5 A BRANCH-AND-PRICE ALGORITHM FOR THE p-MEDIAN DISTRICTING PROBLEM – ONGOING AND PLANNED WORK

Abstract. The p-Median Districting Problem is the most prevalent problem in districting domains. Given a planar graph G, it asks for a p-partition of G into connected and balanced components called districts, such that the partition's compactness, given by its p-median cost, is minimized. In this chapter we present our preliminary work on an exact branch-and-price algorithm to solve this problem. It uses a set partitioning formulation over the set of all feasible districts, whose linear relaxation is known to yield tighter dual bounds than standard formulations. We extend the previous method of Mehrotra et al. [155], who solved this model with column generation within a branch-and-price framework, with several improvements. These include an exact solution to the pricing subproblem, Lagrangean-based dual bounding and variable fixing techniques, early branching strategies and an improved primal solution. Experiments show that our method produces better root lower bounds than other formulations, but that column generation struggles with convergence and degeneracy, which ultimately make the proposed branch-and-price worse than existing solutions. We discuss how we intend to address these issues.

5.1 Introduction

In this chapter we present our ongoing work on an exact branch-and-price algorithm for the p-Median Districting Problem (PMDP). We have defined the PMDP in Section 2.3.2. We chose this problem for three reasons. First, it is also considered by Mehrotra et al. [155], Salazar-Aguilar et al. [197] and Validi et al. [222], who have broken new grounds in districting problems from a MIP perspective and whose methods we build upon. Second, as we saw in Section 2.3.2 p-median as a compactness function has been used widely across several domains, and has a number of desired properties such as favoring connectivity, inducing low routing costs, and being responsive to changes. Third, the problem has no domain-specific constraints, which allows us to strive for generality and maintain our focus on the methods rather than the application. For this reason, here we also consider a single balancing attribute.

We consider the set partitioning formulation of Garfinkel and Nemhauser [78], which we have introduced in Section 2.4.1.3. Rather than an assignment of units to medians, as the Hess model (2.10), it seeks a partition of V from elements of the set $\mathcal{F} = \{V' \subseteq V \mid \text{connected}(V') \wedge L \leqslant w(V') \leqslant U\}$ of connected and balanced districts. (Recall that L and U are the lower and upper balancing limits.) Let variables $z_j \in \{0,1\}^{\mathcal{F}}$ be such that

$$z_j = \begin{cases} 1, & \text{if district } j \in \mathcal{F} \text{ is selected in the final districting plan,} \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

and let $c_j = C_{pm}(j) = \min_{c \in j} \sum_{i \in j} d_{ic}$ be the p-median cost function for district $j \in \mathcal{F}$. The set partitioning model is then defined as:

$$\begin{aligned}
\textbf{minimize} \quad & \sum_{j \in \mathcal{F}} z_j c_j && && (5.2a)\\
\textbf{subject to} \quad & \sum_{j \in \mathcal{F}} z_j = p, && && (5.2b)\\
& \sum_{j \in \mathcal{F} \mid i \in j} z_j = 1 && \forall i \in V, && (5.2c)\\
& z_j \in \{0,1\} && \forall j \in \mathcal{F}. && (5.2d)
\end{aligned}$$

Although this formulation has been used by several authors [78, 155, 46, 17, 99, 161], nearly all proposed methods either consider a heuristically-generated set of districts in place of $\mathcal{F}$, or use heuristic pruning rules to solve the model, which makes the methods not exact. To our knowledge, the only exact method on a related formulation is from Bender et al. [16], who consider a multi-period districting problem and solve it through branch-and-price. However, in that problem the union of all districts is not required to be a partition of V, and districts need not be connected, which makes it fundamentally different from the PMDP.

Mehrotra et al. [155] have used Garfinkel and Nemhauser [78]'s model in a branch-and-price to solve the PMDP. Because the model has exponentially many variables, they solved its linear relaxation by column generation. However, because the pricing subproblem of finding a minimum-cost, connected and balanced district is computationally difficult, Mehrotra et al. [155] solved a simplified version of it which made their branch-and-price not exact.

In this chapter, we revisit the approach of Mehrotra et al. [155] with the goal of proposing an exact branch-and-price method. Since their work, a number of techniques have become available which we believe make this possible. They include:

- efficient ways to enforce connectivity in districting models, studied by Salazar-Aguilar et al. [197] and Validi et al. [222] and which make the pricing subproblem tractable;
- the Lagrangean-based variable fixing approach of Validi et al. [222], which fixes about 90% of variables in the Hess model for instances of fewer than 1000 units, and further eases the burden of the pricing subproblem;
- high-quality heuristic solvers for the PMDP proposed by Validi et al. [222], Ríos-Mercado et al. [193] and ourselves in Chapter 3, which we use both as primal bounds for branch-and-price and to seed initial columns for column generation;
- techniques such as Lagrangean lower bounding for early termination, dual stabilization to mitigate the tailing-off effect, early branching and effective column management techniques, which have appeared as the literature on column generation matured [150, 223, 52]; and
- valid inequalities for knapsack constraints such as lifted cover inequalities, which have been studied at length [98] and could be used as cutting planes in a branch-and-cut-and-price method.

In the following sections we present in detail how we leverage some of these techniques to propose an improved, exact branch-and-price. Then, in Section 5.3 we report on early experiments that show that Model (5.2) produces much better lower bounds than that of Hess et al. [108]. Despite this, we find that column generation converges slowly and struggles with degeneracy, and therefore the proposed branch-and-price requires much more effort than a commercial solver running the Hess model. In Section 5.4 we present our plans to address this.

5.2 Proposed branch-and-price algorithm

We use the solution to the linear relaxation of Model (5.2), which we shall call the *linear master problem* (LMP), as dual bound in a branch-and-bound algorithm. The LMP can be obtained by substituting constraints (5.2d) with constraints $0 \leqslant z_j \leqslant$

$1\ \forall j \in \mathcal{F}$. Since $\mathcal{F}$ is exponentially-sized, we solve the LMP by column generation (Section 5.2.2).

Iteratively, we expand the next node in the branch-and-bound tree and solve its corresponding relaxation. If the optimal relaxed cost is higher than the cost of the incumbent, the node is fathomed. If the optimal solution to the LMP is fractional, we then proceed to generate two child branches that will solve subproblems with restricted domains. We explain our branching strategy in the next section. Otherwise, if it is not fractional, we update the incumbent and do not branch, since further branching cannot improve the solution. We expand nodes depth-first.

The initial incumbent solution is obtained by running the location-allocation heuristic of Validi et al. [222]. We chose this heuristic because it performed better in early tests for the real-world instances of Validi et al. [222] and Moreno et al. [161], which were our starting point. However, more experimental work is needed to determine if this heuristic is generally better than our heuristic of Chapter 3 or that of Ríos-Mercado et al. [193] for a broader instance set.

5.2.1 Branching

We use a variation of Ryan-Foster branching [196], similar to the one used by Mehrotra et al. [155]. It branches on the assignment of a pair of basic units $\{i, j\}$, with one branch forcing i and j to be assigned to the same district and the other forcing them to be assigned to different districts. Let B_{same} be the set of unit pairs which have been fixed to the same district, and B_{diff} be the set of unit pairs which have been fixed to different districts. In the Hess model, this would translate to cuts of the form

$$x_{ik} = x_{jk} \qquad \forall\{i, j\} \in B_{same},\ k \in V, \tag{5.3}$$

$$x_{ik} + x_{jk} \leqslant 1 \qquad \forall\{i, j\} \in B_{diff},\ k \in V. \tag{5.4}$$

Since they are in general not difficult to treat integrally, we include these cuts in the pricing subproblem rather than the RLMP. Namely, the following cuts are added to each pricing subproblem:

$$y_i = y_j \qquad \forall\{i, j\} \in B_{same} \tag{5.5}$$

$$y_i + y_j \leqslant 1 \qquad \forall\{i, j\} \in B_{diff}. \tag{5.6}$$

At the pricing subproblem rooted at unit k this leads to variable fixes $y_i = 1$ for each $\{i,k\} \in B_{same}$, and $y_i = 0$ for each $\{i,k\} \in B_{diff}$.

To choose units i and j to be branched on, given fractional solution $z^f \in \mathbb{R}^k$, Mehrotra et al. [155] selected first the unit i with the highest w_i in the most fractional column $s_1 \in [k]$. Then, they selected the most fractional column $s_2 \neq s_1$ such that $i \in s_2$, and chose some unit j such that $[j \in s_1] \neq [j \in s_2]$. Note that pair (s_1, s_2) must exist, since i) if the solution is not fractional no branching occurs and ii) if unit i is covered partially, constraints (5.2c) force multiple columns to cover i.

In the strategy above, units i and j are not necessarily neighbors in G. While B_{diff} fixes simply induce disjoint assignment constraints on the pricing subproblem, having a set of non-contiguous B_{same} fixes can make pricing significantly harder to solve, especially heuristically, since the selection of a unit v becomes conditioned to also finding a subset of V which connects $\{u \mid \{u,v\} \in B_{same}\}$. To mitigate this, we propose a branching strategy that branches only on neighboring units. Namely, given the most fractional column s_1, we branch on units i,j such that $i \in N(j)$, $i \in s_1$, $j \neq s_1$, and $w_i + w_j$ is maximum. Such a pair must always exist unless $p = 1$, since $\delta(S) \neq \emptyset$ for any $S \neq V$.

This strategy ensures that the subgraph induced by $V_{same} = \bigcup_{p \in B_{same}} p$ has no connected component of size 1. At any branch-and-bound node the current branching decisions therefore induce a modified graph $G_{rf} = (V_{rf}, E_{rf})$ that merges sets of units fixed to B_{same} into a single unit, and removes edges between units fixed to B_{diff}. Let C be the set of connected components of $G(V_{same})$. Namely, we have

$$V_{rf} = (V \setminus V_{same}) \cup \bigcup_{c \in C} v_c$$

for new units v_c associated with each component $c \in C$, and

$$E_{rf} = E(V_{rf}) \setminus B_{diff} \ \cup \ \{\{i, v_c\} \mid i \in V_{rf}, c \in C, \exists k \in c : \{i,k\} \in E \setminus B_{diff}\}.$$

For each new unit v_c, $c \in C$ we set its weight as $w_{v_c} = w(c)$, its distances to units $i \in V$ as $d_{iv_c} = d_{v_c i} = \sum_{k \in c} d_{ik}$, and distances to other units v'_c, $c' \in C$ as $d_{v_c, v'_c} = \sum_{k \in c} \sum_{k' \in c'} d_{kk'}$. After every branching decision set C is updated in amortized cost $O(\alpha(n))$ using a union-find data structure, where α is the inverse Ackermann function [74].

5.2.1.1 *Domain propagation and further pruning*

After branching we use domain propagation to try and generate additional cuts or prune the branch-and-bound node, when possible. To this end we apply, in order, the following tests.

First, if $w_i + w_j > U$ for any $\{i,j\} \in E_{rf}$, add $\{i,j\}$ to B_{diff}. Otherwise, this would lead to a district that would violate upper balancing constraints.

Second, for each articulation node $i \in V_{rf}$ let $A = \{a_1, \dots, a_{|A|}\}$ be the disjoint parts of the articulation such that, for every $\{a_j, a_k\} \subseteq A$, no path in G_{rf} exists from a unit in a_j to a unit in a_k that does not pass through i. Let also $A_L = \bigcup_{a \in A | w(a) < L} a$ be the set of units in parts that weigh less than L. In a feasible solution each such part must be joined with i. Therefore, if $w_i + w(A_L) > U$ or if $\{i,j\} \in B_{diff}$ for any $j \in A_L$, we prune this branch-and-bound node. Otherwise, for each $j \in A_L$ we add $\{i,j\}$ to B_{same}.

Third, note that graph G_{rf} is not necessarily connected, since B_{diff} branches remove edges from E_{rf}, and that no district can have units in two different connected components. Let $B = \{b_1, \dots, b_{|B|}\}$ be the set of connected components of G_{rf}. Then, a feasible solution must contain anywhere between $\lceil w(b)/U \rceil$ and $\lfloor w(b)/L \rfloor$ districts per connected component $b \in B$. This leads to the feasibility problem

$$
\begin{aligned}
\textbf{exists} \quad & x && (5.7a)\\
\textbf{subject to} \quad & \sum_{b \in B} x_b = p, && (5.7b)\\
& \lceil w(b)/U \rceil \leqslant x_b \leqslant \lfloor w(b)/L \rfloor \qquad \forall b \in B, && (5.7c)\\
& x \in \mathbb{N}^{|B|}, && (5.7d)
\end{aligned}
$$

with variable x_b denoting the number of districts placed in connected component b. If Model (5.7) is not feasible, then there is no way to distribute p districts over the components B, and the branch-and-bound node can be pruned. Model (5.7) can be solved by computing $\text{feasible}(0,0)$ in the recurrence

$$
\text{feasible}(i,j) = \begin{cases} \mathit{true} & \text{if } i = |B| \wedge j = p, \\ \mathit{false} & \text{if } i = |B| \wedge j \neq p, \\ \bigvee_{\lceil w(b_i)/U \rceil \leqslant k \leqslant \lfloor w(b_i)/L \rfloor} \text{feasible}(i+1, j+k) & \text{otherwise.} \end{cases} \tag{5.8}
$$

This can be done in $O(p|B|)$ time by dynamic programming on arguments i and j.

Unfortunately, in early experiments we found that these domain propagation techniques were not effective. In nearly all instances the granularity of unit weights is too fine compared to bounds L and U, and so these rules only activate at deep branches where B_{same} induces high weights and G_{rf} is connectivity-restricted by B_{diff}. However, branch-and-bound nodes seldom reached such depths, since they were either fathomed or not branched due to integrality of the LMP solution much earlier.

5.2.2 Solving the LMP with column generation

We use column generation to solve the LMP. For an overview on column generation, see Lübbecke and Desrosiers [150]. Iteratively, it solves the *restricted* linear master problem (RLMP) given by

$$\begin{aligned}
\textbf{minimize} \quad & \sum_{j\in J} z_j c_j && && \text{(5.9a)}\\
\textbf{subject to} \quad & \sum_{j\in J} z_j = p, && && \text{(5.9b)}\\
& \sum_{j\in J \mid i\in j} z_j = 1 && \forall i \in V, && \text{(5.9c)}\\
& 0 \leqslant z_j \leqslant 1 && \forall j \in J, && \text{(5.9d)}
\end{aligned}$$

which is defined over a set of feasible districts $J \subset \mathcal{F}$. Then, it searches for new columns (districts) of negative reduced cost from $\mathcal{F} \setminus J$ to include in J. If no such columns exist, then the optimal solution to the RLMP is also optimal for the LMP, and the algorithm stops.

Let π_0 be the dual variable associated with constraint (5.9b), and $\pi_1 \in \mathbb{R}^n$ be the dual variable vector associated with constraints (5.9c). Then, the reduced cost of including column $z_j, j \in J$ is given by $c_j - \sum_{i\in j} \pi_i^1 - \pi_0$. To find a column with minimum reduced cost, the pricing problem

$$\begin{aligned}
\underset{u\in V}{\textbf{minimize}} \quad & -\pi^0 + \sum_{i\in V\setminus\{u\}} y_i(d_{iu} - \pi_i^1) && && \text{(5.10a)}\\
\textbf{subject to} \quad & L \leqslant \sum_{i\in V} w_i y_i \leqslant U, && && \text{(5.10b)}\\
& y \text{ induces a connected subgraph of } G, && && \text{(5.10c)}\\
& y_u = 1, && && \text{(5.10d)}\\
& y_i \in \{0,1\} && \forall i \in V \setminus \{u\}, && \text{(5.10e)}
\end{aligned}$$

is solved. It consists of n independent subproblems, one for each $u \in V$, and its optimal solution can be obtained by taking the solution y^* of minimum value among the subproblems. If it has negative value, column generation adds the column $\{i \in V \mid y_i^* = 1\}$ to the RLMP.

Pricing does not need to be solved exactly at each iteration: as long as negative reduced cost columns are found, even if heuristically, column generation converges towards the optimal solution [52]. Therefore, to save time we adopt a four-staged approach to generating new columns:

1. to search for columns in a global column pool shared across all branch-and-bound nodes;
2. to search for columns using a simple semi-greedy multistart heuristic to look for columns quickly, without a significant time investment;
3. to run the matheuristic of Mehrotra et al. [155]'s, which solves Model (5.10) with restricted connectivity constraints;
4. to solve Model (5.10) optimally by branch-and-bound.

At any stage, if negative reduced cost columns are found, we add them to the RLMP and reoptimize it, then go back to stage 1. This is akin to a variable neighborhood search [103] through $\mathcal{P}(\mathcal{F})$ space, but without the randomization step. In this way, pricing is only solved optimally at the last step when all heuristics fail, or in order to prove that the RLMP is optimal by showing no more negative reduced cost columns exist. Solving the RLMP and thus updating duals π^0 and π^1 is usually a better approach than to solve pricing to optimality every time, since to find optimal solutions is a bottleneck. Whenever possible we add multiple new columns per iteration, which is generally recommended [150].

Algorithm 14 outlines our column generation method. In the following sections we describe how we obtain set J initially, then detail each of the four stages above. We also explain early branching and Lagrangean bounding techniques we have used.

As a note, we have also experimented with the covering version of Model (5.9), which substitutes constraints (5.9c) by $\sum_{j \in J \mid i \in j} z_j \geqslant 1 \ \forall i \in V$. However, in practice there was little difference in performance between the two formulations.

5.2.2.1 *Initial column set and managing infeasible columns*

Set J in the root node of branch-and-bound is initialized with the districts of the initial primal solution. If the primal solution is feasible, then the introduced columns will be a feasible solution to the RLMP since they are a p-partition of V. If it is not feasible, however, to induce feasibility we introduce p artificial columns $a_1, \dots, a_p$ such that $a_i = V$ and $c_{a_i} = \infty$. Because of their high cost, these columns are priced

Algorithm 14 Solving the LMP by column generation.

Input: a branch-and-bound node with column set J, and the current incumbent solution S.
Output: the LMP solution value.

if we are at the root node of branch-and-bound **then**
  $J \leftarrow \{S_i \mid i \in P\}$
$\mathcal{L} \leftarrow -\infty$
repeat
  solve Model (5.9) with columns J to obtain duals π^0, π^1 and upper bound υ
  if $\upsilon < C_{pm}(S)$ **then** ▷ early branching, see Section 5.2.5
    return $\mathcal{L}$
  $C \leftarrow \text{searchColumnPool}(\pi^0, \pi^1)$
  if $C = \emptyset$ **then**
    $C \leftarrow \text{greedyMultistart}(\pi^0, \pi^1)$
    if $C = \emptyset$ **then**
      $C \leftarrow \text{matheuristic}(\pi^0, \pi^1)$
      if $C = \emptyset$ **then**
        $C, \tau \leftarrow \text{exactPricing}(\pi^0, \pi^1)$
        if $C = \emptyset$ **then**
          return υ ▷ υ is optimal
        $\mathcal{L} \leftarrow \text{solveLagrangean}(\tau, \pi^0, \pi^1)$ ▷ see Section 5.2.3
        if $\mathcal{L} + \epsilon \geqslant \upsilon$ **or** $\mathcal{L} \geqslant C_{pm}(S)$ **then**
          return $\mathcal{L}$
        $\text{lagrangeanFixing}(\tau, \pi^0, \pi^1)$ ▷ see Section 5.2.3.1
  $J \leftarrow J \cup \{C\}$

out immediately in the first iterations of column generation.

Intermediate branch-and-bound nodes inherit the set of columns from their parent, except columns which violate the branching decision (5.5) or (5.6). Because instantiating data structures for the RLMP and the pricing IPs is computationally expensive, in our implementation upon branching one child receives the parent's data structures and the other child has them generated anew. In the former, violated columns z_j are then disabled by setting $c_j = \infty$, while in the latter they are never created. This leads to violated columns being removed from the model after every two branches, on average.

5.2.2.2 *Column pool*

We maintain a global column pool $\mathcal{P} \subseteq \mathcal{F}$ shared by all branch-and-bound nodes, comprising all negative reduced cost columns generated so far. At each iteration, before attempting to generate new columns we first scan $\mathcal{P}$ looking for columns with negative reduced cost w.r.t. current duals π^0, π^1 and are feasible w.r.t. the node's branching decisions. All such columns are then added to the RLMP.

For each column $j \in \mathcal{P}$ we cache its p-median cost $C_{pm}(j)$. Then, j's reduced cost $rc(j) = C_{pm}(j) - \pi^0 - \sum_{i \in j} \pi_i^1$ can be computed in $O(|j|)$ time. We further speed this up by keeping a hash table of the current column set J, and testing whether a candidate $j \in \mathcal{P}$ is in it. If this is the case, then by definition $rc(j) \geqslant 0$. While this test has the same worst-case runtime of $O(|j|)$, in practice it is worth doing. To test whether j respects branching decisions, we also maintain a boolean vector $h_i^j = [i \in j]$ for each $j \in \mathcal{P}$, so that feasibility can be checked in $O(|B_{same}| + |B_{diff}|)$ by testing whether $h_u^j = h_v^j$ for $\{u, v\} \in B_{same}$ and $h_u^j + h_v^j \leqslant 1$ for $\{u, v\} \in B_{diff}$.

5.2.2.3 *Greedy multistart heuristic*

If no more negative reduced cost columns are found in $\mathcal{P}$, we first look for new columns with a fast multistart semi-greedy heuristic. For each "root" unit $u \in V$ we repeat the heuristic $iter_{gr}$ times, and so it is executed $n \cdot iter_{gr}$ times in total, where $iter_{gr}$ is a parameter. Each time the heuristic results in a negative reduced cost column, we add it to the RLMP.

Given root unit $u \in V$, let $v \in V_{rf}$ be the unit which accounts for u in G_{rf}. Then, a greedy solution seeded at v is built as follows.

1. Start with solution $S = \{v\}$, exclusion set $B = \{i \mid \{i, v\} \in B_{diff}\}$, and candidate

set $K = N_{rf}(v)$. Here, $N_{rf}(v) = \{i \mid \{i, v\} \in E_{rf}\}$ denotes the neighborhood of v in G_{rf}, and we loosely extend the definition of B_{diff} to include units in G_{rf}.

2. Remove from K all units k such that either i) $k \in B$, ii) $w(S) + w_k > U$, or iii) $w(S) \geqslant L \wedge d_{kv} - \pi_k > 0$. Then, if $K = \emptyset$, stop and return S. Note that test iii) disallows adding positive cost units if the solution already weighs L, since they cannot improve the solution and the heuristic does not remove units from S, and that we stop adding units before $w(S)$ reaches U.
3. Move one randomly-chosen unit k such that $(f(k_{best}) - f(k))/f(k_{best}) \leqslant \alpha_{gr}$ from K to S, where $f(i) = (\pi_i - d_{iv})/w_i$, $k_{best} = \operatorname{argmin}_{k \in K} f(k)$, and α_{gr} is a parameter. Fitness function f gives preference to assignments with lowest reduced costs contribution over weight ratios, since they have the highest impact in the final reduced cost.
4. Add to K any neighbors of k which are not in $K \cup S$, and add to B any unit i such that $\{i, k\} \in B_{diff}$. Go to step 2.

The full method is shown in Algorithm 15. Note that connectivity is guaranteed, since we only consider the addition of neighbor units. In our implementation, we have fixed $\alpha_{gr} = 0.2$ and $iter_{gr} = 5$.

Algorithm 15 Greedy multistart heuristic for pricing.

Input: a dual vector $\pi \in \mathbb{R}^n$.

for $u \in V$ **do**
    for $iter_{gr}$ iterations **do**
        $v \leftarrow v_c : c \in C, u \in c$
        $S \leftarrow \{v\}$
        $K \leftarrow N_{rf}(v)$
        $B \leftarrow \{i \mid \{i, v\} \in B_{diff}\}$
        while $K \neq \emptyset$ **do**
            $k_{best} \leftarrow \operatorname{argmin}_{k \in K} f(k)$
            $k \leftarrow$ a random element from $\{k \in V \mid (f(k_{best}) - f(k))/f(k_{best}) \leqslant \alpha_{gr}\}$
            $S \leftarrow S \cup \{k\}$
            $B \leftarrow B \cup \{i \in V_{rf} \mid \{i, k\} \in B_{diff}\}$
            $K \leftarrow (K \cup N_{rf}(k)) \setminus (S \cup B)$
            $K \leftarrow K \setminus \{k \in V_{rf} \mid w(S) + w_k > U \vee (w(S) \geqslant L \wedge d_{kv} - \pi_k > 0)\}$
        add column S to the RLMP if $L \leqslant w(S) \leqslant U$.

5.2.2.4 Mehrotra et al. [155]'s matheuristic

If the greedy heuristic above does not find columns of negative reduced cost, in a second step we run the matheuristic proposed by Mehrotra et al. [155] for the pricing problem. It solves by branch-and-bound Model (5.10) with connectivity constraints (5.10c) replaced by tree-like constraints

$$y_i \leqslant \sum_{k \in N(i) | s_{ku} < s_{iu}} y_k \qquad \forall i \in V \setminus \{u\} \tag{5.11}$$

for given center unit u. Here, s_{ij} denotes the length of the shortest path (in hops) from unit i to unit u in G. These constraints ensure that unit $i \in V$ can only be selected if some neighbor $k \in N(i)$ closer to u than i is also selected, and thus force solutions to be shortest-path trees rooted at u. Since Model (5.10) decomposes into n independent subproblems, one for each u, we solve the subproblems separately and add to the RLMP all solutions which have negative reduced cost. Since columns of non-negative reduced cost are not useful, during branch-and-bound we prune any nodes with non-negative lower bounds.

Because constraints (5.11) force new columns to be shortest-path trees from u, they may exclude feasible columns which are not. This leads to a restricted problem that is significantly easier than solving Model (5.10) with e.g. flow or cutset connectivity constraints. In practice, we find that this restricted model is better at generating negative reduced cost columns than the greedy heuristic, but at a higher computational cost. This is why we use it as an intermediary solution between the simple greedy heuristic and the exact solution.

Mehrotra et al. [155] made a further simplification to their heuristic: to fix $y_i = 0$ for any i such that $s_{iu} > \delta$, for parameter δ (in the original paper, δ was fixed to 3). In early experiments, however, we found that doing this did not bring any significant performance benefits, and thus to avoid introducing a parameter that probably does not generalize to multiple instance sizes, we do not use this simplification.

5.2.2.5 *Exact pricing*

Finally, we solve Model (5.10) exactly by branch-and-bound. Like the previous section, because Model (5.10) decomposes into n independent subproblems, we solve each one separately and add to the RLMP all solutions of negative cost. In each problem, we prune intermediate nodes with non-negative lower bounds, since they will never result in a negative reduced cost column.

In the pricing subproblem we model connectivity with a single-commodity flow approach, which we have reviewed in Section 2.4.1.4. Given reference unit u, it uses constraints

$$\sum_{j \in N(i)} (f_{ij} - f_{ji}) = y_i \qquad \forall i \in V \setminus \{u\}, \tag{5.12a}$$

$$\sum_{j \in N(i)} f_{ij} \leqslant (n-1) y_i \qquad \forall i \in V \setminus \{u\}, \tag{5.12b}$$

$$\sum_{j \in N(u)} f_{uj} = 0, \tag{5.12c}$$

$$f_{ij} \geqslant 0 \qquad \forall (i,j) \in A, \tag{5.12d}$$

where A is the arc-directed version of E.

We have also experimented to treat connectivity with the subtour elimination approach of constraints (2.24) and the cutset approach of (2.25) (see Section 2.4.1.4 for an overview). In practice, however, we found that in many subproblems branch-and-bound required a huge number of lazy cuts and sometimes did not terminate within the time limit. We think this is because formulations based on lazy cuts are well-suited when the objective function is correlated to connectivity, such as p-median, since the expected number of cuts is small, but not otherwise. Because Model (5.10)'s objective subtracts dual values, it can have negative coefficients and therefore its optimal solution is less likely to be naturally connected.

5.2.3 Lagrangean dual bounding

Consider the following model obtained by relaxing partition constraints (2.10c) of the Hess model (2.10) in a Lagrangean way:

$$\textbf{minimize} \quad \sum_{i,j \in V} x_{ij} d_{ij} + \sum_{i \in V} \lambda_i \Big(1 - \sum_{j \in V} x_{ij}\Big) \tag{5.13a}$$

$$\textbf{subject to} \quad \sum_{j \in V} x_{jj} = p, \tag{5.13b}$$

$$\text{(2.10d) to (2.10f)},$$

where $\lambda \in \mathbb{R}^n$ is the Lagrangean multiplier vector for the dualized constraints. Combining like terms, we can rewrite objective (5.13a) as

$$\sum_{i,j \in V} \Big(x_{ij}(d_{ij} - \lambda_i)\Big) + \sum_{i \in V} \lambda_i. \tag{5.14}$$

Observe that, for fixed j, constraints (2.10d) to (2.10f) and the first summation of (5.14) amount to an independent subproblem which is equivalent to solving the pricing subproblem (5.10) for center $u = j$, with $\pi_i^1 = \lambda_i$ and $\pi^0 = 0$. Let τ_j be the optimal solution to this subproblem. Then, Model (5.13) equates to

$$\begin{aligned}
\textbf{minimize} \quad & \sum_{j \in V} x_{jj}\tau_j + \sum_{i \in V} \pi_i^1 & & \text{(5.15a)} \\
\textbf{subject to} \quad & \sum_{j \in V} x_{jj} = p, & & \text{(5.15b)} \\
& x_{jj} \in \{0, 1\} \qquad \forall j \in V, & & \text{(5.15c)}
\end{aligned}$$

and can be solved in $O(n \log p)$ time by setting $x_{jj} = 1$ for the p units j with minimal τ_j. This means that when we solve the pricing problem optimally and obtain τ_j for $j \in V$, we can compute Model (5.15) to obtain a valid lower bound $\mathcal{L}$ for the LMP.

We use lower bound $\mathcal{L}$ to halt column generation early, in two ways. First, given the current incumbent value u^* we stop if $\mathcal{L} \geqslant u^*$, since this means the branch-and-bound node will be fathomed. Second, since the current solution υ to the RLMP is an upper bound for the LMP, we define a maximum gap ϵ (in our implementation, we use $\epsilon = 10^{-4}$) and stop iterating if $\mathcal{L} + \epsilon \geqslant \upsilon$, then return $\mathcal{L}$ as a dual bound. This helps alleviate the tailing-off effect which is a common issue in column generation [150], but as a downside it requires us to branch even if the current solution to the RLMP is integral, since we cannot prove its optimality without solving column generation until the end.

5.2.3.1 Variable fixing

We also use Model (5.15) to prove whether an optimal solution must (or must not) contain columns with certain centers. This is done as follows. Let $i_1, \dots, i_n$ be such that $\tau_{i_j} \leqslant \tau_{i_{j+1}}$ for $j \in [n-1]$. Then, if

$$\tau_{i_k} + \sum_{j \in [p-1]} \tau_{i_j} + \sum_{j \in V} \pi_j^1 > u^* \tag{5.16}$$

for $k \in [n] \setminus [p]$ and primal bound u^*, we can fix $x_{kk} = 0$ in the Hess model. In the RLMP, this amounts to setting

$$z_j = 0 \quad \forall j \in J(k), \tag{5.17}$$

where $J(k) = \{j \in J \mid c^{pm}(j) = k\}$. This also means we can safely skip solving the pricing subproblem rooted at $u = k$, since any feasible column it yields can also be obtained from other subproblems.

Similarly, if

$$-\tau_{i_k} + \sum_{j \in [p+1]} \tau_{i_j} + \sum_{j \in V} \pi_j^1 > u^* \tag{5.18}$$

for $k \in [p]$, we can fix $x_{kk} = 1$ in the Hess model. In the RLMP, this amounts to adding constraint

$$\sum_{j \in J(k)} z_j = 1. \tag{5.19}$$

5.2.4 Column management and optimal medians

All three pricing solutions (the greedy heuristic, the matheuristic and exact pricing) consider n subproblems with a fixed median, or root. Note, however, that given root $j \in V$ and column $i \in \mathcal{F}$ generated for j by one of these algorithms, it can be so that $j \neq c^{pm}(i)$, i.e. j is not the optimal median for i. Therefore, before adding column i to the RLMP we compute its optimal median in $O(|i|^2)$ time.

A problem arises when we fix out medians, which we do during Lagrangean bounding and also during instance pre-processing, as we shall see in Section 5.3. If the optimal median $k = c^{pm}(i)$ is such that x_{kk} is fixed to 0, we compute i's cost with the best available median. Namely, the objective coefficient of column i is computed as

$$c_i = \min\Big\{ \sum_{u \in i} d_{uj} \mid j \in i, x_{jj} \text{ is not fixed to } 0 \Big\}. \tag{5.20}$$

When variable x_{kk} is fixed to 0, we also update the cost of all existing columns which use k as median.

The main drawback of this strategy is that it allows columns in the RLMP that will never be selected in an optimal integer solution, even if they are penalized by higher costs. Another issue is that, in practice, the same column often ends up being generated by multiple pricing subproblems of fixed medians, which is unnecessarily repeated work.

One way to address this is to never generate columns with suboptimal centers in the first place. We have experimented with this by forcing the pricing subproblem with root j to only generate columns where j is the optimal median. In Model (5.10), we did this by adding constraints

$$\sum_{i \in V} d_{ij} y_i \leqslant M(1 - y_k) + \sum_{i \in V} d_{ik} y_i \qquad \forall k \in V \setminus \{j\}, \tag{5.21}$$

where M is a large constant. They force that, if $k \in V \setminus \{j\}$ is selected, it cannot be a better median than j. Unfortunately, we found that these constraints made the models too hard to solve, and column generation was intractable. We also experimented with adding these constraints lazily, with little effect. Still, we think the convergence of column generation could be improved if there would be an efficient way to enforce optimal medians.

5.2.5 Early branching

Despite our use of heuristics to speed up column generation and of Lagrangean bounds to combat tailing-off, we find that column generation still takes too long to terminate. We have identified two reasons for this. First, due to degeneracy column generation often gets stuck at plateaus of equal solution value over several iterations, which causes slow convergence. We discuss this issue in more detail in Section 5.3.4. Second, the exact pricing solver sometimes encounters difficult subproblems whose solution requires a substantial number of branch-and-bound nodes, and this tends to dominate the running time.

In an attempt to mitigate these issues, under certain circumstances we stop column generation early and branch. The first is if the RLMP's solution value v is lower than the incumbent value u^*, i.e. when we know the branch-and-bound node will no longer be fathomed. The second is if any call to exact pricing takes longer than a time limit, which we have fixed to 10 seconds. This addresses the first issue above.

The downside of branching early is that we forgo information on the optimal solution to the LMP which could be used to make better branching decisions, or even not branch at all if that solution is integral, thereby potentially reducing the number of nodes expanded. In practice, however, we find that early branching considerably reduces the time spent per branch-and-bound node.

5.3 Experimental work

In this section, we report on early computational experiments. In Section 5.3.1 we describe the instance sets used. In Section 5.3.2 we compare the results of our branch-and-price with the current best methods in the literature, which solve Hess et al. [108]'s model by treating connectivity with flow constraints [222], and subtour elimination constraints [197]. Next, in Section 5.3.3 we compare lower bounds obtained at the root node of branch-and-bound for the different formulations, and discuss how lower bounds could be improved in our case. Then, in Section 5.3.4 we present statistics on column generation and analyze its main issues and bottlenecks, then speculate on how they might be addressed.

We ran all experiments on a PC with a 12-core AMD Ryzen 9 3900X processor and 32 GB of main memory, running Ubuntu Linux 20.04. For each test we set a time limit of 30 minutes, and use only one thread. Our algorithms were coded in C++ and compiled with GCC 9.3.0 with maximum optimization. For solving the MIP models we use CPLEX 20.1.0. Since our algorithm depends little on randomization (namely, to generate initial primal solutions and in greedy multistart pricing, where the randomization is amortized since it is called many times), we ran each test with a fixed random seed.

In all tests, as a preprocessing step we apply the Lagrangean-based variable fixing approach of Validi et al. [222], which we described in Section 2.4.1.5. As we will see in the next section, it fixes a significant percentage (over 80%) of Hess model variables in all instances. For a fair comparison, we apply variable fixing to all algorithm variants we consider in our experiments. To obtain the upper bound needed for variable fixing we run the location-allocation heuristic of Validi et al. [222], which we also use as an initial primal solution, as we have mentioned in Section 5.2.2.1. Since variable fixing depends on solving the LP relaxation of the Hess model, which is very slow for $n \geqslant 500$, we do not include the runtime for solving the LP in our final results. Because we use variable fixing in all compared variants, this does not affect our interpretation of the results.

5.3.1 Test instances

For our tests we use the smaller instance sets SRC and RF, described in Section 3.4. Since sets SRC and RF were generated for problems of $|A| = 3$ balancing attributes, we transform them to single-attribute instances by taking $w_i = (w_i^1 + w_i^2 + w_i^3)/3$. Additionally, we include two instance sets from a political districting domain in the United States, introduced by Validi et al. [222]: *counties* and *tracts*. These real-world instances range in size from $n = 44$ to $n = 1115$, but generally require fewer districts, with $p = 1$ to $p = 53$. In our experiments, we have removed several trivial instances of $p = 1$. For tract-level instances we use a reduced set *tracts-R* which contains only

Table 5.1: Test instance data.

Inst.	Source	#	n	m	p	UB_h	$LB_{\mathcal{L}}$	Fix. (%)	Cens.
SRC	[197]	20	60	101	4	0.08	-2.61	91.4	15
counties	[222]	9	77	188	4	0.11	-5.64	82.3	21
SRC	[197]	20	80	140	5	0.51	-3.82	81.4	44
SRC	[197]	20	100	177	6	0.66	-2.35	84.9	50
SRC	[197]	20	120	215	7	0.53	-2.46	83.7	74
SRC	[197]	10	150	271	8	0.95	-2.35	84.9	88
SRC	[197]	10	200	368	11	1.40	-2.44	81.4	163
RF/DS	[190]	20	500	955	10	1.08	-0.51	88.2	258
RF/DT	[190]	20	500	950	10	1.15	-0.53	86.8	289
tracts-R	[222]	14	667	1,795	4	0.01	-0.13	97.4	50

instances which Validi et al. [222] have solved optimally within 1 hour, since it is unlikely that our method will be able to solve any of the other instances.

Table 5.1 summarizes data for the instances used. We show, for each instance set (Inst.) the number of instances (#), the average instance size (n, m and p), the average initial primal (UB_h) and Lagrangean ($LB_{\mathcal{L}}$, computed by Validi et al. [222]'s variable fixing method) bounds relative to best known lower bound per instance, the percentage (Fix. (%)) of Hess model variables fixed to zero during pre-processing, and the number of center (Cens.) variables not fixed to zero by pre-processing. We see that preprocessing significantly reduces effective instances sizes and fixes most centers, which are usually the complicating variables in the Hess model. This effect is more apparent for instance sets tracts-R and counties. We think this is because their unit topologies are less uniform.

5.3.2 Comparison to other exact approaches

In this section we compare our proposed branch-and-price to the formulations of Validi et al. [222], which we call "Flow", and of Salazar-Aguilar et al. [197], which we call "Subtour". We solve them using CPLEX. Both formulations include inequalities (2.12) for strength, and differ in how connectivity is handled: Flow uses multi-commodity flow constraints (2.27), while Subtour uses subtour elimination constraints (2.24), which are added as lazy constraints during branch-and-bound. We have also considered the cutset-based formulation of Validi et al. [222], but as reported by the authors it performs nearly identically to Flow. This was also corroborated by our early experimentation, hence we only include Flow in the comparison.

Table 5.2 shows the results. Results for our branch-and-price are shown under

Table 5.2: Results for the proposed branch-and-price and the current best solutions in the literature.

Inst.	n	Flow				Subtour				B&P			
		t	LB	UB	Opt.	t	LB	UB	Opt.	t	LB	UB	Opt.
SRC	60	0.5	0.00	0.00	20	0.4	0.00	0.00	20	0.8	0.00	0.00	20
counties	77	4.1	0.00	0.00	9	4.0	0.00	0.00	9	190.7	0.00	0.00	9
SRC	80	3.6	0.00	0.00	20	2.0	0.00	0.00	20	165.5	0.00	0.00	20
SRC	100	21.7	0.00	0.00	20	18.4	0.00	0.00	20	348.0	-0.45	0.00	18
SRC	120	9.5	0.00	0.00	20	5.9	0.00	0.00	20	338.7	-0.24	0.00	18
SRC	150	33.6	0.00	0.00	10	54.2	0.00	0.00	10	720.0	-1.19	0.41	7
SRC	200	440.9	-0.26	0.06	8	302.2	0.00	0.00	9	1,457.3	-1.17	1.18	2
RF/DS	500	978.6	-0.34	0.58	11	642.8	0.00	0.10	17	1,800.0	-0.51	1.17	0
RF/DT	500	1,248.0	-0.32	0.39	8	646.4	0.00	0.21	17	1,422.0	-0.37	1.06	0
tracts-R	667	671.8	-0.06	0.00	10	437.7	0.00	0.00	11	1,686.9	-0.13	0.01	1
Avg./Tot.		341.2	-0.10	0.10	136	211.4	0.00	0.03	153	813.0	-0.41	0.38	95

“B&P”. For each method and instance set we show the average running time (t) in seconds, the average final lower (LB) and upper (UB) bounds relative to the best known lower bound per instance, and the number of instances solved to optimality (Opt.).

At a first glance we see that the subtour formulation of Salazar-Aguilar et al. [197] is a clear winner, finding the best known lower bound in every single run and in less time, ultimately solving the most instances (153). Flow ranks second at 136 instances, as it falls short at larger instances of $n \geqslant 500$. Our branch-and-price is able to handle smaller instances of $n \leqslant 100$ with high running times, but struggles on larger ones, and ultimately solves only 95 instances. As we shall see in the next two experiments, this is caused by a combination of three factors: i) the slow convergence of column generation, ii) the difficulty in solving the pricing subproblem, and iii) weaker lower bounds caused by our lack of automatic cutting plane generation.

5.3.3 Comparison of dual bounds

In this experiment we assess empirically the formulation strength of branch-and-price's set partitioning reformulation, compared to Flow and Subtour. Table 5.3 shows the relevant statistics, using data from the experiment of Section 5.3.2. We report, for each instance set, the average initial heuristic upper bound (UB_h), Lagrangean lower bound ($LB_{\mathcal{L}}$), and Hess LP bound (LB_{LP}), without connectivity constraints. Then, for each formulation we report the lower bound obtained after solving the root node

Table 5.3: Empirical comparison of lower bound strengths.

Inst.	n	UB_h	$LB_{\mathcal{L}}$	LB_{LP}	Flow		Subtour		B&P	
					Nd.	LB_{root}	Nd.	LB_{root}	Nd.	LB_{root}
DU60	60	0.08	-2.61	-2.61	0.6	-0.07	2.5	-0.05	2.7	-0.19
counties	77	0.11	-5.64	-5.64	72.4	-2.26	305.0	-2.36	37.8	-2.27
DU80	80	0.51	-3.82	-3.82	41.1	-1.48	43.7	-1.47	86.7	-1.54
DU100	100	0.66	-2.35	-2.35	21.7	-1.13	13.9	-1.11	190.5	-1.01
DU120	120	0.53	-2.46	-2.46	61.3	-0.90	94.7	-0.92	125.2	-0.75
DU150	150	0.95	-2.35	-2.35	14.1	-1.45	21.9	-1.48	135.3	-1.37
DU200	200	1.40	-2.44	-2.44	44.0	-1.41	58.5	-1.44	45.0	-1.25
DS	500	1.08	-0.51	-0.51	—	-0.46	—	-0.46	—	-0.51
DT	500	1.15	-0.53	-0.53	—	-0.46	—	-0.45	—	-0.37
tracts-R	667	0.01	-0.13	-0.13	0.0	-0.12	0.0	-0.09	1.0	-0.13
Avg./Tot.		0.65	-2.29	-2.29	31.9	-0.97	67.5	-0.98	78.0	-0.94

of branch-and-bound (LB_{root}) and the average number of branch-and-bound nodes (Nd.), for instances which were solved by all variants. All upper and lower bounds are relative to the best known lower bound.

First, we see that Lagrangean bounds and Hess LP bounds are identical in all instances. This was unexpected, considering that reduced cost analysis in the Lagrangean yields substantially more variable fixes than in the Hess LP, in the method of Validi et al. [222]. We believe a further investigation of this phenomenon is warranted. It might be interesting, for instance, to compare optimal bounds between all possible combinations of dualized constraints among (2.10b) to (2.10e).

We also see that branch-and-price's LB_{root} is better than LB_{LP}. This is expected, since as we discussed in Section 2.4.1.3 the set partitioning formulation is tighter than Hess's. Branch-and-price also generally has stronger root bounds than Flow and Subtour when solved with CPLEX, but not always; see e.g. instance sets DU80 and DU100. This is likely because CPLEX uses generic cutting planes to strengthen dual bounds. Looking at the number of branch-and-bound nodes expanded, we see this effect accentuated in practice as both Subtour and Flow actually expand fewer nodes than branch-and-price, despite having weaker formulations. This shows that a tighter formulation alone does not compensate a lack of strong cuts, and marks a clear path for future improvement of our method. Another possibility is that this difference could be due to branching and node expansion strategies.

Comparing Flow and Subtour, we see that Flow expands fewer than half the nodes of Subtour, but spends significantly more effort in each node, as indicated by running times in Table 5.2. Because Subtour has no connectivity cuts at the root, its root

Table 5.4: Column generation statistics.

Inst.	n	Iter.	Cols.	t_{LP}	t_{mh}	t_{ex}	t_{prf}	col_{pl}	col_{gr}	col_{mh}	col_{ex}
DU60	60	18	7.4	1.2	20.2	19.7	8.0	3.5	78.7	13.8	4.0
counties	77	242	3.3	4.3	39.4	45.1	1.8	14.4	51.0	29.3	5.3
DU80	80	31	8.1	3.2	35.4	42.0	3.2	21.3	55.0	17.3	6.4
DU100	100	64	7.1	5.6	34.9	38.9	2.5	19.1	55.5	17.7	7.8
DU120	120	44	10.5	6.9	37.4	38.1	2.7	24.7	52.4	17.0	5.9
DU150	150	51	7.9	8.5	35.7	33.7	1.7	26.8	50.3	17.0	5.9
DU200	200	21	5.5	11.0	31.5	37.1	1.3	40.9	40.6	13.2	5.4
DS	500	330	94.4	39.4	1.0	52.8	0.0	0.0	93.9	6.0	0.0
DT	500	160	68.3	72.1	6.3	7.9	0.2	1.1	81.7	13.6	3.5
tracts-R	667	714	154.4	79.8	13.4	5.6	0.0	0.0	89.8	9.8	0.4

bound is (expectedly) worse than Flow's, on average. Strangely, this was not the case for instance sets DU80 and DU100. For some reason, CPLEX's presolve was able to strengthen the model better when flow constraints were not present.

5.3.4 Column generation analysis

In this section, we analyze how the different components of column generation contribute to convergence and running time. In early experiments, we realized that different parts of column generation were the bottleneck depending on the instance. Our goal in this experiment is to highlight the major issues that make branch-and-price not competitive, and to reflect on how they might be addressed.

In Table 5.4 we display some statistics using data from the experiment of Section 5.3.2. For each instance set we show averages of the number of column generation iterations per branch-and-bound node (Iter.), the number of new columns added per iteration (Cols.), and the running times (relative to the total running time of branch-and-price, in percent) for solving the RLMP with Simplex (t_{LP}), the pricing matheuristic of Section 5.2.2.4 (t_{mh}), exact pricing (t_{ex}), and of the last column generation iteration (t_{prf}), where exact pricing is required to obtain an optimality proof for the RMP. We also show the relative number of columns (in percent) which were added by each of the four methods outlined in Section 5.2.2: searching the global column pool (col_{pl}), greedy heuristic (col_{gr}), matheuristic (col_{mh}) and exact pricing (col_{ex}).

For smaller instances ($n \leqslant 200$), we see that the time to solve the RLMP accounts for less than 10% of running time, as fewer columns are needed to reach optimality. In larger instances ($n \geqslant 500$), however, the number of columns and consequently the LP time increase drastically. Since the RLMP is highly degenerate, in these large instances we find that new columns rarely enter the base, even if they have negative reduced costs. This leads to plateau periods during column generation where the solution value υ of the RLMP remains unchanged despite a very large number of columns being added. In nearly all instances of data sets DS, DT and tracts-R, υ never improved from the initial primal solution, and negative reduced cost columns continued to be added for several iterations until the LP became over encumbered and the time limit was reached. This phenomenon was also recognized by Gurnee and Shmoys [100], who suggest that it is caused by the fact that the columns generated rarely "fit" together to form p-partitions of V, especially in large instances.

In instances that branch-and-price consistently solves ($n \leqslant 120$), most of the computational effort is split between the matheuristic and exact pricing, each taking about 30 to 40% of the total running time. This is despite these methods producing only about a quarter of all columns. However, looking at column t_{prf} we see that the cost to prove optimality is rather small in comparison to t_{ex}, usually less than 5% of the total running time. This indicates that most exact pricing runs could be avoided by having a better heuristic. Moreover, we see the global column pool and greedy heuristic work as intended, as they are extremely fast and combined generate over 70% of all columns, on average.

5.4 Conclusion and planned work

In this chapter we presented our ongoing work on an exact branch-and-price algorithm for the p-Median Districting Problem. We have extended the heuristic of Mehrotra et al. [155], which uses branch-and-price on a set partitioning formulation, with several techniques from the literature. These techniques include i) a layered approach that uses heuristics and a column pool to find columns quickly, ii) an exact solution to the pricing problem which uses flow constraints to model connectivity, iii) the use of early branching during column generation, including Lagrangean dual bounding, and iv) the use of variable fixing techniques from the literature as pre-processing step.

In computational experiments we find that our method generally finds better lower bounds as it uses a tighter formulation, but solves fewer instances than other approaches. This is because column generation faces a number of issues, which we discuss in the following.

First, due to high degeneracy of the RLMP, in large instances column generation is slow to converge as it requires a huge number of columns to make progress on the objective value. Because the RLMP is degenerate, its dual has fewer constraints than variables and thus admits many optimal solutions. As a consequence, column generation alternates between these (often distant) dual solutions of equal value without making progress. A known approach to mitigate this effect, which we intend to investigate, is dual stabilization [56]. It limits the variation between consecutive dual solutions through several techniques, lessening the back-and-forth effect.

Another possible solution to the degeneracy issue is to generate sets of columns that form p-partitions of V, instead of single columns that do not “fit together”. This would make columns more likely to enter the base and not remain unused. This could be done, e.g. by changing the pricing subproblem to generate feasible districting plans such that one or more districts are negative reduced cost columns. This would make pricing significantly more costly to solve, however. Another way is to use the binary partitioning approach of Gurnee and Shmoys [100], which recursively partitions the problem into two smaller subproblems and generates several solutions to each.

Second, as we discussed in Section 5.3.3 our branch-and-price implementation requires more node expansions than CPLEX does for the flow and subtour formulations. This difference is likely due to a combination of i) better lower bounds obtained through the automatic generation of valid inequalities, ii) node selection techniques, iii) branching rules. We will investigate how our method can be improved in these aspects. In particular, (i) appears to be the most promising, and we will consider valid inequalities which have been used for similar capacitated location problems such as lifted cover inequalities [98] and Fenchel cuts [24].

Third, as we saw in Table 5.4, exact pricing consumes almost half the total running time of branch-and-price, but most calls to it could have been avoided if heuristic pricing had found a negative reduced cost column. Therefore, a better heuristic could help to improve running times by up to 30-40%, and possibly more on larger instances. While this will certainly help, it is however clear that this speed up alone will not be sufficient to make branch-and-price competitive.

Last, we will experiment with enforcing connectivity by subtour-based lazy constraints, as in Salazar-Aguilar et al. [197], on the master problem. Currently, exact pricing uses flow constraints to ensure every single column is connected; unfortunately, because of degeneracy, this means significant effort is invested into generating connected columns that end up not being used. To enforce connectivity lazily in the master, rather than fully during pricing, would make exact pricing considerably easier to solve. As a downside, this enlarges significantly the space of feasible columns, which could slow down the convergence of column generation. Another issue might be that we will not be able to stop branching once an integer solution is found, since it could have disconnected columns. In this case, we could use the heuristic of Validi et al. [222] to repair disconnectivity, which was effective in their method.

REFERENCES

[1] R. K. Ahuja, Ö. Ergun, J. B. Orlin, and A. P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1-3): 75–102, 2002.

[2] M. Albareda-Sambola and J. Rodríguez-Pereira. Location-routing and location-arc routing. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, pages 431–451. Springer International Publishing, Cham, Switzerland, 2019.

[3] K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.

[4] A. M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 1979.

[5] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Concorde TSP solver, 2006. URL http://www.tsp.gatech.edu/concorde.

[6] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, United States, 2006.

[7] T. Arani and V. Lotfi. A three phased approach to final exam scheduling. *IIE Transactions*, 21(1):86–96, 1989.

[8] V. Arredondo, M. Martínez-Panero, T. Peña, and F. Ricca. Mathematical political districting taking care of minority groups. *Annals of Operations Research*, 305(1):375–402, 2021.

[9] E. A. Autry, D. Carter, G. Herschlag, Z. Hunter, and J. C. Mattingly. Multi-scale merge-split Markov chain Monte Carlo for redistricting. *Preprint arXiv:2008.08054*, 2020.

[10] K. R. Baker and S. G. Powell. Methods for assigning students to groups: a study of alternative objective functions. *Journal of the Operational Research Society*, 53(4):397–404, 2002.

[11] R. Barnes and J. Solomon. Gerrymandering and compactness: Implementation, flexibility and abuse. *Preprint arXiv:1803.02857*, 2018.

[12] F. Bação, V. Lobo, and M. Painho. Applying genetic algorithms to zone design. *Soft Computing*, 9(5):341–348, 2005.

[13] J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4):299–327, 1959.

[14] R. Becker, I. Lari, M. Lucertini, and B. Simeone. Max-min partitioning of grid graphs into connected components. *Networks*, 32(2):115–125, 1998.

[15] G. Benade, N. Ho-Nguyen, and J. Hooker. Political districting without geography. *Operations Research Perspectives*, 9:100227, 2022.

[16] M. Bender, A. Meyer, J. Kalcsics, and S. Nickel. The multi-period service territory design problem – An introduction, a model and a heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 96:135–157, 2016.

[17] M. Bender, J. Kalcsics, S. Nickel, and M. Pouls. A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. *European Journal of Operational Research*, 269(1):382–396, 2018.

[18] E. Benzarti, E. Sahin, and Y. Dallery. Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems*, 55 (2):587–598, 2013.

[19] P. K. Bergey, C. T. Ragsdale, and M. Hoskote. A simulated annealing genetic algorithm for the electrical power districting problem. *Annals of Operations Research*, 121(1):33–55, 2003.

[20] P. Bertolazzi, L. Bianco, and S. Ricciardelli. A method for determining the optimal districting in urban emergency services. *Computers & Operations Research*, 4(1):1–12, 1977.

[21] M. Blais, S. D. Lapierre, and G. Laporte. Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11): 1141–1147, 2003.

[22] L. Bodin and L. Levy. The arc partitioning problem. *European Journal of Operational Research*, 53(3):393–401, 1991.

[23] S. Borgwardt, A. Brieden, and P. Gritzmann. Geometric clustering for the consolidation of farmland and woodland. *The Mathematical Intelligencer*, 36 (2):37–44, 2014.

[24] E. A. Boyd. Fenchel cutting planes for integer programs. *Operations Research*, 42(1):53–64, 1994.

[25] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26, 2003.

[26] B. Bozkaya, E. Erkut, D. Haight, and G. Laporte. Designing new electoral districts for the city of Edmonton. *Interfaces*, 41(6):534–547, 2011.

[27] A. Brieden, P. Gritzmann, and F. Klemm. Constrained clustering via diagrams: A unified theory and its application to electoral district design. *European Journal of Operational Research*, 263(1):18–34, 2017.

[28] J. Brimberg, N. Mladenović, and D. Urošević. Solving the maximally diverse grouping problem by skewed general variable neighborhood search. *Information Sciences*, 295:650–675, 2015.

[29] Bureau of the Census. Census tracts for the 2020 census. https://www.census.gov/data.html, 2018. Retrieved in 12-12-2022.

[30] A. Butsch. *Districting Problems - New Geometrically Motivated Approaches*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2016.

[31] A. Butsch, J. Kalcsics, and G. Laporte. Districting for arc routing. *INFORMS Journal on Computing*, 26(4):809–824, 2014.

[32] S. I. Caballero-Hernández, R. Z. Ríos-Mercado, F. López, and S. E. Schaeffer. Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. In J. E. Fernandez, S. Noriega, A. Mital, S. E. Butt, and T. K. Fredericks, editors, *Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE)*, pages 422–427, Cancun, Mexico, November 2007. ISBN: 978-0-9654506-3-8.

[33] M. Camacho-Collados, F. Liberatore, and J. M. Angulo. A multi-criteria police districting problem for the efficient and effective design of patrol sector. *European Journal of Operational Research*, 246(2):674–684, 2015.

[34] D. Carter, G. Herschlag, Z. Hunter, and J. Mattingly. A merge-split proposal for reversible Monte Carlo Markov chain sampling of redistricting plans. *Preprint arXiv:1911.01503*, 2019.

[35] R. Carvajal, M. Constantino, M. Goycoolea, J. P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836, 2013.

[36] C. Chou, S. Kimbrough, J. Sullivan-Fedock, C. J. Woodard, and F. H. Murphy. Using interactive evolutionary computation (IEC) with validated surrogate fitness functions for redistricting. In T. Soule, editor, *Proceedings of the 14th Annual Conference on Genetic and Evolutionary computation (GECCO'12)*, pages 1071–1078, Philadelphia, United States, July 2012.

[37] V. Cohen-Addad, P. N. Klein, and N. E. Young. Balanced centroidal power diagrams for redistricting. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'18)*, pages 389–396, Seattle, United States, November 2018.

[38] L. Cooper. Heuristic methods for location-allocation problems. *SIAM Review*, 6 (1):37–53, 1964.

[39] Á. Corberán and G. Laporte. *Arc Routing: Problems, Methods, and Applications*. SIAM, Philadelphia, United States, 2015.

[40] O. Coudert. Exact coloring of real-life graphs is easy. In *Proceedings of the 34th Annual Design Automation Conference*, pages 121–126, Anaheim, United States, June 1997. ACM Press.

[41] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

[42] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6): 667–684, 2002.

[43] D. Datta, J. Malczewski, and J. R. Figueira. Spatial aggregation and compactness of census areas with a multiobjective genetic algorithm: A case study in Canada. *Environment and Planning B: Planning and Design*, 39(2):376–392, 2012.

[44] D. Datta, J. Figueira, A. Gourtani, and A. Morton. Optimal administrative geographies: An algorithmic approach. *Socio-Economic Planning Sciences*, 47 (3):247–257, 2013.

[45] L. S. De Assis, P. M. Franca, and F. L. Usberti. A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, 41:65–75, 2014.

[46] P. D. L. P. De Fréminville, G. Desaulniers, L.-M. Rousseau, and S. Perron. A column generation heuristic for districting the price of a financial product. *Journal of the Operational Research Society*, 66(6):965–978, 2015.

[47] A. M. P. de Mendonça. *Distritamento Aplicado ao Problema de Faturamento em Redes de Serviço*. PhD thesis, Centro Federal de Educaçao Tecnológica Celso Suckow da Fonseca, Rio de Janeiro, Brazil, 2020.

[48] R. F. Deckro. Multiple objective districting: A general heuristic approach using multiple criteria. *Journal of the Operational Research Society*, 28(4):953–961, 1977.

[49] D. DeFord, M. Duchin, and J. Solomon. Recombination: A family of Markov chains for redistricting. *Harvard Data Science Review*, 3:1–58, 2021.

[50] D. Demetriou, L. See, and J. Stillwell. A spatial genetic algorithm for automating land partitioning. *International Journal of Geographical Information Science*, 27(12):2391–2409, 2013.

[51] D. Demetriou, L. See, and J. Stillwell. A parcel shape index for use in land consolidation planning. *Transactions in GIS*, 17(6):861–882, 2013.

[52] G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.

[53] G. A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, s3-2(1):69–81, 1952.

[54] R. G. Dixon. Democratic representation: Reapportionment in law and politics. *American Political Science Review*, 63(2):567–568, 1969.

[55] A. Drexl and K. Haase. Fast approximation methods for sales force deployment. *Management Science*, 45(10):1307–1323, 1999.

[56] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.

[57] A. Duarte, J. Sánchez-Oro, M. G. C. Resende, F. Glover, and R. Martí. Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization. *Information Science*, 296:46–60, 2015.

[58] M. Duchin and B. E. Tenner. Discrete geometry for electoral geography. *Preprint arXiv:1808.05860*, 2018.

[59] J. C. Duque, L. Anselin, and S. J. Rey. The Max-p-regions Problem. *Journal of Regional Science*, 52(3):397–419, 2012.

[60] M. G. Elizondo-Amaya, R. Z. Ríos-Mercado, and J. A. Díaz. A dual bounding scheme for a territory design problem. *Computers & Operations Research*, 44: 193–205, 2014.

[61] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[62] E. Erkut, Y. Ülküsal, and O. Yeniçerioğlu. A comparison of p-dispersion heuristics. *Computers & Operations research*, 21(10):1103–1113, 1994.

[63] R. Z. Farahani and M. Hekmatfar, editors. *Facility Location: Concepts, Models, Algorithms and Case Studies*. Springer-Verlag, Berlin, Germany, 2009.

[64] H. Farughi, M. Tavana, S. Mostafayi, and F. J. Santos Arteaga. A novel optimization model for designing compact, balanced, and contiguous healthcare districts. *Journal of the Operational Research Society*, 71(11):1740–1759, 2020.

[65] J. A. Ferland and G. Guénette. Decision support system for the school districting problem. *Operations Research*, 38(1):15–21, 1990.

[66] E. Fernández, J. Kalcsics, S. Nickel, and R. Z. Ríos-Mercado. A novel maximum dispersion territory design model arising in the implementation of the weee-directive. *Journal of the Operational Research Society*, 61(3):503–514, 2010.

[67] E. Fernández, J. Kalcsics, and S. Nickel. The maximum dispersion problem. *Omega*, 41(4):721–730, 2013.

[68] J. Ferreira Neto. Private communication, 2018.

[69] B. Fifield, M. Higgins, K. Imai, and A. Tarr. Automated redistricting simulation using Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 29(4):715–728, 2020.

[70] M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out steiner trees: a node-based model for uniform edge costs. *Mathematical Programming Computation*, 9(2):203–229, 2017.

[71] B. Fleischmann and J. N. Paraschis. Solving a large scale districting problem: a case report. *Computers & Operations Research*, 15(6):521–533, 1988.

[72] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[73] S. L. Forman and Y. Yue. Congressional districting using a TSP-based genetic algorithm. In E. Cantú-Paz, J. A. Foster, K. Deb, L. David, and R. Rajkumar, editors, *Genetic and Evolutionary Computation – GECCO 2003*, volume 2723 of *Lecture Notes in Computer Science*, pages 2072–2083. Springer, Berlin, Germany, 2003.

[74] M. Fredman and M. Saks. The cell probe complexity of dynamic data structures. In *ACM Symposium on Theory of Computing*, pages 345–354, 1989.

[75] G. García-Ayala, J. L. González-Velarde, R. Z. Ríos-Mercado, and E. Fernández. A novel model for arc territory design: promoting Eulerian districts. *International Transactions in Operational Research*, 23(3):433–458, 2016.

[76] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, New York, United States, 1979.

[77] M. R. Garey, R. L. Graham, D. S. Johnson, and A. C.-C. Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory, Series A*, 21(3):257–298, 1976.

[78] R. S. Garfinkel and G. L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508, 1970.

[79] A. Gelman. *Red State, Blue State, Rich State, Poor State: Why Americans Vote the Way They Do*. Princeton University Press, Princeton, United States, 2009.

[80] M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*. Springer, New York, United States, 2nd edition, 2010.

[81] J. A. George, B. W. Lamar, and C. A. Wallace. Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 31 (1):11–28, 1997.

[82] A. F. Gil, M. G. Sánchez, C. Castro, and A. Pérez-Alonso. A mixed-integer linear programming model and a metaheuristic approach for the selection and allocation of land parcels problem. *International Transactions in Operational Research*, Forthcoming (DOI: 10.1111/itor.13115).

[83] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[84] A. Gliesch and M. Ritt. A generic approach to districting with diameter or center-based objectives. In M. López-Ibáñez, editor, *GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 249–257, Prague, Czech Republic, July 2019. ACM.

[85] A. Gliesch and M. Ritt. A hybrid heuristic for the maximum dispersion problem. *European Journal of Operational Research*, 288(3):721–735, 2021.

[86] A. Gliesch and M. Ritt. A new heuristic for finding verifiable k-vertex-critical subgraphs. *Journal of Heuristics*, 28:61–91, 2022.

[87] A. Gliesch, M. Ritt, and M. C. Moreira. A genetic algorithm for fair land allocation. In P. A. N. Bosman, editor, *GECCO '17: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 793–800, Berlin, Germany, July 2017. ACM.

[88] A. Gliesch, M. Ritt, and M. C. Moreira. A multistart alternating tabu search for commercial districting. In A. Liefooghe and M. López-Ibáñez, editors, *European Conference on Evolutionary Computation in Combinatorial Optimization*, volume 10782 of *Lecture Notes in Computer Science*, pages 158–173, Cham, Switzerland, 2018. Springer.

[89] A. Gliesch, M. Ritt, A. H. S. Cruz, and M. C. O. Moreira. A heuristic algorithm for districting problems with similarity constraints. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, Glasgow, United Kingdom, July 2020. IEEE.

[90] A. Gliesch, M. Ritt, A. H. S. Cruz, and M. C. O. Moreira. A hybrid heuristic for districting problems with routing criteria. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–9, Glasgow, United Kingdom, July 2020. IEEE.

[91] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision S ciences*, 8(1):156–166, 1977.

[92] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.

[93] F. Glover and J.-K. Hao. The case for strategic oscillation. *Annals of Operations Research*, 183(1):163–173, 2011.

[94] S. Goderbauer and J. Winandy. Political districting problem: Literature review and discussion with regard to federal elections in Germany. Technical report, RWTH Aachen, Aachen, Germany, 2018.

[95] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1:69–93, 1991.

[96] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[97] E. C. Griffith. *The Rise and Development of the Gerrymander*. Scott, Foresman, United States, 1907.

[98] Z. Gu, G. L. Nemhauser, and M. W. Savelsbergh. Lifted flow cover inequalities for mixed 0-1 integer programs. *Mathematical Programming*, 85(3):439–467, 1999.

[99] W. Gurnee. Scalable approximations of capacitated k-medians for political districting. Technical report, Cornell University, Ithaca, United States, 2020.

[100] W. Gurnee and D. B. Shmoys. Fairmandering: A column generation heuristic for fairness-optimized political districting. *Preprint arXiv:2103.11469*, 2021.

[101] G. Gutin and A. P. Punnen, editors. *The Traveling Salesman Problem and Its Variations*. Springer, Kluwer, Dordrecht, The Netherlands, 2002.

[102] S. Hanafi, A. Freville, and P. Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search

methods. *Information Systems and Operational Research*, 37(3):236–254, 1999.

[103] P. Hansen, N. Mladenović, J. Brimberg, and J. A. M. Pérez. Variable neighborhood search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, pages 57–97. Springer, Cham, Switzerland, 2019.

[104] S. Har-Peled. On the expected complexity of random convex hulls. *Preprint arXiv:1111.53404*, 2011.

[105] D. Haugland, S. C. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.

[106] A. Hertz and D. Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.

[107] S. W. Hess and S. A. Samuels. Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4-part-ii):41–54, 1971.

[108] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006, 1965.

[109] C. Hierholzer and C. Wiener. Über die möglichkeit, einen linienzug ohne wiederholung und ohne unterbrechung zu umfahren. *Mathematische Annalen*, 6(1):30–32, 1873.

[110] A. K. Hirose, C. T. Scarpin, and J. E. P. Junior. Goal programming approach for political districting in Santa Catarina state: Brazil. *Annals of Operations Research*, 287(1):209–232, 2020.

[111] M. Hojati. Optimal political districting. *Computers & Operations Research*, 23 (12):1147–1161, 1996.

[112] C. Hojny, I. Joormann, H. Lüthen, and M. Schmidt. Mixed-integer programming techniques for the connected max-k-cut problem. *Mathematical Programming Computation*, 13(1):75–132, 2021.

[113] J. H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2):88–105, 1973.

[114] D. L. Horn, C. R. Hampton, and A. J. Vandenberg. Practical application of district compactness. *Political Geography*, 12(2):103–120, 1993.

[115] D. L. Huerta-Muñoz, R. Z. Ríos-Mercado, and R. Ruiz. An iterated greedy heuristic for a market segmentation problem with multiple attributes. *European Journal of Operational Research*, 261(1):75–87, 2017.

[116] A. I. Jarrah and J. F. Bard. Large-scale pickup and delivery work area design. *Computers & Operations Research*, 39(12):3102–3118, 2012.

[117] T. R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley, New York, United States, 2005.

[118] D. S. Johnson and L. A. McGeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. Wiley, Chichester, United Kingdom, 1997.

[119] J. Kalcsics and R. Z. Ríos-Mercado. Districting problems. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, chapter 25, pages 703–741. Springer, Cham, Switzerland, 2nd edition, 2019.

[120] J. Kalcsics, T. Melo, S. Nickel, and H. Gündra. Planning sales territories – A facility location approach. In P. Chamoni, R. Leisten, A. Martin, J. Minnemann, and H. Stadtler, editors, *Operations Research Proceedings*, pages 141–148, Germany, 2002. Springer.

[121] J. Kalcsics, S. Nickel, and M. Schröder. Towards a unified territorial design approach – Applications, algorithms and GIS integration. *Top*, 13(1):1–56, 2005.

[122] J. Kalcsics, S. Nickel, and M. Schröder. A generic geometric approach to territory design and districting. Technical report, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2009.

[123] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. *SIAM Journal on Applied Mathematics*, 37(3):513–560, 1979.

[124] A. R. Kaufman, G. King, and M. Komisarchik. How to measure legislative district compactness if you only know it when you see it. *American Journal of Political Science*, 65(3):533–550, 2021.

[125] J. M. Keil. Polygon decomposition. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 11, pages 491–518. Elsevier, Amsterdam, The Netherlands, 2000.

[126] D. M. King, S. H. Jacobson, E. C. Sewell, and W. K. T. Cho. Geo-graphs: An efficient model for enforcing contiguity and hole constraints in planar graph partitioning. *Operations Research*, 60(5):1213–1228, 2012.

[127] D. M. King, S. H. Jacobson, and E. C. Sewell. Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning. *Mathematical Programming*, 149(1-2):425–457, 2014.

[128] D. M. King, S. H. Jacobson, and E. C. Sewell. The geo-graph in practice: creating united states congressional districts from census blocks. *Computational Optimization and Applications*, 69(1):25–49, 2018.

[129] S. M. Korman. *Graph colouring and related problems in operations research*. PhD thesis, Imperial College London, London, United Kingdom, 1975.

[130] J. Łącki and P. Sankowski. Optimal decremental connectivity in planar graphs. *Theory of Computing Systems*, 61(4):1037–1053, 2017.

[131] J. Łącki, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen. Single source–all sinks max flows in planar digraphs. In *IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 599–608, New Brunswick, Canada, 2012. IEEE.

[132] X. Lai and J. K. Hao. Iterated maxima search for the maximally diverse grouping problem. *European Journal of Operational Research*, 254(3):780–800, 2016.

[133] G. Laporte and I. H. Osman. Routing problems: A bibliography. *Annals of Operations Research*, 61(1):227–262, 1995.

[134] G. Laporte, S. Nickel, and F. Saldanha da Gama, editors. *Location Science*. Springer, Cham, Switzerland, 2nd edition, 2019.

[135] H. Lei, G. Laporte, and B. Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1):67–85, 2012.

[136] H. Lei, G. Laporte, Y. Liu, and T. Zhang. Dynamic design of sales territories. *Computers & Operations Research*, 56:84–92, 2015.

[137] H. Lei, R. Wang, and G. Laporte. Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Computers & Operations Research*, 67:12–24, 2016.

[138] H. A. Levin and S. A. Friedler. Automated congressional redistricting. *Journal of Experimental Algorithmics*, 24:1–24, 2019.

[139] R. Lewis, J. Thompson, C. Mumford, and J. Gillard. A wide-ranging computational comparison of high-performance graph colouring algorithms. *Computers & Operations Research*, 39(9):1933–1950, 2012.

[140] W. Li, R. L. Church, and M. F. Goodchild. The p-compact-regions problem. *Geographical Analysis*, 46(3):250–273, 2014.

[141] F. Liberatore, M. Camacho-Collados, and L. Quijano-Sánchez. Equity in the police districting problem: Balancing territorial and racial fairness in patrolling operations. *Journal of Quantitative Criminology*, 38:1–25, 2022.

[142] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):5–55, 1932.

[143] H.-Y. Lin and J.-J. Kao. Subregion districting analysis for municipal solid waste collection privatization. *Journal of the Air & Waste Management Association*, 58 (1):104–111, 2008.

[144] M. Lin, K.-S. Chin, C. Fu, and K.-L. Tsui. An effective greedy method for the meals-on-wheels service districting problem. *Computers & Industrial Engineering*, 106:1–19, 2017.

[145] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the Traveling-Salesman Problem. *Operations Research*, 21(2):498–516, 1973.

[146] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[147] J. F. López-Pérez and R. Z. Ríos-Mercado. Embotelladoras ARCA uses operations research to improve territory design plans. *Interfaces*, 43(3):209–220, 2013.

[148] T. Lotan, D. Catrysse, and D. Van Oudheusden. Winter gritting in the province of antwerp: a combined location and routing problem. *Belgian Journal of Operations Research, Statistics, and Computer Science*, 36(2-3):141–157, 1996.

[149] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search: Framework and applications. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, chapter 12, pages 129–168. Springer, New York, United States, 2nd edition, 2019.

[150] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations research*, 53(6):1007–1023, 2005.

[151] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 1, pages 281–297, Oakland, United States, 1967.

[152] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, New York, United States, 1990.

[153] C. McCartan and K. Imai. Sequential Monte Carlo for sampling balanced and compact redistricting plans. *Preprint arXiv:2008.06131*, 2020.

[154] M. W. McConnell. The redistricting cases: Original mistakes and current consequences. *Harvard Journal of Law and Public Policy*, 24:103, 2000.

[155] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114, 1998.

[156] J. Min and D. Savage. Why do american indians vote democratic? *The Social Science Journal*, 51(2):167–180, 2014.

[157] F. K. Miyazawa, P. F. Moura, M. J. Ota, and Y. Wakabayashi. Partitioning a graph into balanced connected classes: Formulations, separation and experiments. *European Journal of Operational Research*, 293(3):826–836, 2021.

[158] M. Moeini and O. Wendt. A heuristic for solving the maximum dispersion problem. In A. Fink, A. Fügenschuh, and M. J. Geiger, editors, *Operations Research Proceedings 2016*, pages 405–410, Cham, Switzerland, 2018. Springer.

[159] M. Moeini, D. Goerzen, and O. Wendt. A local search heuristic for solving the maximum dispersion problem. In N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham, and B. Trawiński, editors, *Intelligent Information and Database Systems*, volume 10751 of *Lecture Notes in Computer Science*, pages 362–371, Cham, Switzerland, 2018. Springer.

[160] M. C. O. Moreira, J. A. Ferreira Neto, C. J. Einloft, and N. T. C. Silva. *Desenvolvimento Rural, Sustentabilidade e Prdenamento Territorial,* chapter O uso da busca tabu no ordenamento territorial em assentamentos rurais: reconfigurando o SOTER-PA (Sistema de Ordenamento Territorial da Reforma Agrária e Planejamento Ambiental), pages 265–272. Suprema, Visconde do Rio Branco, Brazil, 2011.

[161] S. Moreno, J. Pereira, and W. Yushimito. A hybrid k-means and integer programming method for commercial territory design: a case study in meat distribution. *Annals of Operations Research*, 286(1):87–117, 2020.

[162] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. C3P Report 826, California Institute of Technology, Pasadena, United States, 1989.

[163] J. G. Moya-García and M. A. Salazar-Aguilar. Territory design for sales force sizing. In R. Z. Ríos-Mercado, editor, *Optimal Districting and Territory Design*, volume 284 of *International Series in Operations Research and Management Science*, chapter 10, pages 191–206. Springer, Cham, Switzerland, 2020.

[164] H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer. Evolution algorithms in combinatorial optimization. *Parallel Computing*, 7(1):65–85, 1988.

[165] J. M. Mulvey and M. P. Beck. Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348, 1984.

[166] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[167] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[168] L. Muyldermans, D. Cattrysse, D. van Oudheusden, and T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139 (3):521–532, 2002.

[169] L. Muyldermans, D. Cattrysse, and D. V. Oudheusden. District design for arc-routing applications. *Journal of the Operational Research Society*, 54(11):1209–1221, 2003.

[170] R. G. Niemi, B. Grofman, C. Carlucci, and T. Hofeller. Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *The Journal of Politics*, 52(4):1155–1181, 1990.

[171] B. Nygreen. European assembly constituencies for Wales – comparing of methods for solving a political districting problem. *Mathematical Programming*, 42 (1):159–169, 1988.

[172] J. Oehrlein and J.-H. Haunert. A cutting-plane method for contiguity-constrained spatial aggregation. *Journal of Spatial Information Science*, 2017 (15):89–120, 2017.

[173] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23(2):166–204, 1981.

[174] E. Parliament and T. C. of the European Union. Directive 2012/19/EU on waste electrical and electronic equipment (WEEE), Jul. 2012. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2012:197:0038:0071:EN:PDF.

[175] F. Pezzella, R. Bonanno, and B. Nicoletti. A system approach to the optimal health-care districting. *European Journal of Operational Research*, 8(2):139–146, 1981.

[176] D. D. Polsby and R. D. Popper. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law & Policy Review*, 9(2): 301–353, 1991.

[177] J. Poot, O. Alimi, M. P. Cameron, and D. C. Maré. The gravity model of migration: The successful comeback of an ageing superstar in regional science. Technical Report IZA DP No. 10329, IZA Institute of Labor Economics, Bonn, Germany, October 2016.

[178] C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

[179] O. A. Prokopyev, N. Kong, and D. L. Martinez-Torres. The equitable dispersion problem. *European Journal of Operational Research*, 197(1):59–67, 2009.

[180] E. C. Reock. A note: Measuring compactness as a requirement of legislative apportionment. *Midwest Journal of Political Science*, 5(1):70–74, 1961.

[181] M. G. C. Resende and C. C. Ribeiro. *Optimization by GRASP: Greedy Randomized Adaptive Search Procedure*. Springer, New York, United States, 2016.

[182] M. G. C. Resende and R. F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1):59–88, 2004.

[183] F. Ricca and B. Simeone. Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426, 2008.

[184] F. Ricca, A. Scozzari, and B. Simeone. Weighted Voronoi region algorithms for political districting. *Mathematical and Computer Modelling*, 48(9-10):1468–1477, 2008.

[185] F. Ricca, A. Scozzari, and B. Simeone. Political districting: From classical models to recent approaches. *Annals of Operations Research*, 204(1):271–299, 2013.

[186] R. Z. Ríos-Mercado. Assessing a metaheuristic for large-scale commercial districting. *Cybernetics and Systems*, 47(4):321–338, 2016.

[187] R. Z. Ríos-Mercado, editor. *Optimal Districting and Territory Design*, volume 284 of *International Series in Operations Research and Management Science*. Springer, Cham, Switzerland, 2020.

[188] R. Z. Ríos-Mercado and J. F. Bard. An exact algorithm for designing optimal districts in the collection of waste electric and electronic equipment through an improved reformulation. *European Journal of Operational Research*, 276(1):259–271, 2019.

[189] R. Z. Ríos-Mercado and H. J. Escalante. GRASP with path relinking for commercial districting. *Expert Systems with Applications*, 44:102–113, 2016.

[190] R. Z. Ríos-Mercado and E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[191] R. Z. Ríos-Mercado and J. F. López-Pérez. Commercial territory design planning with realignment and disjoint assignment requirements. *Omega*, 41(3):525–535, 2013.

[192] R. Z. Ríos-Mercado and J. C. Salazar-Acosta. A GRASP with strategic oscillation for a commercial territory design problem with a routing budget constraint. In

I. Batyrshin and G. Sidorov, editors, *Advances in Soft Computing*, volume 7095 of *Lecture Notes in Artificial Intelligence*, pages 307–318. Springer, Heidelberg, Germany, 2011.

[193] R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás, A. Castrillón, and M. C. López-Locés. A location-allocation-improvement heuristic for districting with multiple-activity balancing constraints and p-median-based dispersion minimization. *Computers & Operations Research*, 126:105106, 2021.

[194] R. Z. Ríos-Mercado, J. L. González-Velarde, and J. R. Maldonado-Flores. Tabu search with strategic oscillation for improving collection assignment plans of waste electric and electronic equipment. *International Transactions of Operational Research*, 30(2):1002–1030, 2023.

[195] M. Ritt and J. Pereira. Heuristic and exact algorithms for minimum-weight non-spanning arborescences. *European Journal of Operational Research*, 287 (1):61–75, 2020.

[196] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, pages 269–280, 1981.

[197] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and M. Cabrera-Ríos. New models for commercial territory design. *Networks and Spatial Economics*, 11(3):487–507, 2011.

[198] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. A bi-objective programming model for designing compact and balanced territories in commercial districting. *Transportation Research Part C: Emerging Technologies*, 19(5):885–895, 2011.

[199] M. A. Salazar-Aguilar, J. L. González-Velarde, and R. Z. Ríos-Mercado. A divide-and-conquer approach to commercial territory design. *Computación y Sistemas*, 16(3):309–320, 2012.

[200] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, J. L. González-Velarde, and J. Molina. Multiobjective scatter search for a commercial territory design problem. *Annals of Operations Research*, 199(1):343–360, 2012.

[201] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. Grasp strategies for a bi-objective commercial territory design problem. *Journal of Heuristics*, 19(2):179–200, 2013.

[202] M. G. Sandoval, J. A. Díaz, and R. Z. Ríos-Mercado. An improved exact algorithm for a territory design problem with p-center-based dispersion minimization. *Expert Systems with Applications*, 146:113150, 2020.

[203] M. G. Sandoval, E. Álvarez-Miranda, J. Pereira, R. Z. Ríos-Mercado, and J. A. Díaz. A novel districting design approach for on-time last-mile delivery: An application on an express postal company. *Omega*, 113:102687, 2022.

[204] O. B. Schoepfle and R. L. Church. A new network representation of a "classic" school districting problem. *Socio-Economic Planning Sciences*, 25(3):189–197, 1991.

[205] M. Schröder. *Gebiete optimal aufteilen*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2001.

[206] J. Schwartz. Cracked, stacked and packed: Initial redistricting maps met with skepticism and dismay. https://indyweek.com/news/northcarolina/cracked-stacked-packed-initial-redistricting-maps-met-skepticism-di 2011. Retrieved in 12-12-2022.

[207] J. A. Segura-Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás, and K. de Alba Romenus. A location-allocation heuristic for a territory design problem in a beverage distribution firm. In *International Conference on Industrial Engineering Theory, Applications, and Practice*, pages 428–434, 2007.

[208] M. I. Shamos. *Computational Geometry*. PhD thesis, Yale University, New Haven, United States, 1978.

[209] T. Shirabe. Districting modeling with exact contiguity constraints. *Environment and Planning B: Planning and Design*, 36(6):1053–1066, 2009.

[210] K. Smith-Miles, D. Baatar, B. Wreford, and R. Lewis. Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, 45:12–24, 2014.

[211] M. T. A. Steiner, D. Datta, P. J. S. Neto, C. T. Scarpin, and J. R. Figueira. Multiobjective optimization in partitioning the healthcare system of Paraná state in Brazil. *Omega*, 52:53–64, 2015.

[212] N. O. Stephanopoulos and E. M. McGhee. Partisan gerrymandering and the efficiency gap. *University of Chicago Law Review*, 82:831, 2015.

[213] Q. J. Stewart. The development of social physics. *American Journal of Physics*, 18(5):239–253, 1950.

[214] G. Syswerda et al. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, volume 3, pages 2–9, George Mason University, United States, 1989.

[215] M. Takahashi and H. Kita. A crossover operator using independent component analysis for real-coded genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 643–649, Seoul, Korea, 2001. IEEE.

[216] R. E. Tarjan. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6):160–161, 1974.

[217] F. Tavares-Pereira, J. R. Figueira, V. Mousseau, and B. Roy. Multiple criteria districting problems: The public transportation network pricing system of the paris region. *Annals of Operations Research*, 154(1):69–92, 2007.

[218] M. Thorup. Worst-case update times for fully-dynamic all-pairs shortest paths. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of Computing*, pages 112–119, Baltimore, United States, May 2005.

[219] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, Philadephia, United States, 2nd edition, 2014.

[220] User "cmglee" at English Wikipedia. Gerrymandering 9-6. https://commons.wikimedia.org/wiki/File:Gerrymandering_9-6.png, 2010. Retrieved in 12-12-2022.

[221] H. Validi and A. Buchanan. Political districting to minimize cut edges. *Mathematical Programming Computation*, 14:623–672, 2022.

[222] H. Validi, A. Buchanan, and E. Lykhovyd. Imposing contiguity constraints in political districting models. *Operations Research*, 70(2):867–892, 2022.

[223] F. Vanderbeck. Implementing mixed integer column generation. In *Column Generation*, pages 331–358. Springer, 2005.

[224] B. Vangerven, D. Briskorn, D. R. Goossens, and F. C. Spieksma. Parliament seating assignment problems. *European Journal of Operational Research*, 296 (3):914–926, 2022.

[225] G. S. Warrington. Quantifying gerrymandering using the vote distribution. *Election Law Journal*, 17(1):39–57, 2018.

[226] G. R. Webster. Reflections on current criteria to evaluate redistricting plans. *Political Geography*, 32:3–14, 2013.

[227] I. K. White and C. N. Laird. *Steadfast Democrats: How Social Forces Shape Black Political Behavior*. Princeton University Press, Princeton, United States, 2020.

[228] J. C. Williams, Jr. Political districting: A review. *Regional Science*, 74(1):13–40, 1995.

[229] S. Yanik, Ö. Sürer, and B. Öztayşi. Designing sustainable energy regions using genetic algorithms and location-allocation approach. *Energy*, 97:161–172, 2016.

[230] H. P. Young. Measuring the compactness of legislative districts. *Legislative Studies Quarterly*, 13(1):105–115, 1988.

[231] A. A. Zoltners and P. Sinha. Sales territory alignment: A review and model. *Management Science*, 29(11):1237–1256, 1983.

[232] A. A. Zoltners, P. Sinha, and S. E. Lorimer. Sales force effectiveness: a framework for researchers and practitioners. *Journal of Personal Selling & Sales Management*, 28(2):115–131, 2008.

[233] B. Çavdar and J. Sokol. A distribution-free TSP tour length estimation model for random graphs. *European Journal of Operational Research*, 243(2):588–598, 2015.