Math. Programming* 149(1-2):425–457.
- King DM, Jacobson SH, Sewell EC (2018) The geo-graph in practice: Creating United States congressional districts from census blocks. *Comput. Optim. Appl.* 69(1):25–49.
- King DM, Jacobson SH, Sewell EC, Cho WKT (2012) Geo-graphs: An efficient model for enforcing contiguity and hole constraints in planar graph partitioning. *Oper. Res.* 60(5):1213–1228.
- Kong Y, Zhu Y, Wang Y (2019) A center-based modeling approach to solve the districting problem. *Internat. J. Geographic Inform. Sci.* 33(2):368–384.
- Lacki J, Nussbaum Y, Sankowski P, Wulff-Nilsen C (2012) Single source: All sinks max flows in planar digraphs. *Proc. IEEE 53rd Annual Sympos. on Foundations of Computer Science (IEEE, New York)*, 599–608.
- Levin HA, Friedler SA (2019) Automated congressional redistricting. *J. Experiment. Algorithmics* 24(1):1–10.
- Linskey A (2012) Foes of congressional map meet target. *The Baltimore Sun*, July 11, [http://articles.baltimoresun.com/2012-07-11/news/bs-md-congressional-redistrict-20120711\\_1\\_congressional-map-new-map-state-board](http://articles.baltimoresun.com/2012-07-11/news/bs-md-congressional-redistrict-20120711_1_congressional-map-new-map-state-board).
- Liu YY, Cho WKT, Wang S (2016) PEAR: A massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm Evolution Comput.* 30:78–92.
- Lykhovyd E (2019) Efficient implementation of Shor’s  $r$ -algorithm using MKL. Accessed January 9, 2020, <https://github.com/zhelih/ralg>.
- Mehrotra A, Johnson EL, Nemhauser GL (1998) An optimization based heuristic for political districting. *Management Sci.* 44(8):1100–1114.
- Mikhalevich VS, Shor NZ, Galustuva LA, Zhurbenko NG, Momot AI, Sibirko AN, Trubin VA, et al. (1977) *Computational Methods of Choice of Optimal Design Decisions* (Naukova Dumka).
- Miller S (2007) *The Problem of Redistricting: The Use of Centroidal Voronoi Diagrams to Build Unbiased Congressional Districts* (Whitman College).
- Murphy FH, Hess SW, Wong-Martinez CG (2013) Politics. Gass SI, Fu MC, eds. *Encyclopedia of Operations Research and Management Science* (Springer, Berlin), 1137–1141.
- NCSL (2019) Redistricting criteria. Accessed June 20, 2019, <http://www.ncsl.org/research/redistricting/redistricting-criteria.aspx>.

- Niemi RG, Grofman B, Carlucci C, Hofeller T (1990) Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *J. Politics* 52(4): 1155–1181.
- Oehrlein J, Haunert J-H (2017) A cutting-plane method for contiguity-constrained spatial aggregation. *J. Spatial Inform. Sci.* 2017(15):89–120.
- Olson B (2019) Impartial automatic redistricting. Accessed June 21, 2019, <https://bdistricting.com/2010/>.
- Rehfeldt D, Koch T (2019) Combining NP-hard reduction techniques and strong heuristics in an exact algorithm for the maximum-weight connected subgraph problem. *SIAM J. Optim.* 29(1):369–398.
- Rehfeldt D, Koch T, Maher SJ (2019) Reduction techniques for the prize collecting Steiner tree problem and the maximum-weight connected subgraph problem. *Networks* 73(2):206–233.
- Resende MGC, Werneck RF (2004) A hybrid heuristic for the  $p$ -median problem. *J. Heuristics* 10(1):59–88.
- Ricca F, Simeone B (2008) Local search algorithms for political districting. *Eur. J. Oper. Res.* 189(3):1409–1426.
- Ricca F, Scozzari A, Simeone B (2008) Weighted Voronoi region algorithms for political districting. *Math. Comput. Modeling* 48(9–10):1468–1477.
- Ricca F, Scozzari A, Simeone B (2013) Political districting: From classical models to recent approaches. *Ann. Oper. Res.* 204(1):271–299.
- Salemi H, Buchanan A (2020) Parsimonious formulations for low-diameter clusters. *Math. Programming Comput.* 12(3):493–528.
- Shirabe T (2005) A model of contiguity for spatial unit allocation. *Geographic Anal.* 37(1):2–16.
- Shirabe T (2009) Districting modeling with exact contiguity constraints. *Environmental Planning B: Planning Design* 36(6):1053–1066.
- Shor NZ (1985) *Minimization Methods for Non-Differentiable Functions*, vol. 3 of Springer Series in Computational Mathematics (Springer, Berlin).
- Svec L, Burden S, Dilley A (2007) Applying Voronoi diagrams to the redistricting problem. *UMAP J.* 28(3):313–329.
- Swamy R, King DM, Jacobson SH (2019a) A case for transparency in the design of political districts. Working paper, University of Illinois at Urbana-Champaign, Urbana, Illinois.
- Swamy R, King DM, Jacobson SH (2019b) Multi-objective optimization for political districting: a scalable multilevel approach. Working paper.
- Validi H, Buchanan A (2019) A note on “A linear-size zero-one programming model for the minimum spanning tree problem in planar graphs.” *Networks* 73(1):135–142.
- Validi H, Buchanan A (2020) The optimal design of low-latency virtual backbones. *INFORMS J. Comput.* 32(4):952–967.
- Vickrey W (1961) On the prevention of gerrymandering. *Political Sci. Quart.* 76(1):105–110.
- Wang Y, Buchanan A, Butenko S (2017) On imposing connectivity constraints in integer programs. *Math. Programming* 166(1–2): 241–271.
- Young PH (1988) Measuring the compactness of legislative districts. *Legislative Stud. Quart.* 13(1):105–115.
- Zoltners AA, Sinha P (1983) Sales territory alignment: A review and model. *Management Sci.* 29(11):1237–1256.

---

**Hamidreza Validi** is a postdoctoral research associate in the Department of Computational and Applied Mathematics at Rice University. He recently earned his PhD in industrial engineering and management from Oklahoma State University. He is broadly interested in combinatorial optimization, integer programming, and network optimization.

**Austin Buchanan** is an associate professor of industrial engineering and management at Oklahoma State University. His research focuses on solving combinatorial optimization problems in networks, particularly those having connectivity or distance constraints. He currently serves as an associate editor for the journals *Networks* and *Optimization Letters* and has served in various leadership roles in the INFORMS Optimization Society and the INFORMS Section on Telecommunications and Network Analytics.

**Eugene Lykhovyd** is a quantitative researcher associate at JPMorgan Chase & Co. in New York, NY. He recently earned his PhD in industrial and systems engineering from Texas A&M University. He is interested in combinatorial optimization, nondifferentiable optimization, and financial modeling.