Memetic Algorithms for Spatial Partitioning Problems

SUBHODIP BISWAS and FANGLAN CHEN, Virginia Tech, USA
ZHIQIAN CHEN, Mississippi State University, USA
CHANG-TIEN LU and NAREN RAMAKRISHNAN, Virginia Tech, USA

Spatial optimization problems (SOPs) are characterized by spatial relationships governing the decision variables, objectives, and/or constraint functions. In this article, we focus on a specific type of SOP called spatial partitioning, which is a combinatorial problem due to the presence of discrete spatial units. Exact optimization methods do not scale with the size of the problem, especially within practicable time limits. This motivated us to develop population-based metaheuristics for solving such SOPs. However, the search operators employed by these population-based methods are mostly designed for real-parameter continuous optimization problems. For adapting these methods to SOPs, we apply domain knowledge in designing spatially aware search operators for efficiently searching through the discrete search space while preserving the spatial constraints. To this end, we put forward a simple yet effective algorithm called swarm-based spatial memetic algorithm (SPATIAL) and test it on the school (re)districting problem. Detailed experimental investigations are performed on real-world datasets to evaluate the performance of SPATIAL. Besides, ablation studies are performed to understand the role of the individual components of SPATIAL. Additionally, we discuss how SPATIAL is helpful in the real-life planning process and its applicability to different scenarios and motivate future research directions.

CCS Concepts: • **Mathematics of computing** → *Combinatorial optimization*; • **Theory of computation** → **Optimization with randomized search heuristics**;

Additional Key Words and Phrases: Metaheuristic, redistricting, spatial optimization

5

1 INTRODUCTION

Spatial optimization has been an active research area, especially in disciplines such as economics, engineering, environmental studies, geography, operational research, and regional science. Church [18] noted that "spatial optimization involves identifying how land use and other activities should

be arranged and organized spatially to optimize efficiency or some other quality measure." It includes many districting, layout, location, and network problems that involve design, operations, and planning [15]. Solving a **spatial optimization problem (SOP)** is equivalent to searching for an optimal assignment of a set of discrete spatial units representing some geographic areas such that some well-defined objectives and/or constraints are satisfied. Alternatively, we can also define this as the use of mathematical or computational techniques for finding solutions to geographic decision problems subjected to design constraints [58]. The optimization variables in a SOP relate to the decision being made with the objective function quantifying the quality of the decision. The constraints impose a set of necessary design considerations that needs to be satisfied. The functions and constraints usually encode spatial properties/topological relationships, including adjacency, contiguity (connectivity), similarity (distance), shape (compactness), and so on [88].

Broadly, SOPs are usually classified as either a *selection* or a *partitioning* problem [94]. Spatial selection problems identify a subgroup of spatial units. Additional spatial constraints need to be satisfied for certain problems, while others only impose continuations on the selected spatial units. Spatial partitioning problems, however, seek to group the spatial units into a number of districts or territories. For instance, consider the districting problem [47], where the objective is to partition a geographic area into groups of contiguous districts (regions) such that each district is balanced with respect to some activity measure, like residing population. Due to the discrete nature of spatial units, SOPs suffer from combinatorial explosion, i.e., the phenomenon where the computing time cost to find the optimal solution of a NP-hard problem increases exponentially with the problem size [32, 63]. Thus, exact optimization techniques like **Integer Programming (IP)** or **mixed-integer Linear Programming (MILP)** cannot solve the problem optimally under practical time constraints [36]. This is why researchers often resort to using approximation methods like *heuristics* and *metaheuristics*, since these methods can find good, but not necessarily optimal, solutions to the problem in a reasonable time. Thus, computational efficiency is the key to designing these methods for solving SOPs [95].

Heuristic methods are mainly designed for solving a particular problem. For instance, let us consider the p-Median Problem (pMP) in location sciences, where the aim is to find facilities or services on p nodes of a network such that the distance from each node to its nearest facility or service node is minimized [37]. Teitz and Bart [87] proposed a heuristic called the vertex-exchange algorithm to solve this problem. This heuristic starts with a randomly selected subset of p nodes from the network and keeps flipping these nodes with unselected ones until such exchange can no longer improve the quality of the solution. In contrast to heuristics, metaheuristics refer to a general problem-solving framework that is composed of a set of high-level problem-independent instructions or strategies for developing heuristic optimization algorithms [85]. Some well-known examples of metaheuristics include evolutionary algorithms [43], simulated annealing [52], tabu search [34], variable neighborhood search [40], and so on. Oftentimes, these methods are inspired by some natural processes, and they can be adapted to solve different kinds of problems. Hence, metaheuristics have become a popular choice amongst practitioners and researchers for solving medium to large instances of combinatorial optimization problems [12], especially in location sciences [66].

Motivated by this, we devise a simple, easy-to-use population-based metaheuristic inspired by the emerging field of Swarm Intelligence.[1] In particular, our approach builds on top of the Artificial

---

[1]Swarm Intelligence is defined as the study and design of computational optimization techniques based on the collective intelligence emerging from a large population of search agents with simple behavioral patterns for communication and interaction. These methods instantiate search moves the closely mimic the complex social behavior of animals such as ant colonies, beehives, bird flocks, and so on [20].

Bee Colony algorithm that is based on the foraging behavior of the swarm of bees [1]. It maintains multiple trial solutions to the SOP under consideration and combines a local search technique with a spatially aware recombination operator resulting in what is commonly known as a *memetic algorithm* [68]. Hence the name **swarm-based spatial memetic algorithm (SPATIAL)**. The search move of the algorithm is modified to explore the discrete search space while preserving the spatial relationships/constraints. preliminary version of this work appeared in Reference [9]. In this article, we present further additions based on theoretical and experimental investigations of our framework. The summary of the extensions and contributions are elucidated below.

- An overview of background details is provided in Section 2, especially in context of spatial partitioning problems like districting in Section 2.1. We then show in Section 2.2 spatial partitioning problems are accompanied by an underlying graph structure that enables the problem to be solved as a graph partitioning problem. Section 2.3 briefly reviews graph-partitioning techniques that motivate our algorithmic approach and the role of domain knowledge in algorithm design.
- Section 3 defines the optimization problem corresponding to a generalized spatial partitioning by leveraging the notions from graph partitioning. We also show how the given formulation can be adapted to problems like school districting in Section 3.1 and provide some pointers on how to adapt other types of spatial partitioning problems using the given framework.
- A detailed outline of the SPATIAL method is proposed in Section 4. In particular, we focus on the two improvement steps: *local search* and *spatially aware recombination*. Additionally, the relationship between locals search and the sampling of partitions based on the theory of Markov Chain is discussed. This is followed by an in-depth discussion on how the recombination operation efficiently searches for solutions in the discrete search space.
- We used the dataset for the school year 2020–2021 here as compared to 2019–2020 used in the previous work [9]. At the onset of the pandemic in 2020, many parents unenrolled their children from public schools in these two districts thereby creating a serious imbalance between the student population and the school capacity. This presented more challenging problem instances to work with.
- We included more sophisticated baseline methods and performed an exhaustive comparison in Section 5. Additional discussions on how solution initialization affects performance and how the algorithms can be made more efficient using alternative measures. We also include a case study showcasing the applicability of SPATIAL in real-life planning.

2 BACKGROUND

This section provides a basic outline of the background details necessary for understanding this research. First, an introduction to spatial partitioning problems are provided in Section 2.1. This is followed by Section 2.2 highlighting how graphs can encode relationships between the spatial units in most of the spatial problems. In fact, the graph-based representation can be used to pose spatial partitioning as a graph partitioning problem. A brief review of different graph partitioning approaches is provided in Section 2.3.

2.1 Spatial Partitioning Problems

The field of spatial optimization is firmly rooted in the classical works on graph theory, where mathematical formalism and theories about spatial arrangement and movement were made. In spite of its historic origin, the term *spatial optimization* first appeared in the literature during the late 1960s and the early 1970s, when a series of articles made use of the term within the context

of regional science [2, 61, 62]. Interestingly, this research domain has followed the developments in Computer Science, which is rich in works on graphs and other discrete data structures.

The term "spatial optimization" was initially used by Ghosh and Craig [30] to describe a set of location-allocation problems, namely the warehouse-location problem [60], the p-median problem [74], and the location set-covering problems [89]. Similarly, researchers in resource management use the term spatial optimization to refer to optimization models for allocating various land-use areas within a forest [41, 42]. In fact, spatial optimization problems appear in different disciplines in different context—location sciences [38, 55, 99], regionalization [24, 54], spatial data mining [39, 64], territory design [46, 79, 100], and so on. Most of these developments have a commonality—optimizing an objective function subjected to a number of constraints (spatial or aspatial) that define the feasibility of solutions. These include a large number of simplistic variations of well-known problems, including the location set covering, p-median, simple plant location problems, and so on. Such normative location problems have garnered a lot of interest in spatial optimization, especially from the research community and the industry. In fact, SOPs are too broad to be addressed all at one time and is outside the scope of this work. In this article, we focus on a spatial partitioning problem called *territory design problem*, often popularly known as *districting* or *zone design* in location sciences [55]. Note that the term *redistricting* is also used to refer to these problems. However, redistricting actually means rearrangement of existing territories. For the sake of clarity, we shall use the term "districting" to refer to these problems.

2.1.1 Districting Problems. Districting is a sub-field of discrete optimization involving some form of partitioning decision. In a typical districting problem, a set of smaller geographic areas, called *basic units* or *spatial units*, are group together into larger geographic areas, called *districts* or *territories*, such that they meet a series of planning criteria and requirements as specified from the application or context [77]. Districting problems arise in different real-life application domains, ranging from political districting over the design of districts for police patrols, schools, social facilities, waste collection, or winter services, to sales and service territory design [47].

Based on the application domain, each category of districting problem is unique from the perspective of modeling, criteria, or constraints. Nonetheless, several common criteria, including balance, compactness, and unique assignment, can be generally applied to most districting problems. "Balance" means that a total amount of resources need to be fairly allocated among the districts. The term resources implies a particular attribute or multiple attributes of each spatial unit. Examples include the number of customers, product demand, population size, workload, and so on. "Compactness" aims to obtain districts composed of basic units with geographical proximity, which can be optimized by minimizing a dispersion function that measures how tightly the area of a district is packed within its perimeter. Unique assignment indicates that each spatial unit must be assigned to only one district, and this requirement assures a complete partitioning of all the spatial units. Additionally, territory "contiguity" needs to be considered while designing the districts/territories.

Interestingly, there is no single approach to model the aforementioned criteria. Therefore, existing works have studied various methods to represent these requirements. During the 1960s to 1980s, the majority of research has focused on sales territory design [100] and political districting problems [76]. Since the early 1990s, a lot of studies on other applications, such as distribution territory design, service-related districting, and, more recently, districting in health care [98], have emerged. Scant attention has been paid to the problem of school districting till now.

School districting. In countries like the U.S., *school districts* play a vital role in the operation of the public school systems. A school district is an administrative unit for overseeing the jurisdiction of public schools and represent a large geographical region that is coterminous with the boundary of a county, city, or a subdivision. The spatial configuration of a school district is composed of smaller
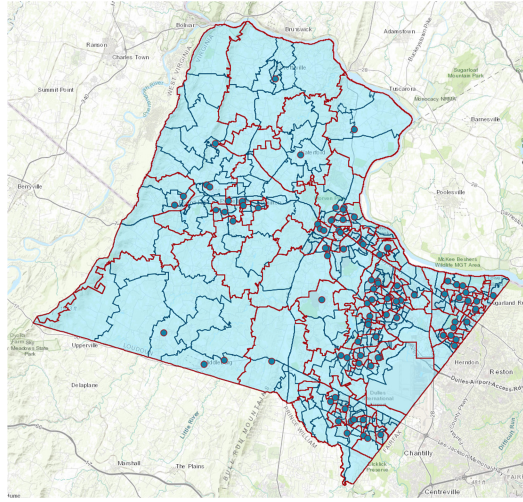
Fig. 1. ArcGIS visualization of the school district of Loudoun county, VA, USA. The smaller polygons (with blue-colored border) represent the SPAs, while the larger polygons (with brown-colored border) represent the SAZs of elementary schools. The dark blue dots represent the locations of all the public schools.

spatial units called planning zones or **student planning areas (SPAs)**. These SPAs are grouped to form a larger geographically contiguous area, called the **school attendance zone (SAZ)**, which defines the boundary of a school. The schools at every grade-level (elementary, middle, and high) have a well-defined boundary often arranged in a hierarchical manner. In a school district, the rule of thumb is that students attend the school assigned to their residing SPA. Figure 1 illustrates a map of a school district along the school locations, school boundaries, and constituent SPAs. Note that in districting problems, a large geographical area, like a county, is partitioned in multiple "districts" or territories. However, in school districting, the term "school district" refers to the entire geographic area, like county or city. To avoid this confusion, one can imagine a school district being composed of contiguous districts or territories, each of which represent a school boundary.

School districting is the process by which the boundaries of public schools (within a school district) are adjusted/redrawn in response to projected growth/decline of student enrollment, change in school capacities, opening/closure of a school, and so on. This is an annual/biannual event that involves the school boundary planners, board members, parents, and other stakeholders and takes up a significant amount of time in reaching a consensus about the final districting plan to implement. Multiple factors (geographic, economic, social) are considered in deciding the school boundaries, thereby making school districting a technically and socially challenging to solve. The complexity of the process piqued the curiosity of the research community, especially in Operations Research.

Sutcliffe et al. [86] summarized the work in this direction until the early 1980s. Since then, not many works have been reported in this direction [27, 56, 81]. Among the few, Schoepfle and Church [81] introduced the term *Generic School Districting Problem*, which refers to a class of school boundary problems involving allocation of students to schools while minimizing a cost/distance function, subject to a set of balancing/activity constraints. Most of the approaches to school redistricting usually solves a continuous **Linear Programming (LP)** or a derived transportation problem to get an optimal or a near-optimal solution (which requires split-resolution) [6, 57]. However, the computational bottleneck does not allow these methods to scale. Caro et al. [16] proposed an IP approach
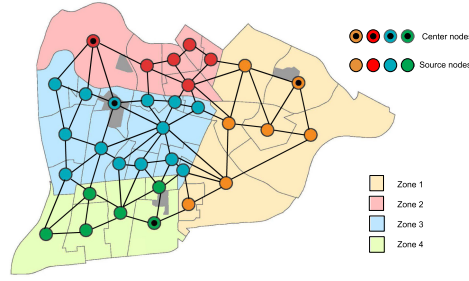
Fig. 2. A large geographical area composed of smaller spatial units can be represented by a planar connected graph. The color coding represents an instance of spatial partitioning, where the each color correspond a territory/zone obtained by partitioning the graph into connected subgraphs.

to the school districting problem by minimizing the total distance travelled by the students. Their model was inspired by the sales territory alignment model in Reference [100] and was perhaps the first approach to account for all the problem-specific constraints, including connectivity. Their model was applied to only 2 (of 22) clusters that the school district of the City of Philadelphia is divided into.

2.2 Graph-based Representation in Spatial Optimization

A geographical area composed of smaller-sized spatial units can be represented as nodes of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes representing the N smaller-sized spatial units and $\mathcal{E}$ is the set of edges connecting adjacent nodes. $\mathcal{G}$ is commonly called the *contiguity graph* or the *dual graph*. It is a planar connected graph with the nodes encoding the spatial entities and the edges capturing the spatial adjacency relationship between the entities. An edge connects two nodes if their corresponding spatial units share a common boundary (more than a single point). Figure 2 illustrates a toy example depicting the graph-based encoding of a geographical area.

A node v may also be represented by its index, i.e., $v \in \{1, 2, \ldots, N\}$. These nodes may have features, i.e., $\mathbf{F}(\mathcal{G}) = \{F_1, F_2, \ldots, F_N\}$, where F_v is the set of features corresponding to node v. Let F_v be represented by a tuple (L_v, A_v), where $L_v = [(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t), (x_1, y_1)]$ is the list of geographic coordinates (latitude-longitude) that define the boundary polygon of the vth spatial unit and A_v is the vector of feature values. Usually, a similarity matrix $\mathbf{W}(\mathcal{G}) = (W_{uv})_{u=1,\ldots,N \ v=1,\ldots,N}$ captures the relationship between any pair of nodes. Popular choices for the similarity metric include the distance function or the binary adjacency matrix. Similarity can also be defined for edges connecting adjacent nodes.

Solving a SOP involves the search for a feasible solution $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, such that the spatial configurations of $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$ satisfy predefined problem criteria/constraints while minimizing certain objective(s). In spatial partitioning problems like districting, we seek a K-partition of the graph $\mathcal{G}$, i.e., $\mathcal{E}' \subset \mathcal{E}$ and $\mathcal{V}' = \mathcal{V}$ such that the nodes in $\mathcal{V}'$ are grouped into K connected subgraphs. Figure 2 shows an instance where a geographic area, encoded by a graph containing $N = 33$ nodes, is partitioned into $K = 4$ territories, each of which is represented by a connected subgraph. Thus, the spatial partitioning problem is equivalent to the **graph partitioning problem (GPP)** [14] described next.

Given a positive integer $K \in \mathbb{N}_{>1}$ and an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with non-negative edge weights, $\omega : \mathcal{E} \to \mathbb{R}_{>0}$, the solution to a GPP seeks a partition Π of $\mathcal{G}$ with blocks of nodes $\Pi = (V_1, \ldots, V_K)$ such that $V_1 \cup \ldots \cup V_K = \mathcal{V}$ and $V_i \cap V_j = \phi \ \forall i, j \in \{1, 2, \ldots, K\}, i \neq j$. Alternatively, the output of a GPP can be represented by a plan on $\mathcal{G}$ described by an assignment

function $\xi : \mathcal{V} \rightarrow \{1, 2, \ldots, K\}$, where $\xi(v) = i$ implies that node v is assigned to block i. A node v is a neighbor of node u if there is an edge $(u, v) \in \mathcal{E}$. If a node $v \in V_i$ has a neighbor $w \in V_j$, $i \neq j$, then v is called boundary node. Correspondingly, an edge that connects two boundary nodes is called *cut edge* and $\mathcal{E}_{ij} = \{(u, v) \in \mathcal{E} : u \in V_i, v \in V_j\}$ is the set of cut edges between two blocks, namely V_i and V_j. An extra balance constraint may exist and enforces that all blocks have roughly equal weights, i.e., it requires that $\forall i \in \{1, \ldots, K\} : |V_i| \leq (1+\tau)|\mathcal{V}|/K$ for some threshold parameter $\tau \in \mathbb{R}_{\geq 0}$. Sometimes we also use weighted nodes with node weights. Weight functions on nodes and edges are extended to sets of such objects by summing their weights. Note that a clustering is also a partition of the nodes. However, K is usually not given in advance, and the balance constraint is removed. Note that a partition is also a clustering of a graph. In both cases, the goal is to minimize a particular objective called the *dispersion function*. This is also identical to the connected K-partition problem, which partitions a graph into K connected sub-graphs where $K \geq 3$, is a well-known NP-hard problem [25].

2.3 Approaches

In computer science, graphs constitute a preferred abstraction when modeling an application problem. Even if the application involves a different problem, partitioning a graph into smaller subgraph is often an important fundamental operation that helps to reduce problem complexity. Next, we briefly survey the different class of GP techniques broadly, an important end-application relevant to the spatial partitioning problems, and discuss the role of domain knowledge in designing spatial search techniques.

(1) *Global algorithms* seeks a partition by directly working on the entire graph. Well-known global methods include *exact algorithms* [4, 13, 21, 28, 48, 59, 82] that rely on the branch-and-bound techniques [53], *spectral partitioning* techniques based on eigendecomposition of the Laplacian matrix, *graph growing* approach based primarily on breadth-first search, *flow-based methods* that make use of the max-flow min-cut theorem, and, last, the *geometric partitioning* techniques that utilize the coordinates of graph nodes in space [31, 84, 92]. The global algorithms are more suited to smaller graphs owing to high computation time, especially the exact methods. Also, these methods are mostly confined to bipartitioning but can be generalized to k-partitioning when k is small.

(2) *Iterative heuristics* start with an initial solution and tries to improve it through a variety of search operations. *Local search* is the most widely used approach that updates a given solution by selecting a new one from the neighborhood. Different ways of defining the neighborhood and the selection strategy gives rise to a variety of techniques. Initial methods like the KL/FM method [29, 50] was more suited for graph bisection. Later, k-way extensions to this method were proposed [49, 71]. Most existing local search algorithms swaps nodes between adjacent blocks of the partition trying to minimize a dispersion function. This results in highly restrictive scope of possible improvement. For instance, the METIS approach cannot create balanced and contiguous partitions [97].

(3) *Multi-level approaches* perform graph partitioning by varying the granularity of the graphs [80, 90, 91]. It consists of the three main phases: coarsening, initial partitioning, and uncoarsening. Coarsening helps to reduce the problem size by iteratively approximating the original input graph with fewer degrees of freedom. This translates to substituting the parallel edges in the input graph with a single edge in the coarsened graph. Coarsening is terminated when the original graph is sufficiently small enough to be *initially partitioned* using some (possibly expensive) exact methods discussed earlier. *Uncoarsening* happens in two steps. First, the partition in the coarse-level graph is translated back to a fine-level graph.

Then, iterative improvement methods (discussed earlier) are usually applied to improve the partition. While multi-level approaches are successful for partitioning large graphs, it becomes challenging to tune this methods for graphs with fixed centers and highly varying degree of balance between the partitions.

(4) *Metaheuristics* have been increasingly applied to the GP domain recently [51]. There is a twofold advantage of using metaheuristics. First, these frameworks are defined in a general sense and hence can be modified to suit the needs of real-life optimization problems in terms of practical constraints like solution quality and execution time. Second, metaheuristics do not put any restriction on the optimization problem formulation (like constraints/ objective functions to be expressed as a linear function of the decision variables). Our focus is on *population-based metaheuristics* like **Evolutionary Algorithms (EAs)**, which are derivative-free global optimization methods inspired by the process of natural evolution [26]. An EA starts by initializing a population of trial solutions to the optimization problem, then it tries to improve the solutions via search operations, like recombination and selection, until a termination criteria is reached. In our work, we augment EAs with local search techniques for solving SOP.

Capacity-Constrained Network-Voronoi Diagram. Problems like districting, especially school districting, can be treated as a **Capacity Constrained Network-Voronoi Diagram (CCNVD)**: "Given a graph and a set of service center nodes, a CCNVD partitions the graph into a set of contiguous service areas that meet service center capacities and minimize the sum of the shortest distances from graph nodes to allotted service centers" [96, 97]. For the school districting problem, the service centers represent the spatial units containing schools inside them. The **Pressure Equalizer (PE)** algorithm and its variants were proposed by Yang et al. [96, 97] to address CCNVD. However, some important differences do exist. In the PE approach, the objective was to minimize the sum of the shortest distances from graph nodes to their allotted service centers. Additionally, the following assumptions were made: All service centers have the same capacity, each non-service-center node has unit demand and all the service centers together could serve the demand of all the non-service-center nodes at any point of time. These simplistic assumptions may not apply for problems like districting. For instance, in school districting, the capacity of the schools and the student population corresponding to the graph nodes can vary considerably. Also, compactness is preferred to distance-based measures due to arbitrary shapes of spatial units forming a school district [11].

2.3.1 Hybrid Metaheuristics.
Recently, researchers try to assimilate ideas from different classes of metaheuristics into a "hybrid" framework. One such framework, called *memetic algorithm* [67], combines the local search technique with the recombination operator of EAs to balance exploration-exploitation. Thus, memetic algorithms benefit from the synergy between iterative improvement (exploitation) of the local search and the recombination operation (exploration) of the population-based methods. We take a step in this direction by integrating a randomized local search within a swarm-intelligent algorithm that mimics the foraging behavior of honeybees [1]. However, adapting EAs to SOPs is non-trivial due to the following: *First*, EAs are designed to solve continuous-valued real-parameter global optimization problems. As such, they employ linear search moves for exploring the decision space by perturbing the incumbent solutions. This strategy is hardly suitable for discovering promising solutions in the discrete decision space of SOPs. *Second*, the presence of spatial constraints (topological properties), like contiguity, make SOPs highly constrained in nature and harder to find feasible solutions. In fact, the infeasible solutions significantly outnumber the feasible solutions with an increase in problem size. This often results in expending tremendous computational effort in finding a feasible solution, especially

when the search operators are not spatially cognizant. *Last*, EAs tend to reinitialize the solutions when they stagnate or violate constraints. In SOPs, such a move is detrimental to preserving the goodness of solutions and lead to loss of valuable information.

2.3.2 Domain Knowledge for Spatial Search. In view of the above challenges, it is increasingly impracticable for vanilla EAs to solve SOPs [44]. This is mostly because the linear search moves are not suited to the discrete nature of problems we encounter. Besides, the traditional constraint handling techniques used in conjunction with EAs are of little help [65]. Hence, we use domain knowledge to guide the search process. Domain knowledge refers to any auxiliary information that may enable a metaheuristic to efficiently search for feasible solutions. It includes both *model-specific* and *problem-specific* instructions. Next, we discuss how model-specific domain knowledge is helpful in conducting spatially aware search within an EA framework.

The first step in solving SOPs is to instantiate initial feasible solution(s) and then improve the solution(s) locally by flipping spatial units between the adjacent (neighboring) territories [69, 70]. There are two types of possible moves: (a) *move one unit* from its present (donor) territory to a neighboring (recipient) territory and (b) *swap/ exchange units* between two neighboring territories. The new solution produced by these moves are kept only if it is feasible and leads to an improvement in the objective function. Additionally, the local nature of the moves restricts the exploration of the decision space beyond the immediate neighborhood of the incumbent solution. However, these moves may also result in breaking the spatial contiguity of the territories involved in the move, thereby leading to an infeasible solution. Path relinking can be helpful in such scenarios for repairing the solutions if they enter the infeasible search space [35]. When infeasible solutions are made feasible again via repair operation, these solution(s) undergo strategic oscillations between the feasible and infeasible decision space and may find better intermediate solutions [33].

3 SPATIAL PARTITIONING AS AN OPTIMIZATION PROBLEM

The optimization formulation corresponding to spatial partitioning problems can be written as

$$(\textbf{P}_0) \quad \underset{\mathbf{X} \in \mathcal{X}}{\text{minimize}} \quad \mathcal{J}\left(\mathbf{X}\right) = \sum_{u=1}^{N} \sum_{v=1}^{N} X_{uv} \cdot D_{uv}, \tag{1a}$$

s. t.

$$\sum_{u=1}^{N} X_{uv} = 1, \qquad \forall v = 1, 2, \ldots, N, \tag{1b}$$

$$\sum_{u=1}^{N} X_{uu} = K, \tag{1c}$$

$$(1 - \tau)\, \mu \cdot X_{uu} \le \sum_{v=1}^{N} X_{uv} \cdot A_v, \qquad \forall u = 1, 2, \ldots, N, \tag{1d}$$

$$\sum_{v=1}^{N} X_{uv} \cdot A_v \le (1 + \tau)\, \mu \cdot X_{uu}, \qquad \forall u = 1, 2, \ldots, N, \tag{1e}$$

$$\sum_{v \in \bigcup_{l \in S} \mathcal{N}^l \setminus S} X_{uv} - \sum_{v \in S} X_{uv} \ge 1 - |S|, \forall u = 1, 2, \ldots, N,\ S \subset \{1, 2, \ldots, N\} \setminus (\mathcal{N}^u \cup \{u\}) \tag{1f}$$

$$X_{uv} \in \left\{0, 1\right\}, \qquad \forall u = 1, 2, \ldots, N,\ \forall v = 1, 2, \ldots, N, \tag{1g}$$

where u and v are the indices corresponding to the nodes of graph $\mathcal{G}$; $\mathbf{X} \in \{0, 1\}^{N \times N}$ is a binary assignment matrix, where $X_{uu} = 1$ implies that node u is center node of a given subgraph; $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the distance (or dissimilarity) matrix, where D_{uv} is a distance[2] between nodes u and v; $\mathbf{A} \in \mathbf{R}_+^{N \times 1}$ is the activity matrix, where A_v is an activity measure with respect to node v. ($\mathbf{P_0}$) is a constrained optimization problem with binary decision variables encoded by $\mathbf{X}$. Solving ($\mathbf{P_0}$) exactly will output an optimal partitioning plan encoded by the solution $\mathbf{X}^*$ that minimizes the objective function (1a), i.e., $\mathcal{J}(\mathbf{X}^*) \leq \mathcal{J}(\mathbf{X}), \forall \mathbf{X} \in \mathcal{X}$, subjected to a set of constraints (1b)–(1f). $\mathcal{X}$ is the set of all possible partitioning plans, i.e., the assignment of N spatial units to K territories as show in Figure 2. As spatial partitioning is analogous to graph partitioning, we shall interchangeably use the following groups of terms: spatial units/nodes and territories/subgraphs.

Constraint (1b) enforces that each node is assigned uniquely to a subgraph. Constraint (1c) ensures that the number of center nodes is K implying that are exactly K subgraphs, since each subgraph has an unique center node. Constraints (1d)–(1e) restricts the total activity measure in a given subgraph to lie within a range of the mean activity measure μ (which is computed as $\mu = \frac{1}{K} \sum_{v=1}^{N} A_v$) as measured by the tolerance parameter τ. Usually the value of τ is kept in the range $[0.01, 0.1]$ depending on the application. These constraints are designed to ensure that a given subgraph has zero activity if it does not contain any center node, i.e., $X_{uu} = 0$. Last, the contiguity constraint (1f) that ensures that each subgraph is connected, where $\mathcal{N}^u$ refers to all the nodes adjacent to a given node u. The connectivity constraint ensures that each territory is geographically contiguous, i.e., we can travel between any two points within a territory without crossing over to another adjacent territory. However, connectivity is expressed by an exponential number of comparisons, i.e., $O(KN2^N)$ [23], thereby making it impracticable to exactly solve moderate to large instances of this problem within a reasonable amount of time.

Remarks. Equation (1) can be solved by exact methods, like LP, IP, and MILP [7]. However, in trying to do so we made some interesting observations. *First*, these methods minimize a linear objective function based on dispersion, as in the p-median problem or the p-center problem [79]. These linear measures of dispersion cannot account for optimizing the non-linear compactness metric. *Second*, the exponentially big connectivity constraints may not guarantee territorial contiguity. *Last*, it may be difficult to find feasible solutions when the bound constraints (1d) and (1e) are tightened by setting the value of τ to be sufficiently low.

Computational complexity. Though the problem ($\mathbf{P_0}$) with exponential number of connectivity constraints, if we are given a p partition $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, then we can check whether each subgraph of $\mathcal{G}'$ is connected or not in polynomial time by using breadth-first-search algorithms. The feasibility of a solution to the problem can be verified in polynomial time, i.e., $\mathbf{P_0}$ is NP. Next, let us consider a particular instance where $\mathcal{G}$ is a planar connected graph. If we take high values of tolerance parameter τ, then we can ensure that the balancing constraints (1d) and (1e) are always satisfied. Then it becomes a pMP, which is a well-known NP-hard problem [37]. Since pMP is reducible to $\mathbf{P_0}$ in polynomial time, we can say that $\mathbf{P_0}$ is NP-hard. Interested readers may refer to [25] for an in-depth analysis of the computational complexity.

3.1 School Districting

The school districting problem follows a spatial partitioning structure where the spatial units (or SPAs) are the nodes of a graph and the school boundaries (or SAZs) are the balanced, connected subgraphs. Hence, we reformulate (1) to define the school districting problem. We do make some

---

[2]Normally, the Euclidean distance between the centroids of two spatial units is considered.

adjustments in the optimization model based on the above-mentioned remarks and problem-specific domain knowledge. We shall visit them in turn.

Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a school district with N SPAs and K schools. The number K varies with the school level L := **Elementary School (ES)**, **Middle School (MS)**, or **High School (HS)**, since we solve the problem at each level independently. Each node (SPA) can be represented as $A_v = ({}_{\mathrm{ES}}p_v, {}_{\mathrm{MS}}p_v, {}_{\mathrm{HS}}p_v, {}_{\mathrm{ES}}c_v, {}_{\mathrm{MS}}c_v, {}_{\mathrm{HS}}c_v)$, $v \in \{1, 2, \ldots, N\}$, where ${}_{\mathrm{L}}p_v$ is the student population residing in SPA i corresponding to the school level L (ES, MS, HS), and ${}_{\mathrm{L}}c_v$ is the program capacity of the schools contained in the same SPA. We assume that all the schools in a school district follow a consistent grade structure with respect to the school levels. For majority of the SPAs that do not enclose any school inside them, we have ${}_{\mathrm{ES}}c_v = 0$, ${}_{\mathrm{MS}}c_v = 0$, and ${}_{\mathrm{HS}}c_v = 0$. We consider a set of center nodes $\overline{\mathcal{V}} = \{\overline{v} | X_{\overline{v}\overline{v}} = 1\}$, $\overline{\mathcal{V}} \subset \mathcal{V}$ corresponding to the SPAs containing schools inside them. Alternatively, we may write $\overline{\mathcal{V}} = \{\overline{v}_1, \overline{v}_2, \ldots, \overline{v}_K\}$, where $\overline{v}_i$ is the node containing the ith school.

While drawing school boundaries, the following must be considered. *First*, each school has a different capacity to accommodate students. This is equivalent to the bound constraints (1d) and (1e), except that mean activity measure μ is replaced by the corresponding school's capacity ${}_{\mathrm{L}}c_v$. *Second*, Euclidean distance between the centroid of the spatial units in (1a) may not be a good representative of the commute time due to the widely varying shapes of these units. Compactness measures that take into account the geometric shape can be a better alternative. These two considerations are linearly weighed using the weight factor λ, $0 \le \lambda \le 1$, in formulating the optimization problem,

$$
\underset{\mathbf{X} \in \mathcal{X}}{\text{minimize}} \; \mathcal{J}_s\big(\mathbf{X}\big) = \lambda \underbrace{\sum_{i=1}^{K} \left| 1 - \frac{\sum_{u=1}^{N} X_{u\overline{v}_i} \cdot {}_{\mathrm{L}}p_u}{\sum_{u=1}^{N} X_{u\overline{v}_i} \cdot {}_{\mathrm{L}}c_u} \right|}_{\text{target balance (aspatial)}} + (1 - \lambda) \underbrace{\sum_{i=1}^{K} \left| 1 - \frac{4\pi \cdot \mathrm{Area}\big(\bigcup_{u=1}^{N}\{u | X_{u\overline{v}_i} = 1\}\big)}{\big[\mathrm{Peri}\big(\bigcup_{u=1}^{N}\{u | X_{u\overline{v}_i} = 1\}\big)\big]^2} \right|}_{\text{target compactness (spatial)}}.
$$

(2)

Several remarks are in order. (1) The *target balance* measures the discrepancy between the schools' capacity and their attending student population. It attains the ideal value of 0 when every school's attending student population (numerator) is equal to its program capacity (denominator). Note that target balance consolidates the bound constraints (1d) and (1e) into an objective or a soft constraint. This helps to deal with solutions that cannot satisfy both these constraints in (1) by penalizing them heavily. (2) The *target compactness* measures how far is a school's boundary from a perfectly compact shape (like a circle). We use the non-linear Polsby–Popper score [73], which is the ratio of the area of a zone to the area of a circle whose circumference is equal to the perimeter of the zone. The more compact the school boundaries become, the closer the value of target compactness gets to 0. (3) Most of the school (re)districting happen to balance the student populations between existing schools. Hence, *target balance* is given more weightage than *target compactness*. In our setting, we always ensure that $\lambda/1-\lambda \ge 2$. (4) Prefixing the center nodes $\overline{\mathcal{V}}$ helps to satisfy the hard constraint (1c) and automatically reduce the size of the optimization problem. Due to this advantage, we prefer to use problem-specific domain knowledge to perform prefixing. In the absence of any such information, clustering algorithms like K-medoids [72] can be used to determine a set of initial center nodes. (5) The remaining hard constraints, i.e., mutually exclusive assignment of nodes (1b) and subgraph connectivity (1f), can be satisfied when initializing a feasible solution $\mathbf{X}$ and then perturbing it locally. (6) In solving Equation (2), the minimizer seeks a K-partition of a geographical area such that the territories are well-balanced, compactly-shaped and geographically contiguous. Overall, this approach generalizes to other spatial partitioning problems, like commercial territory design [78] and political redistricting [93], that involve optimization of similar dispersion metric and subjected to constraints like balance, contiguity, and compactness.
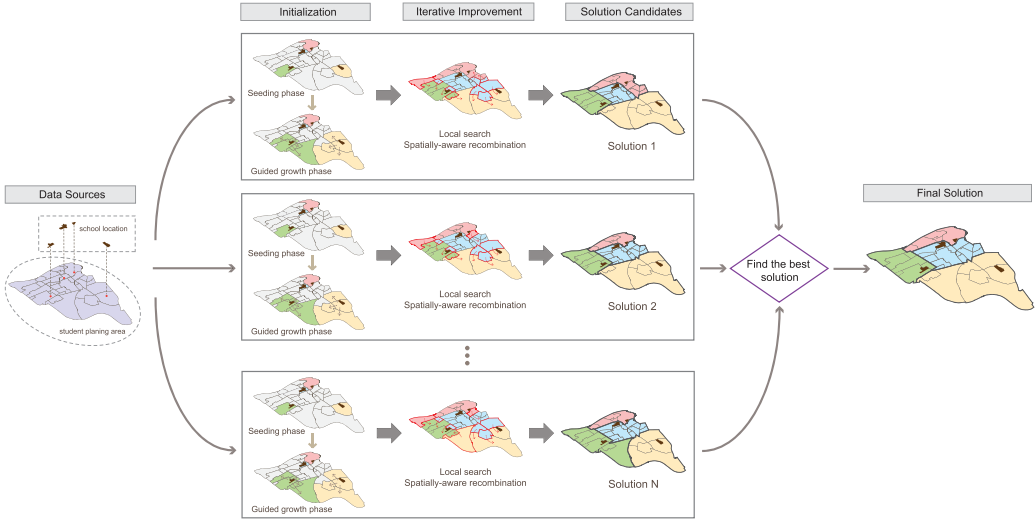
Fig. 3. Outline of the SPATIAL approach for solving the school redistricting problem.

(a) Seeding phase   (b) Guided growth phase
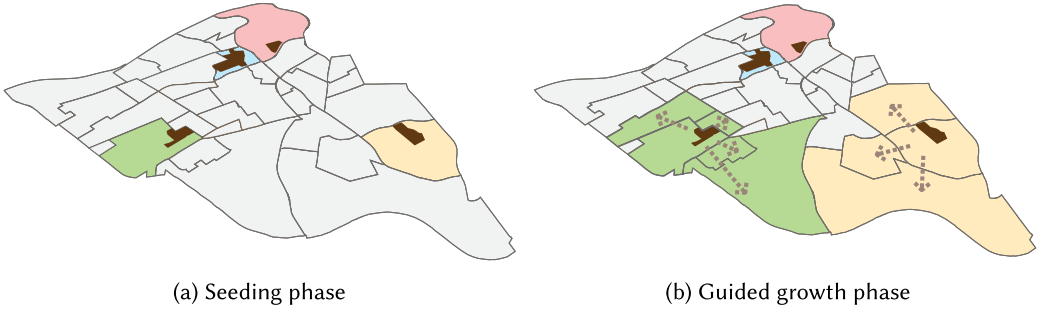
Fig. 4. The seeding phase (a) followed by the guided-growth phase (b) results in a new solution. Seeding identifies the spatial unit corresponding to the center nodes. The guided-growth phase helps to grow the territories by assigning the free spatial units (marked in light grey) based on the adjacency relation.

4 THE SPATIAL ALGORITHM

In this section, we describe our SPATIAL method for solving spatial partitioning problems. SPATIAL starts by initializing a population of randomly generated trial solutions as detailed in Section 4.1. These solutions are iteratively improved till a termination criterion is met. The improvement takes place in two phases, *local improvement* and *spatially aware recombination*, as detailed in Sections 4.2 and 4.3, respectively. The outline of our approach illustrated in Figure 3.

4.1 Initialization of Trial Solutions

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the initialization module instantiates a set of Np trial solutions, $\mathcal{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(Np)}\}$, where the ith solution, $\mathbf{X}^{(i)}$, represents a particular partitioning of $\mathcal{G}$. Note that we are overloading the notation on $\mathbf{X}$ The feasibility of these solutions are ensured by the *seeding phase* followed by the *guided growth phase* as shown in Figure 4. The pseudocode of initialization is provided in Algorithm 1.

Seeding. This step helps to identify the seed units (prefix the center nodes) by leveraging problem-specific domain knowledge and assign each such unit (center node) to an unique territory

---

ALGORITHM 1: Initialization

Input   : Contiguity graph $\mathcal{G}$, Population size Np, School level L.
Output : $\mathcal{X}$: Population of trial solutions
begin

  Determine the center nodes $\overline{\mathcal{V}}$ for school level L and set $K \leftarrow |\overline{\mathcal{V}}|$
  $\mathcal{X} \leftarrow \{\}$ ▷ Empty population
  for $i = \{1, 2, \ldots, Np\}$ **do**

    $\mathcal{V}$: Get the set of nodes in $\mathcal{G}$
    Seeding phase ▷
    Set an initial assignment, i.e., $\mathbf{X}^{(i)} \in 0^{N \times N}$
    for $\overline{v} \in \overline{\mathcal{V}}$ **do**

      $X^{(i)}_{\overline{v}\overline{v}} \leftarrow 1$ ▷ Assignment
      $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\overline{v}\}$

    Guided-growth phase ▷
    do

      Randomly pick a center node $\overline{v}$, $\overline{v} \in \overline{\mathcal{V}}$
      Determine the subgraph V containing $\overline{v}$, i.e., $V = \{u | u \in \mathcal{V}, X^{(i)}_{u\overline{v}} = 1\}$
      Find unassigned nodes adjacent to V, i.e., $\mathcal{N}(V) = \{u | u \in \mathcal{V}, \sum_{v \in \mathcal{V}} X^{(i)}_{uv} = 0\}$
      while $|\mathcal{N}(V)| > 0$ **do**

        u: Randomly select a node from $\mathcal{N}(V)$
        $X^{(i)}_{u\overline{v}} \leftarrow 1$ ▷ Assignment
        $\mathcal{N}(V) \leftarrow \mathcal{N}(V) \setminus \{v\}, \mathcal{V} \leftarrow \mathcal{V} \setminus \{v\}$

    while $|\mathcal{V}| > 0$
    $\mathcal{X} \leftarrow \mathcal{X} \bigcup \{\mathbf{X}^{(i)}\}$

  return $\mathcal{X}$

(subgraph). In the context of school districting, seeding identifies each of the K school-containing SPAs as center nodes as shown in Figure 4(a). This leads to creation of K partial subgraphs with just a single node in it, thereby ensuring that constraint (1c) is satisfied. The assignment of these center nodes remain fixed throughout the partitioning process.

Guided growth. In the next step, the adjacency relationship between the spatial units are leveraged to grow the seed units into complete territories. This generates K connected subgraphs representing a K partition of $\mathcal{G}$. Figure 4(b) shows the guided growth phase, where a territory is randomly picked and grown by adding an adjacent (unassigned) spatial unit to it. The process is repeated until all the spatial units have been assigned to a territory, thereby satisfying constraints (1b) and (1f).

Note that during the growth phase, the adjacent spatial units are added in a random manner to grow the territories without consideration for the quality of the trial solutions generated. Hence, it is more than likely that the trial solutions will have low solution quality. To improve these solutions, we perform two steps of refinement discussed next.

4.2 Local Improvement

The local improvement searches the immediate neighborhood of an incumbent solution for improving the solution quality. Given the ith solution, $\mathbf{X}^{(i)}$, we randomly pick a pair of subgraphs V_z and V_w ($w, z \in \{1, 2, \ldots, K\}$, $w \neq z$) such that they are adjacent, i.e., $|\mathcal{E}_{z, w}| > 0$. Then we
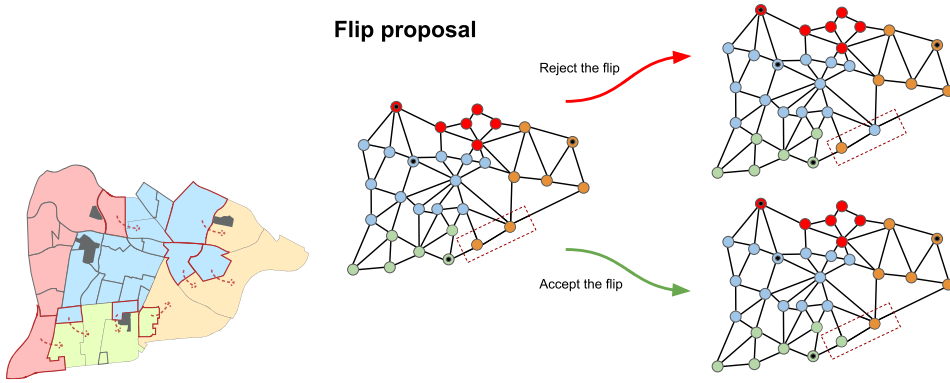
Fig. 5. The local improvement helps to search for neighboring solutions that can be reached by flipping the membership of a boundary node. The Flip proposal involves changing the assignment of a boundary node followed by acceptance/rejection of the new solution.

ALGORITHM 2: Local improvement

Input : Population of solutions $\mathcal{X}$, Contiguity graph $\mathcal{G}$
Output: Updated solution
begin
    for $i = \{1, 2, \ldots, Np\}$ **do**
        $\tilde{\mathbf{X}}^{(i)} \leftarrow \mathbf{X}^{(i)}, flipped \leftarrow False$
        while *not flipped* **do**
            Randomly pick two adjacent subgraphs V_z and V_w, i.e., $|\mathcal{E}_{zw}| > 0$
            Find boundary nodes in V_w: $\mathcal{N}(V_z) = \{v | (u, v) \in \mathcal{E}_{zw}, u \in V_z\}$
            while $|\mathcal{N}(V_z)| > 0$ && *not moved* **do**
                v : From $\mathcal{N}(z)$ pick a random node v
                Move node v from zone V_w to V_z, i.e., $\tilde{X}^{(i)}_{zv} \leftarrow 1, \tilde{X}^{(i)}_{zw} \leftarrow 0$
                if V_w *and* V_z *are contiguous* **then**
                    if $\mathcal{J}(\tilde{\mathbf{X}}^{(i)}) < \mathcal{J}(\mathbf{X}^{(i)})$ $||$ $rand(0, 1) \leq p_r$ **then**
                        $\mathbf{X}^{(i)} \leftarrow \tilde{\mathbf{X}}^{(i)}, flipped \leftarrow True$ ▷ Fitness-based replacement
                  else
                      $\tilde{X}^{(i)}_{zv} \leftarrow 0, \tilde{X}^{(i)}_{zw} \leftarrow 1$ ▷ Revert back the assignment
            $\mathcal{N}(V_z) \leftarrow \mathcal{N}(V_z) \backslash \{v\}$

---

may move a boundary node from V_z to V_w or vice versa. This *flipping* of nodes between adjacent subgraphs result in a new solution $\tilde{\mathbf{X}}^{(i)}$. If this newly produced solution is of better quality, i.e., $\mathcal{J}(\tilde{\mathbf{X}}^{(i)}) < \mathcal{J}(\mathbf{X}^{(i)})$, then $\tilde{\mathbf{X}}^{(i)}$ replaces $\mathbf{X}^{(i)}$ in the population. Note that the connectivity of subgraphs V_z and/or V_w may be broken during flipping, thereby making $\tilde{\mathbf{X}}^{(i)}$ infeasible. To prevent such infeasibility, we only allow a move if it does not break the contiguity of the involved subgraphs. The local improvement operation is illustrated in Figure 5 and the pseudocode is provided in Algorithm 2. Since the local improvement of a solution is independent of other solutions, it can leverage the parallel architecture of the computing platform.

The random selection of subgraphs, i.e., V_z and V_w, for performing node swaps may lead to redundancy. To prevent this, one may sequentially pick a subgraph, say, V_z, from a randomized list
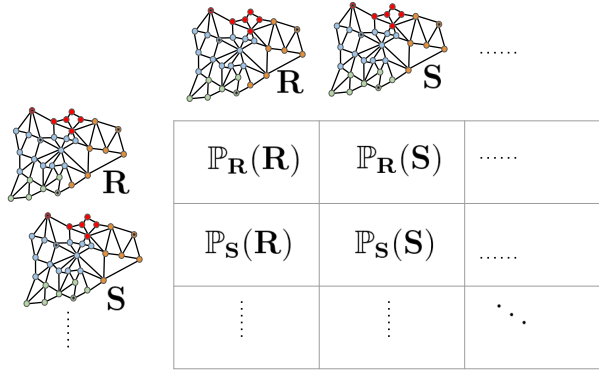
Fig. 6. Theoretically, if the flip proposal is carried out sufficient number of times, then one may approximate the stationary distribution of transitions of the state space, where each state correspond to a districting plan.

of subgraphs and determine adjacent subgraphs, say, V_w, for flipping the node. This is continued until a node flip is made. A flip is made when we find a better neighboring solution or we accept an inferior solution, i.e., $\mathcal{J}(\tilde{X}^{(i)}) > \mathcal{J}(X^{(i)})$, with a very small probability p_r. While the former approach is greedy and prone to getting stuck at local optima, latter one helps in randomization of the search move and is applied in metaheuristics like **Simulated Annealing (SA)** [52].

Markov Chains and Local Search. The local search mechanism here can be thought as instantiating the Flip-based walk, i.e., generating a new solution or *districting plan* by changing the assignment of a single node as shown in Figure 5. Instantiating a series of flips to generate a sample of districting plans is akin to performing a random walk on the states of graph partitions and is encoded by a Flip-based Markov Chain [19]. Relatedly, **Markov chain Monte Carlo (MCMC)** is an effective technique for sampling owing to strong underlying theory, in the form of mixing theorems and convergence properties [22]. In context of redistricting, let us imagine each districting plan representing a *state* and a random walk is being performed on this state space. As the walker traverses from one state to another, we collect each state. On terminating the walk, this collection constitutes the representative sample of the plans. Performing the Flip-based walk involves changing the assignment of individual geographic units along district borders. In the standard MCMC paradigm, altering this basic step adjusts the stationary distribution. Figure 6 gives a rough approximation of the idea.

The main purpose of these sampling-based techniques is to compare the a given districting plan in context of a *representative sample*, i.e., a set of valid alternative plans. Closely following this is the need to relate the sampling distribution to the criteria set forth by domain experts. This may be a tough ask, since any redistricting effort can be accompanied by a varying set of criteria, some of which are difficulty to quantify objectively. Our objective here is different. We use a customized sampling distribution to generate an ensemble of plans and save the best-quality plan, as determined by an objective function.

4.3 Spatially Aware Recombination

During local improvement, the individual solutions are improved independently without any exchange of information between them. Interestingly, it is possible to determine a better intermediate solution by combining features from two solutions. Population-based methods enable mixing of solutions through the recombination operation [26]. This results in better exploration of the search
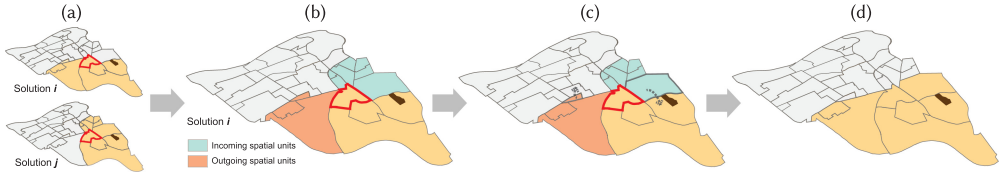
Fig. 7. Illustrating the individual steps involved in the spatially aware recombination operator.

space. However, the vanilla recombination operation is not suitable for spatially constrained problems. Hence, we develop the *spatially aware recombination* operation.

The recombination operation is motivated by the exchange of genetic material between different organisms, which leads to production of offspring. In this process, two (parent) solutions, say, $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$, are selected such that $\mathbf{X}^{(i)}$ is picked randomly while solution $\mathbf{X}^{(j)}$ is selected probabilistically based on the fitness value. The fitness function is defined to allow solutions with lower functional value have higher fitness as this is a minimization problem. For maximization, the fitness can be set equal to the objective functional value. We expect that $\mathbf{X}^{(j)}$ is fitter than $\mathbf{X}^{(i)}$ and thus $\mathbf{X}^{(i)}$ can learn from $\mathbf{X}^{(j)}$. The steps of recombination operation are provided in Algorithm 3.

ALGORITHM 3: Spatially Aware recombination

Input   : $\mathcal{X}$: Population of solutions, $\mathcal{G}$: Contiguity graph
Output : Updated solution
begin

 Find the fitness values: $\mathcal{H}^{(i)} = \frac{1}{1+|\mathcal{J}(\mathbf{X}^{(i)})|}| \ \forall i = 1, 2, \ldots, |\mathcal{X}|$

 for $i = \{1, 2, \ldots, |\mathcal{X}|\}$ **do**

  $\tilde{\mathbf{X}}^{(i)} \leftarrow \mathbf{X}^{(i)}, \tilde{\mathbf{X}}^{(j)}$: Probabilistically selected jth solution based on the fitness value
  Randomly pick a subgraph V such that $0 < |V^{(i)} \cap V^{(j)}| < \min(|V^{(i)}|, |V^{(j)}|)$
  Find the set of incoming nodes $I_V = \{v | v \in V^{(j)} \setminus V^{(i)}$ and $\exists u \in V^{(i)}$ s.t. $(u, v) \in \mathcal{E}\}$ and
  outgoing nodes $O_V = \{u | u \in V^{(i)} \setminus V^{(j)}$ and $\exists v \in V^{(j)}$ s.t. $(u, v) \in \mathcal{E}\}$
  Randomly pick an incoming node $v \in I_V$ and an outgoing node $u \in O_V$
  Simultaneously insert node v into $V^{(i)}$ and remove node u from zone $V^{(i)}$; also update the
  assignments in $\tilde{\mathbf{X}}^{(i)}$
  If $V^{(i)}$ has rendered non-contiguous by the swap operation, repair $\tilde{\mathbf{X}}^{(i)}$

 for $i = \{1, 2, \ldots, |\mathcal{X}|\}$ **do**
  if $\mathcal{F}(\tilde{\mathbf{X}}^{(i)} \leq \mathcal{F}(\mathbf{X}^{(i)})$ **then**
   $\mathbf{X}^{(i)} \leftarrow \tilde{\mathbf{X}}^{(i)}$ ▷ Fitness-based update

---

Suppose a subgraph V is present in both solutions i and j, marked as $V^{(i)}$ and $V^{(j)}$, such that they have a common node. Every subgraph should satisfy this condition, since the center nodes remain unchanged. The subgraph $V^{(i)}$ is modified by simultaneously inserting a node v (present in $V^{(j)}$ but not in $V^{(i)}$) and deleting a node u (present in $V^{(i)}$ but not in $V^{(j)}$). This swapping of node steers solution $\mathbf{X}^{(i)}$ toward the fitter solution $\mathbf{X}^{(j)}$ as illustrated in Figure 7. In doing so, we expect to find intermediate solutions that may have better fitness than the incumbent solution $\mathbf{X}^{(i)}$.

Interestingly, the swapping of the nodes may break the connectivity of the involved subgraphs. To reduce the chances of such undesirable scenarios, we perform the swap operation using boundary nodes. Nevertheless, a repair operation still needs to be applied in case the connectivity of the subgraphs are broken. To repair a solution, we use the breadth-first search traversal
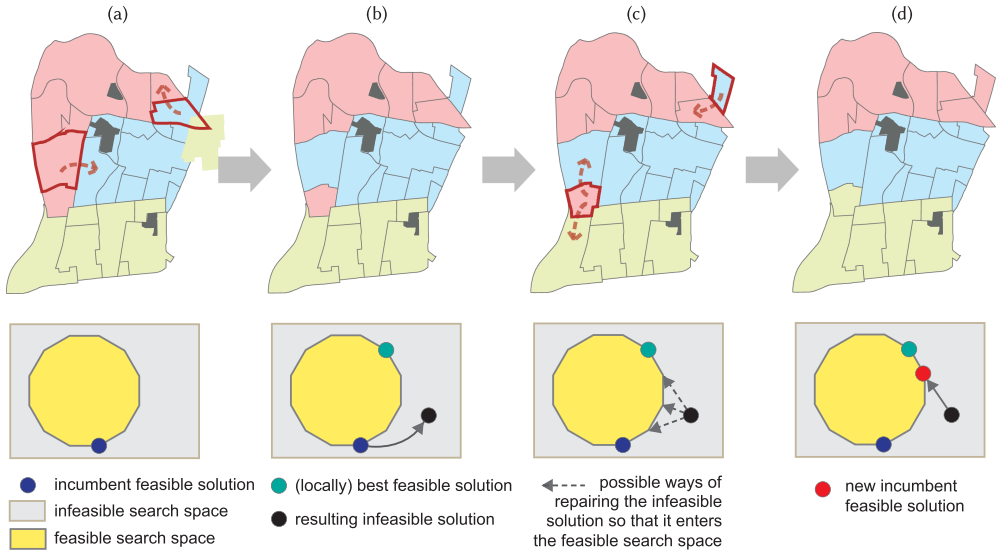
Fig. 8. During the recombination, the swapping of spatial units (a) may result in an infeasible solution (b), which needs to be repaired (c) to make the solution feasible (d). As a result of these moves, the solution oscillates between feasible and infeasible search space as demonstrated above. For illustration purposes, we have depicted this oscillation through a continuous search space. We notice that the recombination plus repair operation may result in a solution that differs from the incumbent solution by multiple hops. As a result, it may approach (locally) optimal solutions.

for enumerating the connected components in the disconnected subgraph, say, V. Then, each connected component is analyzed for the presence of the center node. If center node is absent, then all the nodes in this component is reassigned to the neighboring subgraphs. When no prior information about the center nodes is available, we may retain the largest-sized connected component of V and reassign the other components. The repaired solution $\tilde{\mathbf{X}}^{(i)}$ might be few steps away from the incumbent solution $\mathbf{X}^{(i)}$ in discrete space and thus helps in controlled exploration of the search space. The advantage of repair operation is shown in Figure 8. Note that we have used a depiction of continuous search space in Figure 8 though this is a discrete optimization problem. This was done to simply show the movement of a solution through the repair process.

The solutions newly generated from the recombination operation needs to be updated in the population synchronously based on their fitness values. The solution update is important to keep the fitter solutions in the population so as to make the search progress. With careful implementation, this step can be parallelized in modern computer systems by using multiprocessing.

5 EXPERIMENTATION

In this section, we conduct experiments with the proposed SPATIAL method and some well-known metaheuristics on the school districting problem. The dataset description is provided in Section 5.1 followed by the performance metrics in Section 5.2. Details about setting the objective function and the baseline methods are provided in Sections 5.3 and 5.4, respectively. The comparative evaluation of the baseline methods is performed in Section 5.5 followed by studies on the effect of solution initialization in Section 5.6 and computational analysis in Section 5.7. Next, Section 5.8 focuses on a series of ablation tests for investigating the individual components of SPATIAL. Last, Section 5.9 studies how SPATIAL can aid planners in real-life planning.

Table 1. Summary Statistics of the School Districts for the
School Year 2020–2021

District	#SPAs (N)	#Schools (K)		
		Elementary	Middle	High
X	453	57	16	16
Y	1313	138	26	24

5.1 Dataset

The study was performed on two school districts (counties) located in the mid-Atlantic region of the U.S. These school districts have seen recent population growth in certain areas, thereby making the problem challenging for the SOPs tested here. The following GIS data attributes of both the districts were used for experimentation.

- **SPAs**: The location coordinates of the spatial units along with aggregated student count at different school levels (Elementary, Middle and High).
- **Schools**: The location of the school building, school level, and its program capacity.

In comparison to the dataset used in Reference [9], the only difference is that here we used the data for the school year 2020–2021. We resorted to using the new dataset, since many parents unenrolled their children from the public schools at the onset of the COVID-19 pandemic resulting in more population imbalance. This would present more challenging problem scenarios for the redistricting algorithms. Table 1 presents the summary statistics of this new dataset.

We performed a few additional pre-processing steps. Specifically, we modeled the SPAs as nodes in a graph and generated the adjacency relationship between the nodes. Also, we determined the center nodes, i.e., the nodes corresponding to the spatial units containing schools inside them, by performing point-in-polygon test using the PySal library [75].

5.2 Evaluation Metrics

The solution to the school redistricting problem generated by an algorithm is actually a plan or a zoning configuration of the school boundaries. For evaluating the plan, we utilized two performance metrics that can be interpreted as percentage scores, since they lie in the range [0, 100].

- **Balance** measures the average balance between a school's program capacity and the number of students residing within its boundary. It is calculated as

$$\text{bal}\,(\mathbf{X}) = 100 \times \left| 1 - \frac{1}{K} \sum_{i=1}^{K} \left| 1 - \frac{\sum_{u=1}^{N} X_{u\overline{v}_i} \cdot {}_{\text{L}}p_u}{\sum_{u=1}^{N} X_{u\overline{v}_i} \cdot {}_{\text{L}}c_u} \right| \right|. \tag{3}$$

We penalized both under-enrolled and overburdened schools equally with respect to the capacity of schools. This is an important metric for school planners, since most of the boundary changes occur to achieve a better balance in schools.

- **Compactness** measures how tightly a school's boundary is packed on an average with respect to its perimeter. A scaled Polsby–Popper score [73] is used to measure compactness as

$$\text{com}\,(\mathbf{X}) = \frac{100}{K} \sum_{i=1}^{K} \left| \frac{4\pi \cdot \text{Area}\left(\bigcup_{u=1}^{N} \{u | X_{u\overline{v}_i} = 1\}\right)}{\left[\text{Peri}\left(\bigcup_{u=1}^{N} \{u | X_{u\overline{v}_i} = 1\}\right)\right]^2} \right|. \tag{4}$$

Compact school boundaries often translate to proximal schools that students can walk to and thereby lower the transportation cost incurred by the school district.

5.3 Setting the Objective Function

The two objectives—balance and compactness—are conflicting in nature [11]. Hence an optimizer is needed to tradeoff between the objectives while solving the problem. We ensured that the condition $\lambda/{1-\lambda} \geq 2$ held for both the districts while defining the objective function in Equation (2), i.e., we set the weighing parameter λ in Equation (2) to 0.7 and 0.8 for districts **X** and **Y**, respectively. This could ensure that the importance of population balance is at least twice[3] as that of compactness. The value of λ was higher for district **Y**, since it had higher population imbalance in the schools to begin with. For each school district, we independently solved three instances of the redistricting problem: ES, MS, and HS. We observed varying characteristics of the problem in each instance.

The ES instance of the problem is more challenging than the others due to different factors. First, a school district that has seen recent population growth is most likely due to an influx of young children, which often leads to burgeoning demand for new ESs. Oftentimes, the new ESs are situated at arbitrary locations without sufficient separation between them. This goes against the clustering assumption of well-separated cluster centers. Second, the ESs exhibit a wide variation in their program capacity, with the newly built ones having higher capacity than their older counterparts, thereby making it difficult to balance the student population with the schools' capacity. In attempting to satisfy the schools' capacities, the optimization algorithm may fill in concave segments in the school boundaries with regular-shaped spatial units having a high density of student population. The MS and HS, being well separated and showing fewer deviations of capacity, are comparatively easier to solve. Interestingly, the ES boundaries are more compact than their MS and HS counterparts. In comparison to ES SAZs, a greater proportion of boundaries of MS and HS share borders with the school district's boundary, which is usually zigzagged by naturally occurring geographies (highly irregular geometries). These additional considerations, besides the spatial constraints, make school redistricting a challenging SOP to solve.

5.4 Baseline Methods

The following baseline methods are used for comparative study:

- **Local search-based techniques**: We set the parametric configuration for each of these baselines based on the literature [11].
  - **Stochastic Hill Climbing (SHC)** [45]: A variation of the basic Hill Climbing that searches the immediate neighborhood of a feasible solution in a random manner. If an equally good or a better solution is found, then it replaces and the search continues till a local optimum is obtained.
  - SA [52]: A stochastic version of the Hill Climbing that is based on the process of tempering of metals. It allows for worsening moves to take place if no better solutions are found and can escape local optima.
  - **Tabu Search (TS)** [34]: An algorithm uses a restrictive (tabu) list to forbid revisiting recently explored solutions so that the new neighboring solutions can be explored.
- **Sampling-based techniques**: We include three sampling-based optimization techniques developed based on the link between MCMC and redistricting stated earlier in Section 4.2.
  - **Balanced, Always Accept (BAA)**
  - **Balanced and Compact, Always Accept (BCAA)**
  - **Accept Improving Objective (AIO)**
  For more details about these techniques, refer to Chapter 5 in [8].

---

[3]Balancing a school's attending population w.r.t. its program capacity is the driving forces behind school redistricting. This is consistent with the actual practices of school planners, who give more importance to balance over compactness.

Table 2. Performance of Peer Algorithms on the Problem of School Boundary Formation
in Both the Districts

	District **X**					
Models	Elementary School		Middle School		High School	
	Balance	Compactness	Balance	Compactness	Balance	Compactness
Existing	83.5020 ± 0.0000	32.5344 ± 0.0000	89.7379 ± 0.0000	26.7671 ± 0.0000	87.0786 ± 0.0000	27.3452 ± 0.0000
SA	87.7697 ± 0.8280	38.1032 ± 1.6977	92.3789 ± 0.5726	32.3574 ± 3.6421	96.4240 ± 1.9741	26.9094 ± 2.7749
TS	87.3788 ± 0.7079	36.4537 ± 1.5931	92.5729 ± 0.2888	33.0756 ± 2.1707	95.9435 ± 1.8527	28.3494 ± 2.1524
SHC	86.6755 ± 0.8642	36.5780 ± 1.7189	92.5583 ± 0.1806	32.3115 ± 2.3647	95.6461 ± 1.9090	27.9708 ± 2.1571
BAA	71.7328 ± 2.3193	30.5329 ± 1.5741	89.4501 ± 1.1493	18.6148 ± 1.6085	90.3080 ± 1.8968	16.8370 ± 1.7453
BCAA	71.0820 ± 2.4890	30.7455 ± 1.3047	90.1128 ± 1.5444	18.6181 ± 1.7215	90.9247 ± 1.6280	17.0758 ± 2.0001
AIO	86.6930 ± 1.0247	37.4850 ± 1.5981	92.4021 ± 0.4920	33.4105 ± 2.3518	95.7360 ± 1.7613	30.5854 ± 2.3411
SPATIAL	**87.9353 ± 0.6175**	**38.8988 ± 1.4759**	**92.5926 ± 0.1191**	**37.5398 ± 1.6283**	**97.7948 ± 0.4602**	**31.5193 ± 1.8568**
	District **Y**					
Models	Elementary School		Middle School		High School	
	Balance	Compactness	Balance	Compactness	Balance	Compactness
Existing	82.3835 ± 0.0000	**35.9234 ± 0.0000**	84.2310 ± 0.0000	27.7096 ± 0.0000	86.9541 ± 0.0000	26.8006 ± 0.0000
SA	94.6386 ± 1.0634	30.0782 ± 1.4172	91.3987 ± 1.1006	22.9419 ± 2.4456	93.1795 ± 1.5845	24.2182 ± 2.8013
TS	93.3600 ± 0.7183	29.7212 ± 0.6997	92.1146 ± 0.4123	25.5295 ± 2.8529	**93.6894 ± 1.4057**	26.4599 ± 2.7025
SHC	92.9656 ± 0.9595	29.7697 ± 0.9929	91.4258 ± 0.8386	24.5064 ± 2.5644	93.2757 ± 1.4755	24.3142 ± 2.1289
BAA	67.4731 ± 1.9623	27.1551 ± 0.8379	86.0361 ± 1.2005	10.3140 ± 1.1395	84.8948 ± 2.1824	10.1874 ± 0.8616
BCAA	67.3870 ± 2.1147	27.2899 ± 0.7668	86.6818 ± 1.4137	10.6439 ± 1.1396	85.4392 ± 1.7517	10.1476 ± 0.7313
AIO	93.0592 ± 0.8565	30.7064 ± 0.7959	91.9071 ± 0.5007	25.7025 ± 2.6570	93.4927 ± 1.4288	26.1186 ± 2.2065
SPATIAL	**94.9097 ± 0.4351**	30.2618 ± 0.9105	**92.3780 ± 0.1295**	27.8174 ± 2.2644	92.6337 ± 0.6078	**29.6600 ± 1.5477**

The best performing entries are marked in boldface.

- **SPATIAL** [9]: The population size was set to 10 and 20 for districts **X** and **Y**, respectively. Trial runs were simulated to 1,000 and 2,000 iterations for districts **X** and **Y**, respectively.

Note that we considered two more baseline methods—**Greedy Randomized Adaptive Search Procedure (GRASP)** [78] and MILP [83]. While GRASP's performance was inferior to the other baselines, MILP could not converge to a feasible solution for 4/6 test cases even with a run-time budget of 24 hours. For codes of SA, TS, SHC, and SPATIAL, go to https://github.com/subhodipbiswas/SpatialPartitioning and for the sampling-based techniques, check out https://github.com/subhodipbiswas/SamplingbasedSchoolRedistricting.

5.5 Comparison with Existing Methods

For comparison purposes, we simulated 25 trial runs of each baseline and recorded the final solutions. Each solution represented a districting plan of school boundaries, which were evaluated based on the metrics defined in Section 5.2. In Table 2, we reported the mean and standard deviation of these metrics. We also included the existing school boundary configuration of the school districts. It is marked as Existing in the table. The results revealed that SPATIAL was able to generate better-quality solutions in the majority of the test cases. This especially held for district **X**. Besides achieving better balance, SPATIAL obtained improved compactness score. The difference is especially marked when you compare the Existing plans with the plans generated by SPATIAL.

For district **Y**, the performance of SPATIAL was comparable to the baseline in terms of the balance scores, yet the compactness scores were comparatively better. SPATIAL was the only model that achieves at par or better compactness than the Existing plan. We observed similar trend for district **X**. The baseline methods adopted a greedy approach by continuing to look for better solutions in the local neighborhood of the incumbent solution. In doing so, the solutions lying just outside their immediate neighborhood remained elusive to them. However, the spatially aware recombination technique enabled SPATIAL to find intermediate solutions outside the immediate neighborhood. The repair operation was particularly instrumental in finding such solutions, some

of which may be better than the solutions presented in the immediate neighborhood. This is in line with the findings reported in Reference [33].

The sampling-based methods, with the exception of AIO, do not result in high-quality plans, since they are not optimizing on a particular objective. They are designed to operate like random search methods without a greedy selection procedure. AIO, however, applies greedy selection procedure and in essence similar to the techniques like SA, TS, and SHC.

5.6 Effect of Solution Initialization

To inspect if solution initialization has any effect on the quality of solutions, we created a variant of the baseline methods indicated by an asterisk−Algorithm* is the newly created variant of Algorithm. If, say, SPATIAL starts with randomly generated initial solutions (as depicted in Section 4.1), then the initial solutions in SPATIAL* will correspond to the existing boundary configuration of the school district under consideration. Since the school districts redraw their boundaries frequently (in response to changing needs), the existing boundary configuration represents a (locally) good solution to the problem. This is akin to solving the *school redistricting* problem where we redraw the school boundaries instead of designing them from scratch as in school districting. This subtlety is more of a matter of technicality. Interestingly, while performing school redistricting, we ensured that each school boundary was geographically contiguous to being with. If not, then we would use the repair operation outlined earlier to reassign some SPAs for creating contiguous school boundaries.

We ran 25 simulations on each school level of both the districts and tabulated the performance metrics in Table 3. We observe that on an average the Algorithm* variants are better than Algorithm in both the metrics. This is mostly because the existing solution that Algorithm* starts with is of better quality than the randomly generated solutions used by Algorithm. For district **X**, there is a clear trend showing that Algorithm* is better than Algorithm across all the possible metrics and problem instances. However, we do see some exceptions in district **Y**, especially for MS and HS problem instances. This can be attributed to the Algorithm* variants getting stuck in a local optima. This is highly plausible, since the initial solutions of Algorithm* are very similar[4] to each other. However, Algorithm may manage to escape the local optima courtesy the widely varying range of the starting solutions it is initialized with. The number of local optima increases exponentially with the increase in problem size. Hence, this trend is expected, since district **Y** is almost three times the size of district **X**. Note that both the variants of SPATIAL were able to achieve superior results in majority of the cases.

Nevertheless, the advantage of Algorithm* variants like SPATIAL* is in reconfiguration of the existing school boundaries, i.e., it is useful in scenarios like opening of a new school or closure of an existing school. However, to enable Algorithm* work successfully, it is desired that the existing plan has a high percentage of connected subgraphs. For instance, some of the existing plans of district **Y** did not have geographic contiguity (subgraph connectivity) and hence the disconnected subgraphs have to be repaired. We noticed that a higher proportion of disconnected subgraphs, on being repaired, results in arbitrarily shaped districts. For such districts, it may be difficult to arrive at a better configuration due to the local structure that the problem imposes.

5.7 Computational Complexity and Ideas on Scaling Up

To analyze the computational complexity of these methods, we plotted the wall-clock time for all trial runs in the form of error plots in Figure 9. Runtime analysis reveals that for the smaller

---

[4]The only difference between the solutions are due to the reassignment of discontinuous SPAs of school boundaries for the purposes of maintaining the contiguity of the school boundaries.

Table 3. The Effect of Solution Initialization on Peer Algorithms

| Models | District X | | | | | |
| | Elementary School | | Middle School | | High School | |
	Balance	Compactness	Balance	Compactness	Balance	Compactness
Existing	83.5020 ± 0.0000	32.5344 ± 0.0000	89.7379 ± 0.0000	26.7671 ± 0.0000	87.0786 ± 0.0000	27.3452 ± 0.0000
SA	87.7697 ± 0.8280	38.1032 ± 1.6977	92.3789 ± 0.5726	32.3574 ± 3.6421	96.4240 ± 1.9741	26.9094 ± 2.7749
SA*	87.9809 ± 0.5044	40.9826 ± 1.0907	92.5345 ± 0.2965	37.1108 ± 2.1572	97.3246 ± 0.5107	33.0759 ± 1.8799
TS	87.3788 ± 0.7079	36.4537 ± 1.5931	92.5729 ± 0.2888	33.0756 ± 2.1707	95.9435 ± 1.8527	28.3494 ± 2.1524
TS*	88.2290 ± 0.3177	40.4481 ± 0.5532	92.7145 ± 0.0019	38.1273 ± 0.1339	97.3869 ± 0.1536	32.7663 ± 0.3416
SHC	86.6755 ± 0.8642	36.5780 ± 1.7189	92.5583 ± 0.1806	32.3115 ± 2.3647	95.6461 ± 1.9090	27.9708 ± 2.1571
SHC*	88.0565 ± 0.3398	40.4219 ± 0.6413	92.6503 ± 0.1009	37.9347 ± 1.1750	97.5288 ± 0.6344	33.3857 ± 1.6935
BAA	71.7328 ± 2.3193	30.5329 ± 1.5741	89.4501 ± 1.1493	18.6148 ± 1.6085	90.3080 ± 1.8968	16.8370 ± 1.7453
BAA*	83.0738 ± 0.6002	32.4948 ± 1.0189	92.1247 ± 0.3935	28.4179 ± 0.8139	94.5031 ± 0.7808	26.3284 ± 1.3810
BCAA	71.0820 ± 2.4890	30.7455 ± 1.3047	90.1128 ± 1.5444	18.6181 ± 1.7215	90.9247 ± 1.6280	17.0758 ± 2.0001
BCAA*	84.1403 ± 0.5138	32.2777 ± 1.0458	92.2637 ± 0.3897	29.6189 ± 1.0030	95.5958 ± 0.8448	26.2145 ± 1.6023
AIO	86.6930 ± 1.0247	37.4850 ± 1.5981	92.4021 ± 0.4920	33.4105 ± 2.3518	95.7360 ± 1.7613	30.5854 ± 2.3411
AIO*	87.7953 ± 0.4931	41.0278 ± 0.7990	92.6865 ± 0.0594	37.1659 ± 0.8954	96.6412 ± 1.0957	33.7013 ± 1.7627
SPATIAL	87.9353 ± 0.6175	38.8988 ± 1.4759	92.5926 ± 0.1191	37.5398 ± 1.6283	97.7948 ± 0.4602	31.5193 ± 1.8568
SPATIAL*	**88.2474 ± 0.3008**	**41.6610 ± 0.9948**	**92.6639 ± 0.0642**	**39.9138 ± 0.7408**	**98.0286 ± 0.2052**	**35.2453 ± 1.2159**
Models	District Y					
	Elementary School		Middle School		High School	
	Balance	Compactness	Balance	Compactness	Balance	Compactness
Existing	82.3835 ± 0.0000	35.9234 ± 0.0000	84.2310 ± 0.0000	27.7096 ± 0.0000	86.9541 ± 0.0000	26.8006 ± 0.0000
SA	94.6386 ± 1.0634	30.0782 ± 1.4172	91.3987 ± 1.1006	22.9419 ± 2.4456	93.1795 ± 1.5845	24.2182 ± 2.8013
SA*	95.1757 ± 0.5959	33.9204 ± 0.9149	91.4492 ± 0.6952	28.2519 ± 1.6041	92.2339 ± 0.4015	30.4304 ± 1.2035
TS	93.3600 ± 0.7183	29.7212 ± 0.6997	92.1146 ± 0.4123	25.5295 ± 2.8529	**93.6894 ± 1.4057**	26.4599 ± 2.7025
TS*	93.9882 ± 0.2143	**37.4500 ± 0.4358**	89.8580 ± 0.6495	**31.9823 ± 1.0187**	92.0073 ± 0.1227	34.3255 ± 0.7516
SHC	92.9656 ± 0.9595	29.7697 ± 0.9929	91.4258 ± 0.8386	24.5064 ± 2.5644	93.2757 ± 1.4755	24.3142 ± 2.1289
SHC*	94.2043 ± 0.5056	35.3233 ± 0.6058	90.7112 ± 1.3864	29.9091 ± 1.3810	91.9250 ± 0.0950	32.1575 ± 1.0527
BAA	67.4731 ± 1.9623	27.1551 ± 0.8379	86.0361 ± 1.2005	10.3140 ± 1.1395	84.8948 ± 2.1824	10.1874 ± 0.8616
BAA*	80.1400 ± 0.4253	32.3159 ± 0.4638	81.5796 ± 1.0063	22.8711 ± 0.9850	87.7556 ± 0.6411	22.6101 ± 0.7538
BCAA	67.3870 ± 2.1147	27.2899 ± 0.7668	86.6818 ± 1.4137	10.6439 ± 1.1396	85.4392 ± 1.7517	10.1476 ± 0.7313
BCAA*	81.3120 ± 0.3927	32.0952 ± 0.5303	83.1391 ± 0.9270	22.4642 ± 0.7627	88.9047 ± 0.5496	22.2048 ± 0.7405
AIO	93.0592 ± 0.8565	30.7064 ± 0.7959	91.9071 ± 0.5007	25.7025 ± 2.6570	93.4927 ± 1.4288	26.1186 ± 2.2065
AIO*	94.3157 ± 0.5046	35.8568 ± 0.7109	91.3974 ± 0.9010	31.1680 ± 1.5590	91.8972 ± 0.1067	33.0105 ± 1.1274
SPATIAL	94.6537 ± 0.3691	30.1714 ± 0.6381	**92.3885 ± 0.1194**	28.9142 ± 1.1682	92.6819 ± 0.7202	29.5414 ± 1.3540
SPATIAL*	**95.3731 ± 0.3617**	36.1492 ± 0.7468	91.7854 ± 0.2029	31.8754 ± 1.3194	91.9185 ± 0.1156	**34.6683 ± 0.9838**

The best performing entries are marked in boldface.

District **X**, SPATIAL takes longer time than the local search algorithms SA and SHC. TS is comparatively more expensive than any of the local search methods but much quicker than the sampling-based techniques that simulate 10 million flips. Interestingly, for the larger District **Y**, TS takes the longest time and shows wide variation in the runtime. This high variance can be attributed to a multitude of factors: strong dependence on initial solution, randomized order of search moves, and the tendency to get trapped in local optima. We noticed that the methods applying local search do suffer from these issues. Also, the design of local search and the computation of target compactness may be reasons for a higher computation time. Efficient algorithm design can lead to overcoming these bottlenecks. Interestingly, the sampling-based methods are built on top of the GerryChain library, which is scalable to larger-sized problem instances.

The calculation of the Polsby–Popper compactness measure in techniques like SA, SHC, TS, and SPATIAL, is a key computational bottleneck and it is imperative to look for faster alternatives. We posit some ways in which techniques like SPATIALcan be made faster. *First*, spatial recombination can be promoted at scale by swapping multiple nodes instead of pairwise swap in use presently. This will help to promote better exploration of the search space. *Second*, finding good initial

(a) Elementary school
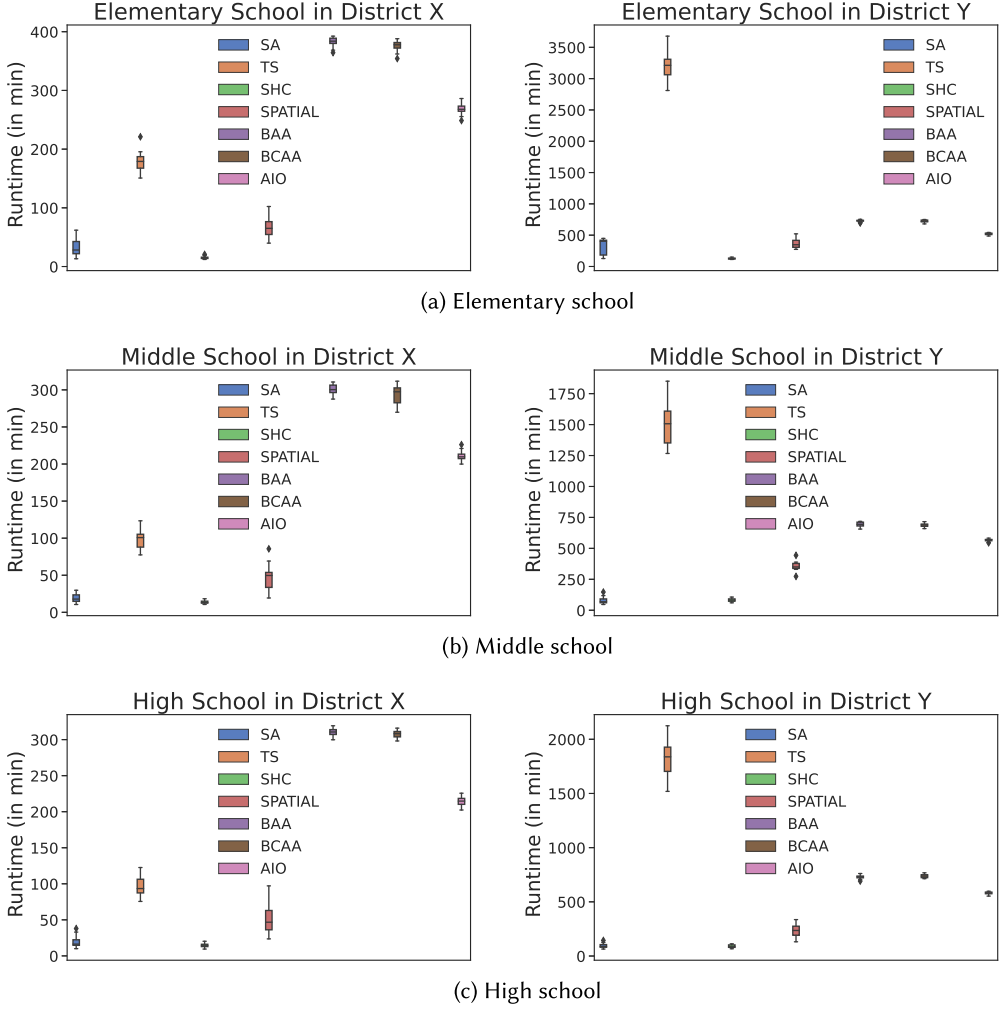
(b) Middle school

(c) High school

Fig. 9. The wall-clock time of the different methods reported for all the problem instances. The increase in the problem size causes significant increase in runtime. This can be attributed to the combinatorial explosion in the search space.

solution can help to accelerate the search significantly. For instance, replacing the random assignment of nodes (in the guided-growth phase in Section 5.6) with a distance-based assignment can lead to more compact territories to being with. *Last*, for large-sized graphs roughly corresponding to a grid structure and having high number of nodes per subgraph, the compactness measure can be closely approximated by the edge cuts.

Figure 10 shows that a partition with compact territories can be obtained by minimizing the number of edge cuts. The idea of edge cuts is a popular concept in graph partitioning literature where algorithms like min-cut exist [14]. While this approximation might not produce compact territories for problems like school redistricting, it might work very well for political districting problems [3, 5].

As shown in Figure 10, to get compact boundaries we need to minimize the number of edge cuts or retain the maximum number of existing edges while partitioning graphs. As such, the number
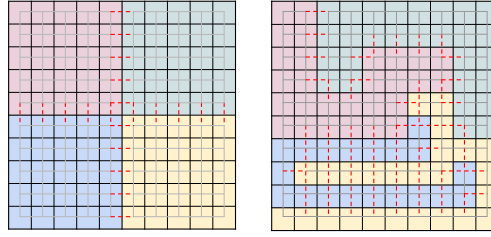
Fig. 10. The dual graph of a 10×10 grid contains 180 edges (marked in grey). The red markers indicate the edges removed to generate the corresponding partitions. Only 20 of these edges will be removed by a partition that divides the grid into four compact territories (left). However, a plan with arbitrary-shaped territories (right) could remove up to 54 edges.

of edges retained can be a proxy to the compactness metric used here. Simulations on the school redistricting problem have revealed 5–10× speedup in the algorithms using local search. However, it comes at an added cost. The compactness of the boundaries suffer, this may lead to unnecessarily elongated school boundaries and consequently increased commute time for students or higher transportation costs for the school administration. However, for designing school boundaries in city blocks, most of which are organized in square blocks, the notion of edge cuts can be useful. Hence, the usage of the edge cuts for the school districting problem should be adopted with caution.

5.8 Ablation Study

5.8.1 How Effective Are the Search Operators? To understand the effectiveness of the search operators (*local search* and *spatially aware recombination*), we simulated 25 sample runs of SPATIAL on district **X** with three possible configurations by selectively activating the operators. They are as follows:

- Only the local search operator is activated.
- Only the recombination operator is activated.
- Both the operators are activated.

The results of each configuration are depicted in Figure 11 as point estimates of evaluation metrics and their corresponding error plots. For fair comparison, we seeded the random numbers to ensure that the starting solutions are similar for the different configurations. Thus, any difference in the performance of these configurations can solely be attributed to the search operators. The local search operator helps in bringing about improvement in balance scores, whereas the recombination operator is responsible for high compactness scores. As mentioned earlier, recombination is less greedy than local improvement and is able to find (better) intermediate solutions beyond the immediate neighborhood of the incumbent solution. Interestingly, when both the operators are active, we noticed that the combined effect of the operators resulted in overall improvement in the quality of solutions. While local improvement resulted in exploitation of the decision space, recombination caused controlled exploration of the search space. Both are important for designing practicable school boundaries.

5.8.2 What Effect Does Population Size Have on Performance? SPATIAL employs a population of trial solutions for solving SOPs. To study the effect of population size Np on the performance of the algorithm, 15 runs of SPATIAL are simulated on the three test cases of district **X** for different values of population size, $Np \in \{10, 20, 30, 40, 50\}$. We observed increasing the population size does not always translate to improvement in performance. This is a classic case of exploration–exploitation tradeoff prevalent in optimization algorithms. A higher population size contributes

(a) Elementary school
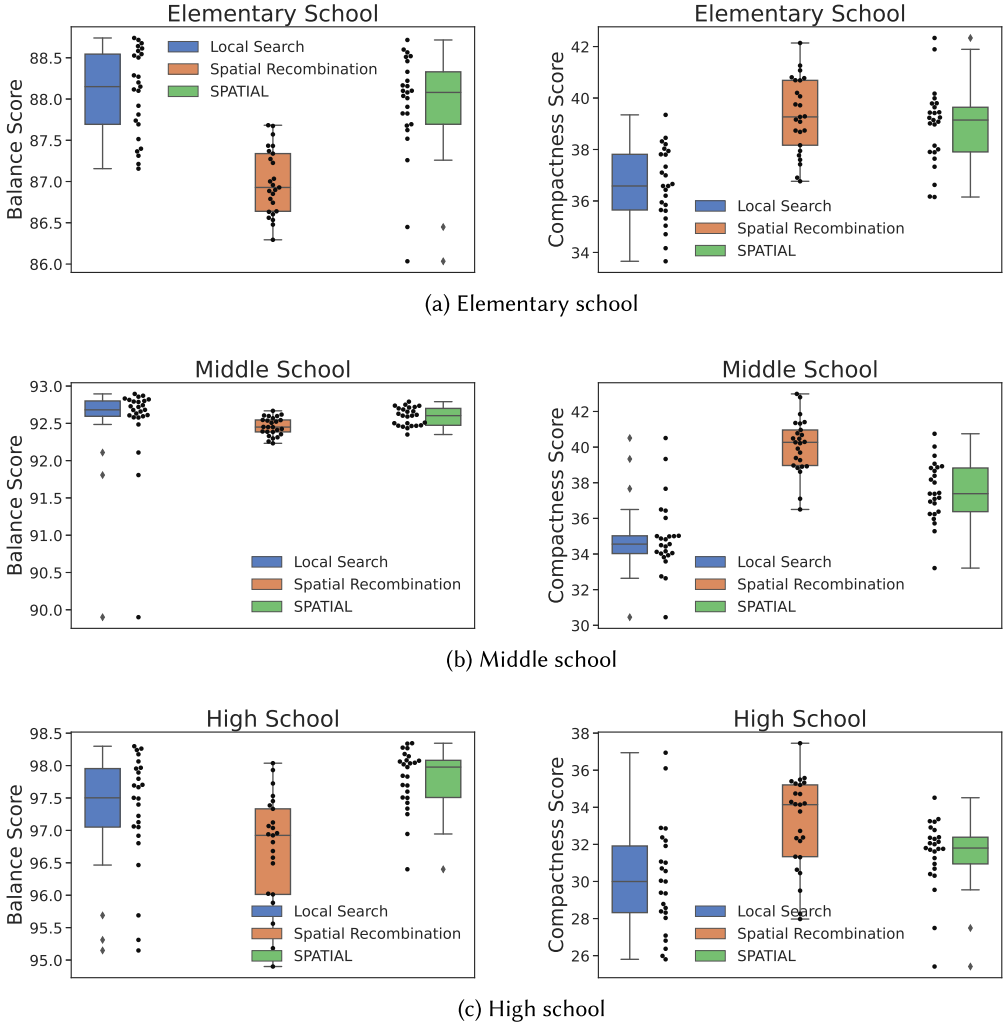

(b) Middle school

(c) High school

Fig. 11. The performance metrics obtained by the different combinations of search operators in SPATIAL. We observed that the combined effect of both the operators resulted in better-quality solutions.

to diversification (exploration), since more solutions can be distributed over the decision space. To enhance intensification (exploitation), the number of iterations need to increase in proportion to the increase in population size. This would result in longer execution time. We came across another related observation: Population-based metaheuristics without recombination operation may not benefit from a large population size, especially when the objective function is multi-modal and the functional landscape has attraction basins with a local minimum. This happened since the solutions present in this attraction basins would quickly eliminate other promising solutions (that could have led to a better optima) due to fitness-based replacement of the solutions. The objective function in Equation (2) presents similar challenges as it is multi-modal in nature. The recombination operation (along with the repair operation) helps to balance such basins of attraction and preserve the solution diversity. Our observation is in line with recent findings by Chen et al. [17].

Table 4. Comparing the Existing Plans with the Automated Plans of District **X**

	Elementary School		Middle School		High School	
	SPATIAL*	Existing	SPATIAL*	Existing	SPATIAL*	Existing
Compactness score	**42.19**	32.53	**41.15**	26.77	**36.07**	27.35
Mean distance traveled (in miles)	0.75	0.75	**1.24**	1.28	1.63	**1.52**
Max distance traveled (in miles)	**9.84**	11.75	**15.51**	17.14	15.84	**14.19**
Balance score	**88.60**	83.50	**92.78**	89.74	**98.27**	87.08
Number of balanced schools	**42**/57	31/57	**15**/17	14/17	**16**/16	14/16
(in %)	**73.68**	54.39	**88.24**	82.35	**100.00**	87.50
Number of under-enrolled schools	**0**/57	1/57	0/17	0/17	0/16	0/16
(in %)	**0.00**	1.75	0.00	0.00	0.00	0.00
Number of overcrowded schools	**15**/57	25/57	**2**/17	3/17	**0**/16	2/16
(in %)	**26.32**	43.86	**11.76**	17.65	**0.00**	12.50
Students displaced	8253/37521	–	2376/20059	–	3269/26728	–
(in %)	22.00	–	11.85	–	12.23	–

5.9 Using SPATIAL in Real-life Planning

In Section 5.6, we show how algorithms like SPATIAL* can be used in redesigning the school boundaries to arrive at an alternative districting plan. Here, we analyze the practical implications of using SPATIAL* in redistricting the school boundaries of both the districts. The evaluation entails including additional metrics beyond the ones defined in Section 5.2.

- **Distance-based metrics:** As an alternative to compactness, we do include some distance-based measures to get an idea of the distance traveled by students to reach schools.
  - Mean distance traveled: This is the average distance a student travels to reach their respective schools. For computing this metric, we weighted the distance between the centroid of the SPA and the assigned school location with the student population in the SPA.
  - Maximum distance traveled: This is the maximum distance between the centroid of a SPA and its assigned school.
- **Balance-based metrics:** In addition to balance metric, we included three more metrics to highlight what proportion of schools are balanced, overcrowded, or under-enrolled.
  - Number of balanced schools: The number of schools in which the attending student population is between 80 and 120% of the school's program capacity.
  - Number of under-enrolled schools: The number of schools in which the attending student population is below 80% of the school's program capacity.
  - Number of overcrowded schools: The number of schools in which the attending student population is above 120% of the school's program capacity.
- **Ethical metrics:** Displacing students should be minimized, since these students lose social ties to their cohorts. Thus, assessing a plan in terms of the social impact it may have on students is equally important. We show how many students may get displaced if a given plan is implemented.

For both the districts, we compare the best automated plan generated by SPATIAL* against the existing plan and tabulate the results in Tables 4 and 5. The results in Table 4 reveal that the automated plans of District **X** have higher compactness values than the existing plan. However, that does not always translate to less distance traveled. In fact, the mean distance traveled by students increases in the automated plan is roughly same as in the existing plan. Interestingly, the automated plans were able to balance a greater proportion of schools thereby relieving the overcrowding/under-enrollment in the present schools. However, the improved balance comes at a cost. The last row reveals that if the automated plan is to be implemented, then 22%, 11.85%, and 12.23% of the

Table 5. Comparing the Existing Plans with the Automated Plans of District **Y**

	Elementary School		Middle School		High School	
	SPATIAL*	Existing	SPATIAL*	Existing	SPATIAL*	Existing
Compactness score	**36.75**	35.92	**34.28**	27.71	**36.44**	26.80
Mean distance traveled (in miles)	0.71	**0.68**	**1.68**	1.77	1.87	1.88
Max distance traveled (in miles)	5.05	**4.13**	**11.75**	12.26	**13.51**	14.02
Balance score	**95.81**	82.38	**92.09**	84.23	**91.90**	86.95
Number of balanced schools	**132**/138	91/138	**22**/26	17/26	**21**/24	18/24
(in %)	**95.65**	65.94	**84.62**	65.38	**87.50**	75.00
Number of under-enrolled schools	**2**/138	17/138	**0**/26	1/26	**3**/24	6/24
(in %)	**1.45**	12.32	**0.00**	3.85	**12.50**	25.00
Number of overcrowded schools	**4**/138	30/138	**4**/26	8/26	**0**/24	0/24
(in %)	**2.90**	21.74	**15.38**	30.77	0.00	0.00
Students displaced	21891/100278	–	6306/28647	–	10749/59593	–
(in %)	21.83	–	22.01	–	18.04	–

students in the elementary, middle, and high schools, respectively, will be displaced with respect to the existing plan. We noticed similar tendencies in Table 5 for District **Y**. The only difference being the percentage of students displaced. District **Y** has a high imbalance to begin with and the school administration deal with this issue by making use of modular classrooms for accommodating the extra students. However, these modular classrooms are expensive and incurs additional operational cost in the schools. The automated plan ended up changing the assignment of 15–20% of the SPAs to achieve a better balance thereby resulting in high value of students displaced.

This massive reshuffling of students is generally not encouraged and may only be done once in a span of 4 to 5 years. Usually, in such scenarios, the final say about which criteria to prioritize lies with the school planners. In doing so, they may consider other factors, including presence of geographic or humanmade barriers, access to walk zones, socioeconomic diversity, the number of students displaced, and so on. In fact, in an ill-defined spatial problem like school districting, any automated plan cannot avoid modification. The common practice is to use arrive at the final plan by modifying a base plan or an automated plan based on subjective judgement.

6 CONCLUSION AND FUTURE WORK

This article proposes a metaheuristic-based approach for solving spatial optimization problems like school districting. We highlight the (computational) difficulty of using exact methods for solving such problems and motivate the need for sophisticated heuristics. To this end, the SPATIAL approach makes use of spatially cognizant search operators for seeking improved solutions by searching through the discrete search space characterized by spatial constraints. We illustrate two key points here. First, we show how the idea of local search has theoretical underpinnings that flow from the idea of MCMC sampling on graph partitions. This even led us to compare additional sampling-based techniques for designing school boundaries. Second, the spatially aware recombination along with the repair operation is instrumental in obtaining better-quality solutions. An in-depth experimental investigation helped ascertain the efficacy of SPATIAL and related methods in solving spatial partitioning problems. We also highlight some existing drawbacks in the framework and provide pointers on ways to improve the framework.

Some possible research directions that can be undertaken in near future are as follows. First, modifying the recombination operation by incorporating multiple swaps with repair can aid in further exploration. Even techniques like ejection-chain methods can be helpful in this regard. Second, developing a multi-objective version of SPATIAL by including multiple decision criteria. This will help to incorporate other ethical considerations, including socioeconomic diversity,

equal opportunity, past displacements, and so on, into the algorithmic model. Third, integrating an sampling-based technique like MCMC with a population-based metaheuristic can help augment the search process by enhancing the diversity of the solutions. Last, we can apply SPATIAL for solving similar SOPs like political districting. This may require modifying the objective function and constraint-handling technique.

ACKNOWLEDGMENTS

REFERENCES

[1] Bahriye Akay, Dervis Karaboga, Beyza Gorkemli, and Ebubekir Kaya. 2021. A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems. *Appl. Soft Comput.* (2021), 107351.

[2] Nurudeen Alao. 1971. Two classes of distance minimization problems: A review, some new interpretations, and extensions. *Geogr. Anal.* 3, 4 (1971), 299–319.

[3] Micah Altman. 1998. *Districting Principles and Democratic Representation.* Ph.D. Dissertation. California Institute of Technology.

[4] Michael Armbruster, Marzena Fügenschuh, Christoph Helmberg, and Alexander Martin. 2008. A comparative study of linear and semidefinite branch-and-cut methods for solving the minimum graph bisection problem. In *Integer Programming and Combinatorial Optimization*, Andrea Lodi, Alessandro Panconesi, and Giovanni Rinaldi (Eds.). Springer Berlin, 112–124.

[5] Amariah Becker and Justin Solomon. 2022. Redistricting algorithms. Springer International Publishing, Cham, 303–340.

[6] Peter C. Belford and H. Donald Ratliff. 1972. A network-flow model for racially balancing schools. *Operations Research* 20, 3 (1972), 619–628.

[7] Dimitris Bertsimas and John N. Tsitsiklis. 1997. Introduction to linear optimization. Vol. 6. Athena Scientific Belmont, MA.

[8] Subhodip Biswas. 2022. Spatial Optimization Techniques for School Redistricting. Ph.D. Dissertation. Virginia Tech. http://hdl.handle.net/10919/110433.

[9] Subhodip Biswas, Fanglan Chen, Zhiqian Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2020. Incorporating Domain Knowledge into Memetic Algorithms for Solving Spatial Optimization Problems. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems (Seattle, WA, USA) (SIGSPATIAL'20)*. Association for Computing Machinery, New York, NY, USA, 25–35. https://doi.org/10.1145/3397536.3422265

[10] Subhodip Biswas, Fanglan Chen, Zhiqian Chen, Andreea Sistrunk, Nathan Self, Chang-Tien Lu, and Naren Ramakrishnan. 2019. REGAL: A Regionalization Framework for School Boundaries. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (Chicago, IL, USA) (SIGSPATIAL'19)*. Association for Computing Machinery, New York, NY, USA, 544–547. https://doi.org/10.1145/3347146.3359377

[11] Subhodip Biswas, Fanglan Chen, Andreea Sistrunk, Sathappan Muthiah, Zhiqian Chen, Nathan Self, Chang-Tien Lu, and Naren Ramakrishnan. 2020. Geospatial Clustering for Balanced and Proximal Schools. *Proceedings of the AAAI Conference on Artificial Intelligence 34, 09 (Apr. 2020)*, 13358–13365. https://doi.org/10.1609/aaai.v34i09.7058

[12] Christian Blum, Jakob Puchinger, Günther R. Raidl, and Andrea Roli. 2011. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing* 11, 6 (2011), 4135–4151.

[13] Lorenzo Brunetta, Michele Conforti, and Giovanni Rinaldi. 1997. A branch-and-cut algorithm for the equicut problem. *Math. Program.* 78, 2 (1997), 243–263.

[14] Aydın Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. 2016. *Recent Advances in Graph Partitioning.* Springer International Publishing, Cham, 117–158. https://doi.org/10.1007/978-3-319-49487-6_4

[15] Kai Cao. 2017. 1.17 Spatial optimization for sustainable land use planning. In *Comprehensive Geographic Information Systems* (2017), 244.

[16] Felipe Caro, Takeshi Shirabe, Monique Guignard, and Andrés Weintraub. 2004. School redistricting: Embedding GIS tools with integer programming. *J. Operat. Res. Soc.* 55, 8 (2004), 836–849.

[17] Tianshi Chen, Ke Tang, Guoliang Chen, and Xin Yao. 2012. A large population size can be unhelpful in evolutionary algorithms. *Theor. Comput. Sci.* 436 (2012), 54–70.

[18] Richard L. Church. 2018. Tobler's law and spatial optimization: Why bakersfield? *Int. Region. Sci. Rev.* 41, 3 (2018), 287–310.

[19] Daryl R. DeFord, Moon Duchin, and Justin Solomon. 2019. Recombination: A family of markov chains for redistricting. arXiv:1911.05725. Retrieved from http://arxiv.org/abs/1911.05725.

[20] Javier Del Ser, Eneko Osaba, Daniel Molina, Xin-She Yang, Sancho Salcedo-Sanz, David Camacho, Swagatam Das, Ponnuthurai N. Suganthan, Carlos A. Coello Coello, and Francisco Herrera. 2019. Bio-inspired computation: Where we stand and what's next. *Swarm Evol. Comput.* 48 (2019), 220–250.

[21] Daniel Delling and Renato F. Werneck. 2012. Better bounds for graph bisection. In *Algorithms − ESA 2012*, Leah Epstein and Paolo Ferragina (Eds.). Springer, Berlin, 407–418.

[22] Persi Diaconis. 2009. The markov chain monte carlo revolution. *Bull. Am. Math. Soc.* 46, 2 (2009), 179–205.

[23] Andreas Drexl and Knut Haase. 1999. Fast approximation methods for sales force deployment. *Manage. Sci.* 45, 10 (1999), 1307–1323.

[24] Juan Carlos Duque, Raúl Ramos, and Jordi Suriñach. 2007. Supervised regionalization methods: A survey. *Int. Region. Sci. Rev.* 30, 3 (2007), 195–220.

[25] Martin E. Dyer and Alan M. Frieze. 1985. On the complexity of partitioning graphs into connected subgraphs. *Discr. Appl. Math.* 10, 2 (1985), 139–153.

[26] Agoston E. Eiben and James Edward Smith. 2003. *Introduction to Evolutionary Computing*. Springer, Berlin.

[27] Jacques A. Ferland and Gilles Guénette. 1990. Decision support system for the school districting problem. *Operat. Res.* 38, 1 (1990), 15–21.

[28] Carlos Eduardo Ferreira, Alexander Martin, C. Carvalho de Souza, Robert Weismantel, and Laurence A. Wolsey. 1998. The node capacitated graph partitioning problem: A computational study. *Math. Program.* 81, 2 (1998), 229–256.

[29] Charles M. Fiduccia and Robert M. Mattheyses. 1982. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*. IEEE, 175–181.

[30] Avijit Ghosh and C. Samuel Craig. 1984. A location allocation model for facility planning in a competitive environment. *Geogr. Anal.* 16, 1 (1984), 39–51.

[31] John R. Gilbert, Gary L. Miller, and Shang-Hua Teng. 1998. Geometric mesh partitioning: Implementation and experiments. *SIAM J. Sci. Comput.* 19, 6 (1998), 2091–2110.

[32] Kenneth C. Gilbert, David D. Holmes, and Richard E. Rosenthal. 1985. A multiobjective discrete optimization model for land allocation. *Manage. Sci.* 31, 12 (1985), 1509–1522.

[33] Fred Glover and Jin-Kao Hao. 2011. The case for strategic oscillation. *Ann. Operat. Res.* 183, 1 (2011), 163–173.

[34] Fred Glover and Manuel Laguna. 1998. Tabu search. In *Handbook of Combinatorial Optimization*. Springer, 2093–2229.

[35] Fred Glover, Manuel Laguna, and Rafael Marti. 2003. *Scatter Search and Path Relinking: Advances and Applications*. Springer US, Boston, MA, 1–35. https://doi.org/10.1007/0-306-48056-5_1

[36] Fred Glover and Kenneth Sörensen. 2015. Metaheuristics. *Scholarpedia* 10, 4 (2015), 6532.

[37] S. Louis Hakimi. 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operat. Res.* 12, 3 (1964), 450–459.

[38] Trevor S. Hale and Christopher R. Moberg. 2003. Location science research: A review. *Ann. Operat. Res.* 123, 1-4 (2003), 21–35.

[39] Jiawei Han, Micheline Kamber, and Anthony K. H. Tung. 2001. Spatial clustering methods in data mining. *Geogr. Data Min. Knowl. Discov.* (2001), 188–217.

[40] Pierre Hansen and Nenad Mladenović. 2018. *Variable Neighborhood Search*. Springer International Publishing, Cham, 759–787. https://doi.org/10.1007/978-3-319-07124-4_19

[41] John G. Hof and Michael Bevers. 1998. *Spatial Optimization for Managed Ecosystems*. Columbia University Press, New York, NY.

[42] John G. Hof and Linda A. Joyce. 1992. Spatial optimization for wildlife and timber in managed forest ecosystems. *Forest Sci.* 38, 3 (1992), 489–508.

[43] John H. Holland. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press. https://doi.org/10.7551/mitpress/1090.001.0001

[44] Catherine M. Hosage and Michael F. Goodchild. 1986. Discrete space location-allocation solutions from genetic algorithms. *Ann. Operat. Res.* 6, 2 (1986), 35–46.

[45] Ari Juels and Martin Wattenberg. 1995. Stochastic hillclimbing as a baseline method for evaluating genetic algorithms. In *Proceedings of the 8th International Conference on Neural Information Processing Systems (NIPS'95)*. MIT Press, Cambridge, MA, 430–436.

[46] Jörg Kalcsics, Stefan Nickel, and Michael Schröder. 2005. Towards a unified territorial design approach-applications, algorithms and GIS integration. *TOP* 13, 1 (2005), 1–56.

[47] Jörg Kalcsics and Roger Z. Ríos-Mercado. 2019. Districting problems. In *Location Science*. Springer, 705–743.
[48] Stefan E. Karisch, Franz Rendl, and Jens Clausen. 2000. Solving graph bisection problems with semidefinite programming. *INFORMS J. Comput.* 12, 3 (2000), 177–191.
[49] George Karypis and Vipin Kumar. 1998. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* 48, 1 (1998), 96–129.
[50] Brian W. Kernighan and Shen Lin. 1970. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Techn. J.* 49, 2 (1970), 291–307.
[51] Jin Kim, Inwook Hwang, Yong-Hyuk Kim, and Byung-Ro Moon. 2011. Genetic approaches for graph partitioning: A survey. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11)*. Association for Computing Machinery, New York, NY, 473–480. https://doi.org/10.1145/2001576.2001642
[52] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220, 4598 (1983), 671–680.
[53] Ailsa H. Land and Alison G. Doig. 2010. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958–2008*. Springer, 105–132.
[54] Philip M. Lankford. 1969. Regionalization: Theory and alternative algorithms. *Geogr. Anal.* 1, 2 (1969), 196–212.
[55] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama (Eds.). 2019. *Introduction to Location Science.* Springer International Publishing, Cham, 1–21. https://doi.org/10.1007/978-3-030-32177-2_1
[56] David S. Lemberg and Richard L. Church. 2000. The school boundary stability problem over time. *Socio-Econ. Plan. Sci.* 34, 3 (2000), 159–176.
[57] Robin Segerblom Liggett. 1973. The application of an implicit enumeration algorithm to the school desegregation problem. *Manage. Sci.* 20, 2 (1973), 159–168.
[58] Arika Ligmann-Zielinska. 2017. *Spatial Optimization.* American Cancer Society, 1–6.
[59] Abdel Lisser and Franz Rendl. 2003. Graph partitioning using linear and semidefinite programming. *Math. Program.* 95, 1 (2003), 91–101.
[60] Alan S. Manne. 1964. Plant location under economies-of-scale-decentralization and computation. *Manage. Sci.* 11, 2 (1964), 213–235.
[61] Duane Francis Marble and Bruce M. Anderson. 1972. *LANDUSE: A Computer Program for Laboratory use in Economic Geography Courses.* Technical Paper No. 8. Association of American Geographers.
[62] R. S. Mathieson. 1969. The soviet contribution to regional science: A review article. *J. Region. Sci.* 9, 1 (1969), 125–140.
[63] Nimrod Megiddo and Kenneth J. Supowit. 1984. On the complexity of some common geometric location problems. *SIAM J. Comput.* 13, 1 (1984), 182–196.
[64] Jeremy Mennis and Diansheng Guo. 2009. Spatial data mining and geographic knowledge discovery-An introduction. *Comput. Environ. Urban Syst.* 33, 6 (2009), 403–408.
[65] Efrén Mezura-Montes and Carlos A. Coello Coello. 2011. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm Evol. Comput.* 1, 4 (2011), 173–194.
[66] Nenad Mladenović, Jack Brimberg, Pierre Hansen, and José A. Moreno-Pérez. 2007. The p-median problem: A survey of metaheuristic approaches. *Eur. J. Operat. Res.* 179, 3 (2007), 927–939.
[67] Pablo Moscato et al. 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech Concurrent Computation Program, C3P Report 826 (1989), 1989.
[68] Pablo Moscato and Carlos Cotta. 2010. *A Modern Introduction to Memetic Algorithms.* Springer US, Boston, MA, 141–183. https://doi.org/10.1007/978-1-4419-1665-5_6
[69] Stuart S. Nagel. 1964. Simplified bipartisan computer redistricting. *Stan. L. Rev.* 17 (1964), 863.
[70] Stan Openshaw and Liang Rao. 1995. Algorithms for reengineering 1991 census geography. *Environ. Plan. A* 27, 3 (1995), 425–446.
[71] Vitaly Osipov and Peter Sanders. 2010. N-level graph partitioning. In *Algorithms–ESA 2010*. Springer, Berlin, 278–289.
[72] Hae-Sang Park and Chi-Hyuck Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* 36, 2 (2009), 3336–3341.
[73] Daniel D. Polsby and Robert D. Popper. 1991. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law Pol. Rev.* 9, 2 (1991), 301–353.
[74] Charles S. ReVelle and Ralph W. Swain. 1970. Central facilities location. *Geogr. Anal.* 2, 1 (1970), 30–42.
[75] Sergio J. Rey and Luc Anselin. 2010. PySAL: A python library of spatial analytical methods. In *Handbook of Applied Spatial Analysis*. Springer, 175–193.
[76] Federica Ricca, Andrea Scozzari, and Bruno Simeone. 2011. Political districting: From classical models to recent approaches. *4OR* 9, 3 (2011), 223–254.
[77] Roger Z. Ríos-Mercado. 2020. *Research Trends in Optimization of Districting Systems.* Springer International Publishing, Cham, 3–8. https://doi.org/10.1007/978-3-030-34312-5_1
[78] Roger Z. Ríos-Mercado and Elena Fernández. 2009. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Comput. Operat. Res.* 36, 3 (2009), 755–776.

[79] María Angélica Salazar-Aguilar, Roger Z. Ríos-Mercado, and Mauricio Cabrera-Ríos. 2011. New models for commercial territory design. *Netw. Spat. Econ.* 11, 3 (2011), 487–507.

[80] Peter Sanders and Christian Schulz. 2011. Engineering multilevel graph partitioning algorithms. In *Algorithms–ESA 2011*, Camil Demetrescu and Magnús M. Halldórsson (Eds.). Springer, Berlin, 469–480.

[81] O. Benjamin Schoepfle and Richard L. Church. 1989. A fast, network-based, hybrid heuristic for the assignment of students to schools. *J. Operat. Res. Soc.* 40, 11 (1989), 1029–1040.

[82] Norbert Sensen. 2001. Lower bounds and exact algorithms for the graph partitioning problem using multicommodity flows. In *Algorithms–ESA 2001*, Friedhelm Meyer auf der Heide (Ed.). Springer, Berlin, 391–403.

[83] Takeshi Shirabe. 2009. Districting modeling with exact contiguity constraints. *Environ. Plan. B: Plan. Des.* 36, 6 (2009), 1053–1066.

[84] Horst D. Simon. 1991. Partitioning of unstructured problems for parallel processing. *Comput. Syst. Eng.* 2, 2–3 (1991), 135–148.

[85] Kenneth Sörensen and Fred Glover. 2013. Metaheuristics. *Encycl. Operat. Res. Manage. Sci.* 62 (2013), 960–970.

[86] Charles Sutcliffe, John Board, and Paul Cheshire. 1984. Goal programming and allocating children to secondary schools in reading. *J. Operat. Res. Soc.* 35, 8 (1984), 719–730.

[87] Michael B. Teitz and Polly Bart. 1968. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operat. Res.* 16, 5 (1968), 955–961.

[88] Daoqin Tong and Alan T. Murray. 2012. Spatial optimization in geography. *Ann. Assoc. Am. Geogr.* 102, 6 (2012), 1290–1309.

[89] Constantine Toregas and Charles ReVelle. 1972. Optimal location under time or distance constraints. In *Papers of the Regional Science Association*, Vol. 28. Springer, Springer International Publishing, 131–143.

[90] Chris Walshaw. 2004. Multilevel refinement for combinatorial optimisation problems. *Ann. Operat. Res.* 131, 1 (2004), 325–372.

[91] Chris Walshaw. 2008. *Multilevel Refinement for Combinatorial Optimisation: Boosting Metaheuristic Performance.* Springer, Berlin, 261–289. https://doi.org/10.1007/978-3-540-78295-7_9

[92] Roy D. Williams. 1991. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurr.: Pract. Exp.* 3, 5 (1991), 457–481.

[93] Justin C. Williams Jr. 1995. Political redistricting: A review. *Pap. Region. Sci.* 74, 1 (1995), 13–40.

[94] Ningchuan Xiao. 2008. A unified conceptual framework for geographical optimization using evolutionary algorithms. *Ann. Assoc. Am. Geogr.* 98, 4 (2008), 795–817.

[95] Ningchuan Xiao. 2015. *GIS Algorithms.* SAGE.

[96] KwangSoo Yang, Apurv Hirsh Shekhar, Dev Oliver, and Shashi Shekhar. 2013. Capacity-constrained network-voronoi diagram: A summary of results. In *Advances in Spatial and Temporal Databases*. Springer, Berlin, 56–73.

[97] KwangSoo Yang, Apurv Hirsh Shekhar, Dev Oliver, and Shashi Shekhar. 2015. Capacity-constrained network-voronoi diagram. *IEEE Trans. Knowl. Data Eng.* 27, 11 (2015), 2919–2932.

[98] Seda Yanık and Burcin Bozkaya. 2020. *A Review of Districting Problems in Health Care.* Springer International Publishing, Cham, 31–55. https://doi.org/10.1007/978-3-030-34312-5_3

[99] Jing Yao, Xiaoxiang Zhang, and Alan T. Murray. 2018. Spatial optimization for land-use allocation: Accounting for sustainability concerns. *Int. Region. Sci. Rev.* 41, 6 (2018), 579–600.

[100] Andris A. Zoltners and Prabhakant Sinha. 1983. Sales territory alignment: A review and model. *Manage. Sci.* 29, 11 (1983), 1237–1256.