Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4708135, 13 pages
https://doi.org/10.1155/2017/4708135

Research Article

A GRASP-Tabu Heuristic Approach to Territory Design for Pickup and Delivery Operations for Large-Scale Instances

Rosa G. González-Ramírez,[1] Neale R. Smith,[2] Ronald G. Askin,[3] José-Fernando Camacho-Vallejo,[4] and Jose Luis González-Velarde[2]

[1] *Faculty of Engineering and Applied Sciences, Universidad de Los Andes Chile, Monseñor Álvaro Portillo 12455, Las Condes, Santiago, Chile*
[2] *Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Eugenio Garza Sada 2501, 64849 Monterrey, NL, Mexico*
[3] *School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287-8809, USA*
[4] *Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Av. Universidad s/n, 66450 San Nicolás de los Garza, NL, Mexico*

Correspondence should be addressed to Neale R. Smith; nsmith@itesm.mx

Received 28 April 2017; Revised 28 September 2017; Accepted 2 October 2017; Published 29 October 2017

Academic Editor: Federica Caselli

We address a logistics districting problem faced by a parcel company whose operations consist of picking up and delivering packages over a service region. The districting process aims to find a partition of the service region into delivery and collection zones that may be served by a single vehicle that departs from a central depot. Criteria to be optimized are to balance workload content among the districts and to create districts of compact shape. A solution approach based on a hybrid procedure that combines elements of GRASP and Tabu Search (TS) is proposed to solve large-scale instances. Numerical experimentation is performed considering different instance sizes and types. Results show that the proposed solution approach is able to solve large-scale instances in reasonable computational times with good quality of the solutions obtained. To determine the quality of the solutions, results are compared with CPLEX solutions and with the current real solution to highlight the benefits of the proposed approach. Conclusions and recommendations for further research are provided.

1. Introduction

In this study, we consider a logistic districting problem faced by a parcel company that operates in the metropolitan area of Monterrey, Mexico. This company is concerned with the delivery and pickup of packages within Monterrey. The latter is divided into districts for logistic purposes. Each district is served by a single vehicle that departs from a central depot, in which the packages are received. At the end of the day, the vehicle comes back to the depot. Currently, drivers take the route decisions in a dynamic manner during the day due to the variability of the demand. That is, it is assumed that the driver may receive new dispatch orders along the day.

The company's current practice is to redesign the districting configuration every year and a half in order to account for demand variations. Although the day to day locations of the customers and the daily volume of demand are stochastic, the logistics planning managers consider that a reasonable approach is to use the data of a high workload day which is representative of the current and growing demand. The number of pickups and deliveries may vary, but, at the time, the company was considering data sets of about 1,100 customers, with about equal numbers of pickups and deliveries. In general, the days chosen by the company to serve as representative days have above-average demand and a geographical distribution of demand that is considered to be "typical," that is, not unusually high or low in any area with respect to an average behavior.

Once a representative day is selected, the logistics manager identifies the locations of the customers on a map. Then, the districts are readjusted considering their estimated capacity. But this readjustment needs to consider the compact shape of the district, and all the required arrangements to balance workload content among all the districts must be made. It takes about three weeks for the managers to complete that process.

The managers of the company are mainly interested in balancing the workload and in the shape of the districts. Accordingly, we define two criteria to optimize during the districting process: workload balance and compactness. The former is important because it tends to minimize the number of vehicles needed and, consequently, it reduces capital investment, opportunity costs, and possibly the vehicle maintenance costs. The latter criterion is important because it tends to reduce travel distances within a district.

We propose a mathematical model for the districting problem abovementioned. Following the assumptions considered by the logistic planning manager, we assume that a representative day can be properly chosen. This approach is reasonable because it has been working well for the company over years. So, we adopt that assumption and provide a tool to support and make a more efficient districting design. Also, we propose a heuristic solution methodology that can find good districting configurations in large-scale instances. Note that the use of a heuristic algorithm is justified by the difficulty of the problem, which has been shown to be NP-Complete by [1].

The remainder of this paper is organized as follows. Section 2 presents a literature review regarding districting problems. Section 3 presents the proposed mathematical model. Section 4 describes the heuristic algorithm and Section 5 presents the numerical results showing the suitability of the proposed algorithm to solve large-scale instances. Conclusions and recommendations for future research are given in Section 6.

2. Literature Review

Despite the fact that this literature review is devoted to districting problems that involve routing decisions, it is worthy to mention one of the pioneer papers in districting area. Keeney [2] addresses the problem of partitioning a service area into districts. Each district corresponds to an existing facility that provides service to that particular district. Later, researchers keep paying attention to this topic, solving districting problems in diverse contexts, such as politics, sales territory alignment, schools, health care systems, emergency services, and logistics (see [3–6]).

As it can be inferred from above, some of the applications of districting problems involve an interrelation with routing decisions. This fact has motivated the researchers to merge both decision processes in the same problem. One of the initial findings appears in [7], in which the method for vehicle routing proposed in [8] is compared against a methodology based on a districting approach. Wong and Beasley [9] considered the Vehicle Routing Using Fixed Delivery Areas (VRFDA), which is closely related to districting problems. In that problem, a service area is divided into fixed subareas and the daily route in each of them may change from day to day. The authors propose a methodology for minimizing the traveled distance. A special case of the VRFDA is the Fixed Routes Problem (FRP) studied in [10], in which the service region is divided into subareas but each route does not vary from day to day.

Daganzo [11] proposes an approximation method for the design of multiple-vehicle delivery zones seeking tours of minimum total length. The objective in that paper is to explore the impact that the zone shape has on the expected length of each route. Later, in [12], a methodology in which the region is partitioned into zones of nearly rectangular shape elongated toward the source is presented. The number of points considered in the instances is large compared to the capacity of the vehicles; this fact complicates the problem's resolution. Newell and Daganzo [13] analyze the districting of a region in which the underlying network of roads forms a dense ring-radial network. The authors proposed an approximation method for the design of multiple-vehicle delivery tours that minimizes the total traveled distance. The design of delivery zones for distributing perishable freight without transshipment is studied in [14]. Lei et al. [15] propose the multiple traveling salesperson and districting problem, which considers multiple periods and depots. The authors propose an adaptive large neighborhood search metaheuristic.

Regarding problems for designing delivery tours within districts, it can be mentioned [16], in which the design of multiple-vehicle delivery tours that satisfy time constraints for letter and parcel pickup is studied. The authors propose a methodology that involves partitioning the region into approximately rectangular delivery zones that are arranged into concentric rings around the depot using a continuous approximation of the model. Rosenfield et al. [17] study the problem of planning service districts with a time constraint. One of their main contributions is the derivation of analytical expressions to determine the optimal number of service districts for the US postal system. A methodology to design multidelivery tours associated with the servicing of an urban region of irregular shape is presented in [15]. The authors' methodology is based on a sweep approach and assumes a rectangular grid structure. Novaes et al. [18] present a methodology for solving the same problem as [19], but they used a continuous approach to represent the region. Previous works are extended by [20] introducing some improvements to the ring-radial model. The authors present a special case of a Voronoi diagram and model the problem with a continuous approximation.

Muyldermans et al. [21, 22] address the problem of districting for salt spreading operations on roads. They assume that each district is served by a single facility aiming to minimize the deadheading distances and the number of vehicles required to service the region. The authors present a novel model based on a graph with demand at the nodes. Another districting problem modelled on a graph but considering the demand in the arcs is presented in [23], in which the concept of Eulerian districts is introduced.

Haugland et al. [24] consider the problem of designing districts for vehicle routing problems with stochastic demands. They presented a Tabu Search (TS) and multistart heuristics to solve the problem. Also, a multiobjective dynamic stochastic districting and routing problem is considered in [15]. In this problem, the customers of a territory stochastically evolve over the planning periods. Therefore, several objectives must be optimized via a coevolutionary algorithm. Another nondeterministic districting problem is studied in Sheu [25], in which an integrated fuzzy-optimization customer grouping based logistics distribution methodology is presented. The proposed method groups customers' orders primarily based on the multiple attributes of customer demands. Then, the customer group-based delivery service priority is determined, and finally the routes for deliveries are established.

Concerning multiple objectives to be considered in a districting model, Tavares-Pereira et al. [26] consider a districting problem with multiple criteria and propose an evolutionary algorithm with local search to approximate the Pareto front. Also, a problem related to a bottled beverage distribution company is studied in [27]. Hence, a biobjective programming model is introduced considering dispersion and balancing with respect to the number of customers in each district as performance criteria. Two continuous location-districting models applied to transportation and logistics problems are developed in [28]. A Voronoi diagram is combined with an optimization algorithm for solving both proposed models.

In [29, 30], the same districting design problem for the operations of a parcel company is studied. In both references, the same mathematical model is proposed but in [30] a two-stage approach is considered. The second stage is related to a mathematical model that considers the solution of the first stage and, then, aims to minimize the dispersion of the workload in the districts. Small and medium size instances are solved via a hybrid algorithm. That algorithm is unable to efficiently solve large size instances. In [31], there is a review of the state of the art regarding modelling techniques and solution methods for problems in supply chain planning that handle districting or customer clustering.

The problem herein addressed is closely related to [29, 30]. However, there are significant differences among them. Particularly, in this paper we are proposing an adjusted heuristic that is capable of solving large-scale instances with low computational effort. In previous papers, only limited size instances are solved, but herein the proposed heuristic efficiently solves larger instances. The main differences between the algorithms is that the procedure for constructing feasible solutions, the neighborhoods considered, and the movements explored during the local search are modified. The latter changes allow the heuristic to solve large size instances. Moreover, instances with different structure are tested, and the heuristic algorithm shows to be robust. Specifically, a set of symmetric instances is considered in which the optimal solution is known due to their particular structure. These instances allow properly measuring the efficiency of the proposed heuristic.

Furthermore, other important characteristics of this research are that we aim to optimize the balance of workload content and compactness in a single objective function instead of a single objective (as usually). Also, parcel applications are scarce in the literature. Hence, this research is worthy due to parcel companies handling many customers and the proposed heuristic can efficiently solve these situations. The modeling structure proposed in this work also differs from others found in the literature, mainly in the fact that the districting problem is modeled as a graph in which demand occurs at the nodes.

In our situation under study, districting is a strategic decision that is not defined day to day and demand points are not fixed. We also distinguish between pickup and delivery operations. The only reviewed paper to do so is [16], but the objective is to minimize the expected length of the tours and the workload balance is addressed as a constraint. So, it differs from the model presented in this paper, in which we aim to balance workload content and also achieve districts of compact shape.

3. Mathematical Formulation

Consider a connected, undirected graph $G(V, E)$, where V is the set of vertices and E is the set of edges. In general, the graph is not complete. All the edges $e_{ik} = (v_i, v_k)$, $e_{ik} \in E$, $(v_i, v_k) \in V \times V$, have a positive length and represent a real path between adjacent vertices v_i and v_k. Distances between vertices are edge lengths for adjacent pairs and shortest path distances for nonadjacent pairs. A district is defined as a subset of the vertices and $J = \{1, \dots, m\}$ is the district set. Each vertex represents a customer that may require either a pickup or a delivery. The depot is defined as the vertex $\{0\}$.

We define wp_i and wd_i as the number of pickups/deliveries requested by each demand point i, $i \in V$; and Stp_i and Std_i as the stopping time per pickups/delivery in each demand point i, $i \in V$. As mentioned previously, the distance between a pair of demand points is denoted as d_{ik}. Sp represents the average vehicle speed. W and Z are continuous variables that represent the maximum workload content and compactness metrics, respectively. D_j is a continuous auxiliary variable that takes the value of the travel time from the depot to the farthest point in district j, $j \in J$.

The districting procedure seeks to optimize two criteria simultaneously: balancing workload among the districts and compactness of their shapes. Compactness is not defined uniquely for all the districting problems in the literature and it is generally defined according to the application context. For this problem, we define it as the distance between the two furthest apart vertices in a district, that is, $\max\{d_{ik}\}$, considering all vertices v_i and v_k belonging to a district. We employ a minimax objective in which the maximum compactness metric among the districts is minimized. Although the compactness metric that we employ could have the same value for districts of different shape (e.g., a long thin district), when combined with the objective to balance workload content, in practice it produces districts of approximately circular or square shape.

For each district, the workload content is defined as the time required to perform all required pickups and deliveries and also includes an estimation of the time required to travel from the depot to a vertex in the district. The latter is approximated by the time required to travel from the depot to the farthest point in the district. Despite the fact that the closest vertex in the district or the centroid could also be employed to approximate the line haul distance from the depot, considering that the vehicle will visit all vertices in a district during the route, the farthest vertex is a reasonable metric. Additionally, the use of the farthest vertex simplifies the mathematical model. In order to balance the workload content among districts, the maximum workload among all districts is minimized. The following expression shows how the workload of a district is calculated:

$$\sum_{i\in V} Std_i \cdot wd_i \cdot X_{ij} + \sum_{i\in V} Stp_i \cdot wp_i \cdot X_{ij} + \frac{D_j}{Sp}. \quad (1)$$

We propose a linear single objective mixed integer model in which a weighted sum of the compactness metric and the maximum workload content assigned to a district is minimized. Each metric is normalized with respect to an estimation of the optimal values of compactness and workload content (Nz and Nw, resp.).

Since a feasible number of vehicles considered in the model are known a priori, the capacity of the vehicles is not explicitly considered in the model. To ensure that a feasible solution with respect to vehicle capacity is found, we balance district workloads and we keep the number of pickups and deliveries within certain limits. These limits will not allow pickups and deliveries to be intermixed, and this will release vehicle capacity by deliveries to allow subsequent pickups. The limits on the number of pickups and deliveries for each district are denoted by α and β, respectively.

The following decision variables are defined:

$$X_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is assigned to district } j, \ j \in J \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The mathematical model is as follows:

$$\min \quad \frac{\lambda W}{Nw} + \frac{(1-\lambda) Z}{Nz} \quad (3)$$

$$\text{subject to} \quad \sum_{j\in J} X_{ij} = 1 \quad \forall i \in V \quad (4)$$

$$\sum_{i\in V} wp_i \cdot X_{ij} \le \alpha \quad \forall j \in J \quad (5)$$

$$\sum_{i\in V} wd_i \cdot X_{ij} \le \beta \quad \forall j \in J \quad (6)$$

$$D_j \ge d_{i0} \cdot X_{ij} \quad \forall j \in J,\ i \in V \quad (7)$$

$$Z \ge \frac{d_{ik}\left(X_{ij} + X_{kj} - 1\right)}{Sp} \quad \forall j \in J,\ i \in V,\ k \ne i \in V \quad (8)$$

$$W \ge \sum_{i\in V} Std_i \cdot wd_i \cdot X_{ij} + \sum_{i\in V} Stp_i \cdot wp_i \cdot X_{ij} + \frac{D_j}{Sp} \quad \forall j \in J \quad (9)$$

$$X_{ij} \in \{1,0\}; \quad D_j \ge 0; \quad W \ge 0; \quad Z \ge 0; \quad \forall j \in J,\ i \in V. \quad (10)$$

Equation (3) is the objective function that minimizes a weighted average of the maximum workload and compactness. Both terms in (3) are normalized and the relative weighting is given by λ, $0 \le \lambda \le 1$. Normalization parameters are estimated based on their optimal solution values as follows: Nw is determined by estimating an average workload per district when it is approximately balanced by dividing the total workload of the region. Nz is estimated as the traveling time between the two furthest points in a district, assuming that the service region is equally divided into m districts of equal shape (assuming a circular shape for normalization parameters estimation). Constraints (4) guarantee that each demand point is assigned to exactly one district. Constraints (5) and (6) ensure that each district has a maximum number of α pickups and β deliveries, respectively. Constraints (7) guarantee that D_j takes the value of the time from the depot to the farthest vertex of each district j. Constraints (8) require that Z take the value of the maximum travel time between the vertices assigned to a district. Constraints (9) guarantee that W takes the value of the maximum workload from among the districts. Constraints (10) are the binary and nonnegativity requirements.

4. Solution Methodology

The redistricting problem has been shown to be NP-Complete by Altman (1997). Moreover, since constraints (5) and (6) impose a limit in the amount of pickups and deliveries assigned to a district, it turns out that even finding a feasible solution is a difficult task. Motivated by the difficulty of the problem, we propose a multistart heuristic that hybridizes GRASP and Tabu Search. The procedure seeks a feasible solution (partition of the customers into districts) of the model proposed in Section 3 aiming to obtain good partition quality in terms of the values of the objective function in which the weighted sum of compactness and work balance metrics are optimized. We will further describe the steps of the algorithm.

The general procedure consists of two phases: construction of a feasible initial partition and improvement by a local search. These phases are named as FIP-Construction and Local-Search, respectively. In the first phase, districts are conformed by partitioning all the customers into the districts in set J. Hence, a solution of the proposed hybrid algorithm corresponds to a districting decision. The Local-Search procedure incorporates a Tabu Search (TS) short term memory, in which the recently visited partitions are labeled as Tabu active during a predefined number of iterations. The aspiration criterion allows a Tabu active move only if the resulting partition is better than the current best one. Among

```
Hybrid GRASP-TS Procedure
Input: $P :=$ instance of the problem $(G, d_{ik}, wp_i, wd_i, Sp, \lambda, Stp_i, Std_i)$
Output: $x^* = (x_{11}, x_{12}, \ldots, x_{1m}, \ldots, x_{|V|,m}) :=$ The best partition of graph $G$,
or empty set with no feasible partition.
Set $f^* = \infty, x^* = \{\phi\}$;
 Do until a stopping condition is met
     For $n = 1$ to $NSeeds$ do
         $x \leftarrow$ FIP-Construction ($n$);
         $x \leftarrow$ Local-Search ();
         If $f(x) < f^*$ then
                 $f^* = f(x)$ and $x^* = x$;
     End for
 End do
Return $x^*$
```

Algorithm 1: Pseudocode of the solution procedure.

all the partitions created and improved, the overall best one is reported as the final output of the algorithm.

We propose two local search procedures that may be applied independently or in two different combinations. This results in four strategies that attempt to improve the current partition (set of districts) constructed during each iteration of the algorithm. A key concept is the *adjacency* of vertices and districts. A vertex is considered to be adjacent to a district if there exists at least one edge connecting the vertex with another one that already belongs to the district. Then, we restrict the allocation of the vertices only to its adjacent districts. Knowledge of the adjacency helps to avoid unnecessary evaluations that may result in long computational times in the local search and also to enhance compactness of the constructed partition.

The encoding of a partition x is as follows: there is a list associated with each district, in which the index of the customers associated with it are included. Moreover, the quality (f) of each partition x will be measured as the weighted sum of both objectives considered in (3). Algorithm 1 shows the pseudocode of the procedure in which f^* represents the best value of the objective function given by (3) and x^* is the corresponding best partition found. The term *seed* will be used hereafter to denote a vertex chosen to initiate the construction of a district. *NSeeds* is the number of seed selection procedures used in the FIP-Construction method. We employ five different procedures to select a set of seeds. The procedures are used in sequential order, $n = 1, \ldots, NSeeds$, during the iterations of the hybrid algorithm. In each iteration a feasible partition is attempted to be constructed and, if found, improved.

4.1. Phase I: Feasible Initial Partition Construction (FIP-Construction)

We apply a multistart approach, in which a determined number of initial solutions are attempted to be constructed based on a greedy randomized approach. For the construction of m districts, the same number of seeds would be chosen. Algorithm 2 shows the pseudocode of the construction procedure, for which the two main steps are selection of a set of m seeds (Seeds_Selection) which employs a single seed selection procedure in each iteration (n) and allocation of vertices to the districts associated with each seed (Vertices_Allocation). Set Ψ contains the unassigned vertices. To enhance compactness, vertices are attempted to be assigned to the closest seed as long as adjacency conditions are fulfilled. If an initial feasible partition has been constructed, a local search procedure is applied. Otherwise the partition is discarded.

4.1.1. Seeds Selection

Five different approaches are used in this step ($n = 1, \ldots, NSeeds$). Each time the algorithm performs an iteration to construct and improve a feasible partition, the five approaches are sequentially employed. Algorithm 3 shows the pseudocode of the general procedure, in which $\overline{V}$ is the set of potential vertices from which seeds, $s_j \in Seed$, are selected. $G(n)$ is the greedy function employed in each procedure. The greedy and semirandom functions are further described.

P-Dispersion Greedy Function. The objective is to select the seeds that are most dispersed relative to each other. The greedy function computes the sum of the distances between the already selected seeds and a potential vertex. Vertices associated with larger sums are considered to be more desirable.

Neighborhood Greedy Function. Vertices are evaluated according to the number of vertices that are located within a *neighborhood* defined by a threshold distance. The vertices with more *neighbors* are considered to be better. Once a seed is selected, the neighbors of the seed (adjacent vertices) are discarded as potential candidates.

Angle Greedy Function. Based on the location of the vertices. The service region is partitioned into m wedge-shaped radial sectors of equal size. The RCL is formed by the vertices whose locations are closest to the boundary between two sectors.

Workload Greedy Function. It is similar to previous method but the region is divided into sectors of approximately equal workload content, measured by the number of vertices on it.

```
FIP-Construction ($n$)
Input: $P \coloneqq$ instance of the problem $(G, (x_i, y_i), d_{ik}, wp_i, wd_i, Sp, \lambda, Stp_i, Std_i)$
Output: $x = (x_{11}, x_{12}, \ldots, x_{1m}, \ldots, x_{|V|,m}) \coloneqq$ A feasible partition of graph $G$,
or empty set if no feasible partition was constructed
Set $x = \varnothing$; set $Seed = \varnothing$, and set $i \in \Psi, \forall i \in V$
  $x \leftarrow$ Seeds_Selection ($n$)
  $x \leftarrow$ Vertices_Allocation ()
If $x$ is feasible, return $x$
Else return ()
```

Algorithm 2: Pseudocode of the construction procedure.

```
Seeds_Selection ($n$)
Input: $P \coloneqq$ instance of the problem $(G, d_{ik}, wp_i, wd_i, Sp, \lambda, Stp_i, Std_i)$
Output: Set $Seed$.
Set $\overline{V} = V$
For $j = 1$ to $m$ do
    If $\overline{V} \neq \varnothing$
        If $n = 1, \ldots, NSeeds - 1$
            $\overline{V} \leftarrow$ Greedy_Evaluation ($n$)
            RCL $\leftarrow$ {$k$ best elements}
            Randomly select $s_j \in Seed$ among the elements of the RCL
            For the Neighborhood greedy function, discard from $\overline{V}$ the neighbors of $s_j$
        Else, $\overline{V} \leftarrow$ Semi-Random procedure, (neighbors of $s_j$ are discarded from $\overline{V}$)
    Else, randomly select $s_j \in Seed$ from among the discarded vertices
End
Return {$Seed$}
```

Algorithm 3: Pseudocode of the greedy randomized seed selection procedures.

Semirandom Selection. A vertex is randomly selected as a seed at each iteration and the adjacent vertices are discarded as potential seeds. If all the points have been eliminated before all seeds are selected, the remaining seeds are selected randomly.

4.1.2. Vertices Allocation. Once the set of seeds has been selected, districts are formed around the seeds. This procedure not only attempts to create a feasible partition but also strives for compactness by assigning vertices to their closest seed subject to satisfying adjacency requirements. Vertices are allocated to districts in three consecutive steps. In each step, we attempt to construct a feasible partition, and the next step is applied only if a feasible partition was not found in the previous step.

The procedure is as follows: (1) a seed, s^*, is randomly selected and the N closest vertices to the seed are inspected with the aim of assigning the closest adjacent vertex that satisfies the capacity of the district. N is a control parameter and its value is determined based on preliminary experimentation (see Section 5.2). The procedure is repeated until a stopping condition is met. (2) The remaining unassigned vertices are attempted to be assigned to an adjacent district, respecting capacity, and each vertex is explored only once. (3) The remaining unassigned vertices are attempted to be assigned to an adjacent districting, even if capacity is violated, but always aiming to assign the vertex to the district that least increases its number of pickups or deliveries beyond its capacity limits. If an infeasible assignment of vertices is generated, a last procedure is applied based on local search (exchange of vertices between adjacent districts). The aim of the latter is to find a feasible partition; in case that this is not possible, the partition is discarded.

4.2. Phase II: Local Search Phase (LS). After a feasible partition has been constructed, its quality can be measured by computing the corresponding objective function value according to (3). Hence, the improvement of the current partition is sought. The LS phase contains four neighborhood structures, two of them are referred to as *"1-Step LS"* (1-S) and *"K-Steps/Pairs LS"* (K-S/P). Two additional neighborhood structures result from a combination of the above-mentioned neighborhood structures: *"Hyperheuristic LS"* (HypLS) and *"2-Iterations LS"* (2-IterLS). In an iteration of the LS phase, the evaluation of the objective function of all the partitions over a search space determined by one of the four neighborhood structures proposed is performed. In case of ties, we select the partition that presents the least dispersion of the workload content among the districts.

Each neighborhood structure implements a Tabu Search short term memory: recently visited solutions are labeled as *Tabu active* during t iterations. Tabu permanence t was defined as a static value. The aspiration criterion allows a Tabu active move only if the resulting partition is better than the current best one. Local search procedure is applied until stopping conditions are met.

For this phase, to simplify the exploration, a partition x is codified as the matrix x_{ij}, with $x_{ij} = 1$ if vertex i is assigned to district j and $x_{ij} = 0$ otherwise. A movement in partition x, consists in changing the vertex i adjacent to district j', that is, from $x_{ij} = 0$ and $x_{ij'} = 1$, for all i and $j \neq j' \in J$. An exchange neighborhood of partition x consists of the set of partitions that, starting with $x_{ij} = 1$ and $x_{lj'} = 1$, with point i adjacent to district j' and point i' adjacent to district j, results from setting $x_{ij} = 0$, $x_{ij'} = 1$, $x_{lj'} = 0$, and $x_{lj} = 1$ for all $i, l \in V$and $j \neq j' \in J$. The neighborhood structures are the following.

1-Step LS (1-S), $N1(x)$. Each iteration is concluded by performing the best move found and setting the partition with $x_{ij} = 1$ as Tabu active during t iterations. A list of the three best partitions is also maintained so that a final attempt is made to improve the three best ones found in hopes of finding a better partition.

k-Steps/Pair LS (k-S/P), $N2(x)$. This neighborhood is an extension of the algorithm proposed by Kernighan and Lin [32] for graph partitioning. It consists of the set of partitions that results during a number of steps, $k = 1, \ldots, K$. In each step k, all moves and exchanges are explored for a pair of adjacent districts which are randomly selected. Partitions with $x_{ij} = 1$ and/or $x_{lj'} = 1$ are set as Tabu active during t iterations. During each step k, the best partition, x^k, is selected, and the search is repeated starting with that partition x^k. Each iteration is concluded by selecting the best overall partition among $x^1, \ldots, x^K$. The suggested value of K is $\lfloor n/2 \rfloor$, where n is the maximum number of vertices allocated to one of the two districts under consideration. To enhance diversity, probabilities of the districts being selected are adjusted during the iterations in order to enhance the search over spaces that have not been considered.

Hyperheuristic-LS (HypLS). Consider a codified partition x, where $x_{ij} = 1$ if vertex i is assigned to district j. The neighborhood of x is a combination of $N1(x)$ and $N2(x)$ neighborhoods. The procedure randomly selects one of the two neighborhood structures and search over the space defined by the corresponding neighborhood. This procedure is repeated until a stopping condition is met, also maintaining a list with the three best partitions to perform a final attempt to improve them based on $N1(x)$.

2-Iterations LS (2-IterLS). Consider a partition x, where $x_{ij} = 1$ if vertex i is assigned to district j. The neighborhood structure is the union of the $N1(x)$ and $N2(x)$ neighborhoods. Given a current partition, the procedure consists of evaluating all the possible ones according to each search, selecting the best partition over all the space. This procedure is repeated a number of iterations until a stopping condition is met. The three best partitions are also maintained so that a final attempt to improve them is done based on $N1(x)$.

The best overall partition is reported as the output of the proposed algorithm. Notice that this corresponds to a feasible partition of the model proposed in Section 3; that is, it is a districting configuration that satisfies all the constraints and with best weighted sum of workload balancing and compactness metrics.

5. Numerical Experimentation

Numerical experimentation is performed to test the performance of all the procedures described above. The proposed algorithm is implemented in C programming language. We generate a vast set of instances of different types and sizes. For comparison purposes, the smallest size instances are solved with CPLEX 11.1.

5.1. Instance Generation. We generate several types of instances in order to test the different components of the proposed algorithm under different conditions. The characteristics of each type of instance are summarized in Table 1. Instance sizes considered are small instances (50 vertices/5 districts and 200 vertices/10 districts), medium instances (450 vertices/15 districts), and large instances (1000 vertices/20 districts and 1500 vertices/30 districts). The Parcel instances have 1109 vertices and 28 districts. For all the instance types, the time required to perform a pickup and a delivery was set to a value of 10 and 5 minutes, respectively. For the test instances, the travel time to an adjacent vertex is assumed to be included in these values.

To generate the first three types of instances, the limits on the number of pickups and deliveries were defined based on two levels of capacity: tight (T) and less restricted (LR). The relative weighting factor varies over three values: $\lambda = 0.25$, 0.5, and 0.75. The speed values vary over three values: 25, 30, and 35 kilometers/hour. Three replicates for each of the five instance sizes were generated. For each replicate, additional instances were generated by varying three factors: the value of the relative weighting factor, the two levels of capacity limits, and the three values of speed, resulting in a total of $3 \times 2 \times 3 \times 3 \times 3 \times 5 = 810$ instances.

To generate the urban type instances, we define the location of the vertices over the entire metropolitan region of Monterrey by generating a set of what we call *base vertices*, from which vertices are randomly selected to form each problem instance, for a large size instances (2211 vertices) and medium and small instances (1185 vertices). For each of the five instance sizes, three replicates were generated, and each of them was tested varying over the two capacity limits, the three values of the relative weighting factor, and the three values of speed, resulting in a total of $5 \times 3 \times (2 \times 3 \times 3) = 270$ instances.

Parcel instances were generated considering GPS data provided by the parcel company that was collected during

Table 1: Instance characteristics.

Type	Characteristics
Symmetric	Unrealistic instance but used to measure the performance of the heuristic for larger instances. Optimal solutions consist of "symmetric" districts: districts with the same workload content and the same shape. To generate the instance, a district is created and rotated to form the rest of the districts symmetrically. Workload content is also equal for all the districts. Euclidean distances are computed even if the points are not directly connected to guarantee symmetry.
Semisymmetric	Very similar to the symmetric instance, but Euclidean distances are computed only for those points connected by an edge. For the rest a shortest path distance is computed, and, hence, there is no guarantee that an optimal solution corresponds to a symmetric configuration.
Asymmetric	More general instances consist of vertices uniformly distributed over a plane using Euclidean distances for those vertices connected by an edge and shortest path distances for the rest.
Urban	Instances that consist of vertices distributed over a region that resembles the structure of the metropolitan area of Monterrey, Mexico.
Parcel	Instances generated with GPS data of a representative workday provided by the parcel company. This data set is a full size real instance.

Table 2: Asymmetric, Semi-Symmetric and Urban instances results.

Size	Metric	Asymmetric				Semisymmetric				Urban			
		K-S/P	1-S	HypLS	2-IterLS	K-S/P	1-S	HypLS	2-IterLS	K-S/P	1-S	HypLS	2-IterLS
50_5	Max	13.27%	8.73%	13.93%	12.69%	0.00%	0.00%	0.00%	0.00%	12.91%	0.02%	4.57%	0.00%
	Avg	4.00%	1.59%	2.09%	1.20%	0.00%	0.00%	0.00%	0.00%	1.64%	0.00%	0.57%	0.00%
	Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
200_10	Max	−19.28%	−19.79%	−20.08%	−20.06%	5.70%	0.00%	0.00%	0.00%	−24.10%	−23.14%	−28.32%	−28.79%
	Avg	−37.21%	−40.21%	−40.34%	−40.77%	−42.50%	−45.54%	−46.05%	−46.12%	−45.04%	−49.33%	−50.45%	−51.68%
	Min	−53.63%	−55.54%	−55.23%	−54.54%	−56.35%	−59.87%	−59.87%	−59.87%	−65.55%	−68.16%	−69.89%	−72.16%

some representative days. The instance has a size of 1109 vertices and 28 districts. Each edge represents a real distance between a pair of adjacent vertices. The adjacency is determined respecting the real street structure of the city. Workload content (pickup or delivery) is randomly assigned to each vertex, and a single replicate is tested with each of the two capacity limits defined and over the three values of lambda and the average speed, resulting in total $2 \times 3 \times 3 = 18$ instances. In total 810 + 270 + 18 = 1,098 instances are solved.

5.2. Stopping Rules and Control Parameters. A limit time of 7200 seconds is set for CPLEX. For the heuristic and all the neighborhood structures explored we set a limit time of 3600 seconds. An iteration of the heuristic is allowed to proceed as long as it begins before the 3600-second time limit. The last iteration itself may take a variable amount of time to complete. Based on preliminary experimentation we define a maximum number of iterations for the FIS-Construction and for the local search procedures; the value of the Tabu permanence is set to four iterations, and the size of the RCL is set to a value of three.

5.3. Results and Discussion. For the symmetric instances, the symmetric optimal solution is compared to the solution found by the heuristic and CPLEX. For the rest of the instances, a gap with respect to the best integer solution reported by CPLEX is computed as described by (11). Positive gaps are obtained when CPLEX finds better solutions:

$$\text{Gap} = \frac{\text{Heur\_sol} - \text{CPLEX\_sol}}{\text{CPLEX\_sol}}\%. \tag{11}$$

Tables 2–8 present results summarized either by type and instance size or by weighting factor. The tables present the maximum, average, and minimum value over the instances that were solved. Table 2 shows the gaps with respect to CPLEX results for the asymmetric, semisymmetric, and urban instances, considering those instance sizes in which CPLEX was able to find at least an integer solution (50_50 and 200_10). CPLEX found the optimal solution for the smallest size instances of 50_5. For the semisymmetric instances, all the tested solution methods found the optimal solution. For the asymmetric instances of 50_5, the 1-S method found the smallest of the maximum gaps (8.7%) but the 2IterLS found the best average gap (1.20%). For the urban instances of 50_5, the 2-IterLS method found the optimal solutions. Good results are observed in general with a maximum gap of less than 14%.

Table 3 shows the results for the asymmetric and urban instances by weighting factor and speed factors, for the instance size of 50_5, which CPLEX solved to optimality. We do not present results of the semisymmetric instances in the table because, for all the instances, all the tested solution procedures found the optimal solution.

As can be observed, the results do not vary too much according to the values of both factors and no significant pattern in the performance of the heuristics was found with respect to the values of the factors. Observe that, regarding

Table 3: Asymmetric and Urban 50_5 results by weighting factor and speed factors.

Param.	Values	Metric	K-S/P	1-S	HypLS	2-IterLS	K-S/P	1-S	HypLS	2-IterLS
			Asymmetric				Urban			
Speed	25	Max	8.26%	7.82%	13.93%	3.24%	12.91%	0.01%	4.56%	0.00%
		Avg	3.89%	1.64%	2.27%	0.67%	1.62%	0.00%	0.57%	0.00%
		Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	30	Max	13.27%	8.73%	13.83%	5.49%	12.91%	0.02%	4.56%	0.00%
		Avg	3.94%	1.68%	2.31%	1.16%	1.62%	0.00%	0.57%	0.00%
		Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	35	Max	12.70%	8.25%	10.37%	12.69%	12.91%	0.02%	4.56%	0.00%
		Avg	4.18%	1.44%	1.69%	1.77%	1.67%	0.00%	0.57%	0.00%
		Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
λ	0.25	Max	5.11%	1.65%	2.69%	1.45%	6.59%	0.02%	2.34%	0.00%
		Avg	5.11%	1.65%	2.69%	1.45%	1.45%	0.00%	0.51%	0.00%
		Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	0.50	Max	10.41%	8.73%	10.78%	9.17%	12.91%	0.01%	4.56%	0.00%
		Avg	3.98%	1.78%	2.01%	1.18%	2.82%	0.00%	0.99%	0.00%
		Min	0.10%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	0.75	Max	6.24%	5.68%	5.28%	5.12%	2.68%	0.02%	0.96%	0.00%
		Avg	2.92%	1.33%	1.58%	0.98%	0.64%	0.00%	0.21%	0.00%
		Min	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 4: Asymmetric, semisymmetric, and urban results by capacity level.

SIZE	50_5						200_10					
Capacity	Less restricted			Tight			Less restricted			Tight		
Gap	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min
Urban												
K-S/P	2.88%	0.78%	0.00%	12.91%	2.49%	0.00%	−24.70%	−45.66%	−65.55%	−24.10%	−44.42%	−65.10%
1-S	0.02%	0.00%	0.00%	0.00%	0.00%	0.00%	−30.75%	−50.24%	−68.16%	−23.14%	−48.42%	−67.78%
HypLS	1.17%	0.27%	0.00%	4.56%	0.87%	0.00%	−30.61%	−50.37%	−69.70%	−28.32%	−50.53%	−69.89%
2-IterLS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	−31.56%	−51.34%	−70.57%	−28.79%	−52.01%	−72.16%
Semisymmetric												
K-S/P	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	5.70%	−40.64%	−55.67%	−28.31%	−44.36%	−56.35%
1-S	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	−44.07%	−59.87%	−30.76%	−47.01%	−59.48%
HypLS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	−44.37%	−59.87%	−30.76%	−47.73%	−59.67%
2-IterLS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	−44.50%	−59.87%	−30.76%	−47.73%	−59.67%
Symmetric												
K-S/P	8.26%	3.11%	3.11%	13.27%	4.90%	4.90%	−19.28%	−37.45%	−53.63%	−21.66%	−36.98%	−50.01%
1-S	3.06%	0.92%	0.92%	8.73%	2.25%	2.25%	−19.79%	−39.35%	−54.47%	−25.30%	−41.06%	−55.54%
HypLS	4.56%	0.90%	0.90%	13.93%	3.28%	3.28%	−20.08%	−39.93%	−54.45%	−23.63%	−40.75%	−55.23%
2-IterLS	1.40%	0.32%	0.32%	12.69%	2.09%	2.09%	−20.06%	−40.26%	−54.54%	−24.44%	−41.29%	−53.78%

the weighting factor, in general, maximum gaps were worse for an equal weighting factor for both criteria (0.5), but that, for the average gaps, some heuristics performed better with a low value of lambda (0.25) and others with a high value of lambda (0.75).

Table 4 presents results for the same instance sizes and types, but with respect to the capacity level. As can be observed, the tight instances were more difficult for the procedures, and it was also more difficult to find feasible solutions for this instance type.

For the symmetric instances, the optimal solution (the symmetric solution) corresponds to symmetric districts with the same workload content and shape. Gaps are estimated similarly as in (11) but now considering the value of the optimal solution instead of the solution found by CPLEX. Table 5 reports the results found based on the instance size.

Table 5: Symmetric instances results with respect to the optimal solution.

SIZE	GAP (%)	SYMMETRIC				
		CPLEX	K-S/P	1-S	HypLS	2-IterLS
50_5	Max	0.00%	20.22%	0.00%	0.00%	0.00%
	Avg	0.00%	3.34%	0.00%	0.00%	0.00%
	Min	0.00%	0.00%	0.00%	0.00%	0.00%
200_10	Max	155.34%	37.78%	22.33%	23.08%	22.33%
	Avg	16.31%	17.32%	4.86%	8.68%	2.93%
	Min	0.00%	8.05%	0.00%	0.00%	0.00%
450_15	Max		28.14%	22.62%	26.89%	26.16%
	Avg		12.70%	7.79%	11.44%	9.24%
	Min		2.80%	0.00%	0.00%	0.00%
1000_20	Max		38.08%	18.81%	23.96%	24.02%
	Avg		12.54%	7.17%	9.08%	7.87%
	Min		4.10%	2.87%	1.74%	1.74%
1500_30	Max		34.57%	33.69%	34.57%	33.69%
	Avg		10.38%	13.77%	12.85%	8.86%
	Min		0.00%	0.00%	0.00%	0.00%

Table 6: Symmetric instances results by capacity level.

Size	50_5						200_10					
Capacity	Less restricted			Tight			Less restricted			Tight		
Gap	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min
CPLEX	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	155.34%	96.78%	54.81%	145.40%	89.20%	37.48%
K-S/P	20.22%	6.67%	0.00%	0.00%	0.00%	0.00%	37.78%	18.33%	9.36%	26.72%	16.30%	8.05%
1-S	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	22.33%	4.86%	0.00%	22.33%	4.86%	0.00%
HypLS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	23.08%	9.76%	2.82%	23.08%	7.60%	0.00%
2-IterLS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.77%	0.65%	0.00%	22.33%	5.22%	0.00%
Size	450_15						1000_20					
Capacity	Less restricted			Tight			Less restricted			Tight		
Gap	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min
K-S/P	27.61%	13.55%	4.15%	28.14%	11.85%	2.80%	31.19%	12.03%	4.91%	38.08%	13.04%	4.10%
1-S	20.95%	5.10%	0.00%	22.62%	10.49%	3.00%	11.72%	6.61%	2.86%	18.81%	7.72%	2.86%
HypLS	26.16%	10.78%	0.00%	26.89%	12.10%	1.35%	23.54%	9.43%	4.56%	23.96%	8.73%	1.74%
2-IterLS	26.16%	9.14%	0.00%	22.62%	9.33%	0.49%	16.27%	6.79%	1.74%	24.02%	8.94%	3.29%
Size	1500_30											
Capacity	Less restricted			Tight								
Gap	Max	Avg	Min	Max	Avg	Min						
K-S/P	34.57%	9.85%	0.00%	34.57%	10.90%	0.00%						
1-S	33.69%	14.40%	0.00%	33.25%	13.14%	0.00%						
HypLS	34.57%	11.97%	0.00%	34.13%	11.27%	0.00%						
2-IterLS	33.69%	9.39%	0.00%	26.89%	8.33%	0.00%						

We estimated gaps between CPLEX and the heuristics with respect to the optimal solution. CPLEX found the optimal solution only for the instances of 50 vertices. The positive gaps reported for CPLEX for the instances of 200 vertices indicate that it did not find an optimal solution within the time limit.

Table 6 presents results for the symmetric instances based on the capacity level. Observe that the K-S/P procedure did not perform as well on 50_5 less restricted instances than on the tight instances. Also for the 200_10 instances, most of the heuristics, and CPLEX found better results for the tight instances. Overall, the 1500_30 instances were the most difficult to solve.

Table 7 shows the computational times for the rest of the instance types, by instance size. We present the maximum and average computational times in seconds for the instances that CPLEX could solve and for each of the heuristics. Low computational times are observed for all the instance types, with the heuristics reaching the maximum computational

Table 7: Computational times (seconds).

Size	Metric	CPLEX	K-S/P	1-S	HypLS	2-IterLS	CPLEX	K-S/P	1-S	HypLS	2-IterLS
				Asymmetric					Semisymmetric		
50_5	Max	7203.00	0.99	0.56	0.81	1.31	471.63	10.38	7.19	9.27	13.91
	Avg	3200.60	0.80	0.41	0.65	1.06	70.33	9.03	5.94	8.30	12.02
200_10	Max	7209.90	20.40	16.44	20.64	32.91	3605.50	84.81	53.36	78.16	112.75
	Avg	4604.90	17.80	14.01	17.88	28.15	3603.00	61.69	43.56	58.16	80.47
450_15	Max		150.40	140.05	167.89	260.78		84.81	53.36	78.16	112.75
	Avg		119.60	116.52	132.65	209.39		61.69	43.56	58.16	80.47
1000_20	Max		1793.30	1173.80	339.13	1555.9		608.38	431.92	450.78	737.47
	Avg		1441.00	1032.10	295.02	1303.1		494.73	351.37	404.31	585.84
1500_30	Max		3779.50	3829.10	3754.70	3898		2233.50	1595.50	4097.20	2774.30
	Avg		3610.30	3733.50	2546.70	3730.8		1815.40	1286.20	2638.10	2251.10
Size	Metric			Urban					Symmetric		
50_5	Max	1174.60	0.92	0.42	0.64	1.03	621.96	11.37	7.12	10.09	15.32
	Avg	467.10	0.70	0.31	0.51	0.80	84.94	9.53	6.14	8.66	12.43
200_10	Max	3606.00	13.35	6.59	11.73	17.86	3605.50	72.32	50.61	65.17	98.34
	Avg	3603.70	9.72	5.47	8.64	12.00	2201.90	60.00	45.00	56.56	78.36
450_15	Max		96.28	59.00	90.81	120.89		72.3	50.61	65.17	98.34
	Avg		86.28	48.80	84.35	94.97		60.00	45.00	56.56	78.36
1000_20	Max		577.90	413.00	558.00	558.00		660.70	379.21	533.70	770.64
	Avg		952.30	603.70	878.60	878.60		561.60	317.43	488.30	620.73
1500_30	Max		5307.00	40970	4097.20	4260.00		2855.80	1523.90	4097.70	2917.20
	Avg		3912.20	34970	3840.50	3867.50		2224.10	982.30	2873.30	2430.00

Table 8: Parcel instances results.

Gap	Inst.	K-S/P	1-S	HypLS	2-IterLS	Inst.	K-S/P	1-S	HypLS	2-IterLS
Max		−21.80%	−25.26%	−23.54%	−24.93%		−19.92%	−19.39%	−22.02%	−23.48%
Avg	T	−32.30%	−38.70%	−36.06%	−38.23%	All	−31.40%	−34.23%	−34.90%	−37.11%
Min		−43.67%	−50.45%	−47.01%	−49.87%		−43.67%	−50.45%	−47.01%	−49.87%
Max		−19.92%	−19.39%	−22.02%	−23.48%		−19.92%	−19.39%	−22.02%	−23.48%
Avg	LR	−30.51%	−29.76%	−33.74%	−35.99%	Speed 25	−30.22%	−34.21%	−34.88%	−37.09%
Min		−39.75%	−38.83%	−43.97%	−46.92%		−39.74%	−50.43%	−46.99%	−49.86%
Max		−33.19%	−38.83%	−43.95%	−46.90%		−19.98%	−19.45%	−22.08%	−23.53%
Avg	λ: 0.25	−39.96%	−44.63%	−45.48%	−48.39%	Speed 30	−31.99%	−34.23%	−34.90%	−37.11%
Min		−43.67%	−50.45%	−47.01%	−49.87%		−43.66%	−50.44%	−47.00%	−49.86%
Max		−31.80%	−31.02%	−35.16%	−37.52%		−19.32%	−19.99%	−19.46%	−22.10%
Avg	λ: 0.5	−33.37%	−35.69%	−36.40%	−38.71%	Speed 35	−29.56%	−32.00%	−34.25%	−34.92%
Min		−34.92%	−40.39%	−37.64%	−39.90%		−38.60%	−43.67%	−50.45%	−47.01%
Max		−19.92%	−19.39%	−22.02%	−23.48%		4041.4	1146.68	1569.92	1690.34
Avg	λ: 0.75	−20.89%	−22.36%	−22.82%	−24.23%	Comput. time	1632.66	1125.19	1538.04	1612.27
Min		−21.85%	−25.33%	−23.62%	−24.98%		1479.25	1103.93	1506.26	1535.26

time only for the asymmetric and urban largest size instances of 1500_30, which are the most realistic and difficult to solve.

Table 8 displays the results obtained with each method for the parcel instances. A gap is computed with respect to the current solution similarly to the gap presented in (11). The heuristic solution and the current districting configuration are compared evaluated with (3). Compared to the current districting design, all the procedures are able to find a better districting configuration, with the 2-IterLS heuristic finding the best results. Computational times were generally low.

Figures 1 and 2 each depict an example of a districting configuration. Figure 1 presents a graph in which districts overlap, a configuration that is not desirable for the company. In contrast, Figure 2 shows a configuration in which the districts can be distinguished from each other. The geography of the territory also determines the shape of the districts as

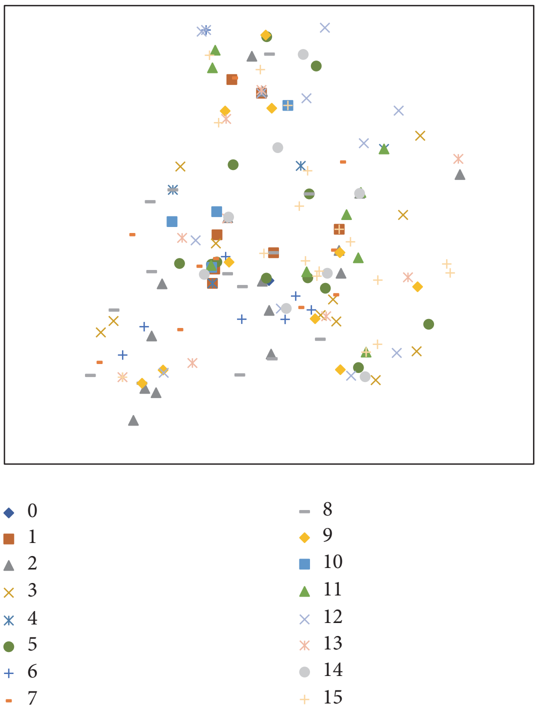

Figure 1: Illustration of districts that overlap.

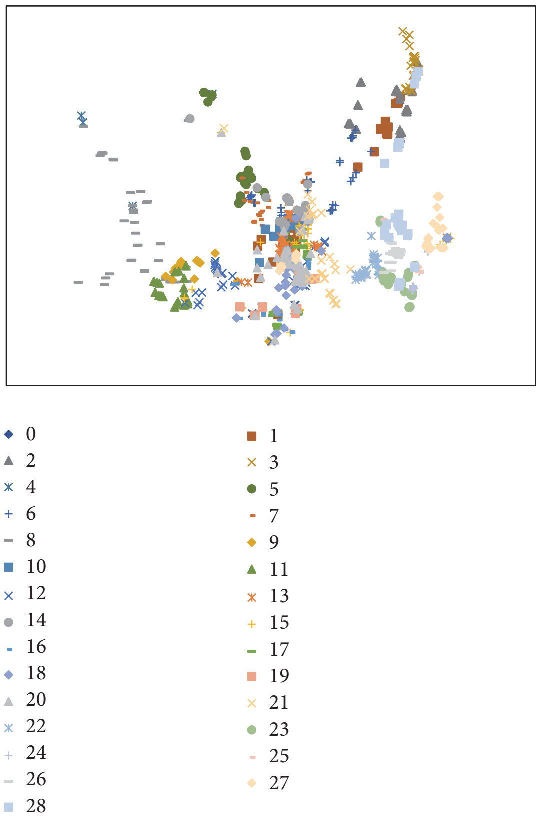

Figure 2: Illustration of districts configuration for the parcel instance.

can be seen in Figure 2, in which the districts conform to the urban geography of the region which is characterized by having several hills within and around the city.

6. Conclusions and Recommendations for Further Research

We present a mathematical formulation of the districting problem faced by a parcel company that serves a determined region. The model seeks to balance the workload content among the districts and to form districts of compact shape. The difficulty of the problem motivated us to develop a heuristic approach based on a hybrid heuristic that combines elements of GRASP and Tabu Search. To test the performance of the heuristic, we generate instances of several types and sizes, including a real instance using data from a parcel company. The results show good performance by the proposed approach, with good quality solutions and low computational times. The capacity factor is found to be the most significant factor because it conditions the difficulty of the instance to even find a feasible solution.

Further research includes formulating a stochastic version of the problem and analyzing different demand scenarios. The problem may also be solved as a biobjective optimization problem to find an efficient frontier instead of a single solution. Different mathematical programming solution approaches, such as Lagrangian Relaxation and decomposition, may also be explored. Although the heuristic is fast, it is not practical for the company to redesign the districts on a daily basis, because of managerial and human issues. Changing the districting configuration very frequently could make it very difficult for the drivers to relearn the limits of their assigned districts, so a reasonable frequency should be defined to allow the drivers to adapt to their districts and avoid making the sorting process more subject to mistakes. It is important to consider that despite the speed of the heuristic, gathering data from the delivery and pickup scanners and the GPS devices requires considerable time because the system is not yet fully automated. In addition, not every day is a suitable representative day, which must meet several criteria and must be chosen with care. For these reasons, another future research topic could be to determine the optimal redesign frequency.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Altman, “Is Automation the Answer: The Computational Complexity of Automated Redistricting,” *Rutgers Computer and Law Technology Journal*, vol. 23, no. 1, pp. 81–142, 1997.

[2] R. L. Keeney, “A Method for Districting Among Facilities,” *Operations Research*, vol. 20, no. 3, pp. 613–618, 1972.

[3] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub, “School redistricting: Eembedding GIS tools with integer programming,” *Journal of the Operational Research Society*, vol. 55, no. 8, pp. 836–849, 2004.

[4] D. S. Lemberg and R. L. Church, "The school boundary stability problem over time," *Socio-Economic Planning Sciences*, vol. 34, no. 3, pp. 159–176, 2000.

[5] K. Haase and S. Müller, "Management of school locations allowing for free school choice," *OMEGA - The International Journal of Management Science*, vol. 41, no. 5, pp. 847–855, 2013.

[6] J. A. Ferland and G. Guenette, "Decision support system for the school districting problem," *Operations Research*, vol. 38, no. 1, pp. 15–21, 1990.

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

[8] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.

[9] K. Wong and J. Beasley, "Vehicle routing using fixed delivery areas," *Omega* , vol. 12, no. 6, pp. 591–600, 1984.

[10] J. E. Beasley, "Fixed routes," *Journal of the Operational Research Society*, vol. 35, no. 1, pp. 49–55, 1984.

[11] C. F. Daganzo, "The length of tours in zones of different shapes," *Transportation Research Part B: Methodological*, vol. 18, no. 2, pp. 135–145, 1984.

[12] C. F. Daganzo, "The distance traveled to visit N points with a maximum of C stops per vehicle: an analytic model and an application," *Transportation Science*, vol. 18, no. 4, pp. 331–350, 1984.

[13] G. F. Newell and C. F. Daganzo, "Design of multiple-vehicle delivery tours—I a ring-radial network," *Transportation Research Part B: Methodological*, vol. 20, no. 5, pp. 345–363, 1986.

[14] A. F. W. Han and C. F. Daganzo, "Distributing nonstorable items without transshipment," *Transportation Research Record, 1061*, pp. 32–41, 1986.

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

[16] A. Langevin and F. Soumis, "Design of multiple-vehicle delivery tours satisfying time constraints," *Transportation Research Part B: Methodological*, vol. 23, no. 2, pp. 123–138, 1989.

[17] D. B. Rosenfield, I. Engelstein, and D. Feigenbaum, "An application of sizing service territories," *European Journal of Operational Research*, vol. 63, no. 2, pp. 164–172, 1992.

[18] A. G. N. Novaes, J. E. Souza De Cursi, and O. D. Graciolli, "A continuous approach to the design of physical distribution systems," *Computers & Operations Research*, vol. 27, no. 9, pp. 877–893, 2000.

[19] A. G. N. Novaes and O. D. Graciolli, "Designing multi-vehicle delivery tours in a grid-cell format," *European Journal of Operational Research*, vol. 119, no. 3, pp. 613–634, 1999.

[20] L. C. Galvão, A. G. N. Novaes, J. E. Souza De Cursi, and J. C. Souza, "A multiplicatively-weighted Voronoi diagram approach to logistics districting," *Computers & Operations Research*, vol. 33, no. 1, pp. 93–114, 2006.

[21] L. Muyldermans, D. Cattrysse, D. Van Oudheusden, and T. Lotan, "Districting for salt spreading operations," *European Journal of Operational Research*, vol. 139, no. 3, pp. 521–532, 2002.

[22] L. Muyldermans, D. Cattrysse, and D. Van Oudheusden, "District design for arc-routing applications," *Journal of the Operational Research Society*, vol. 54, no. 11, pp. 1209–1221, 2003.

[23] G. García-Ayala, J. L. González-Velarde, R. Z. Ríos-Mercado, and E. Fernández, "A novel model for arc territory design: promoting Eulerian districts," *International Transactions in Operational Research*, vol. 23, no. 3, pp. 433–458, 2016.

[24] D. Haugland, S. C. Ho, and G. Laporte, "Designing delivery districts for the vehicle routing problem with stochastic demands," *European Journal of Operational Research*, vol. 180, no. 3, pp. 997–1010, 2005.

[25] J.-B. Sheu, "A hybrid fuzzy-optimization approach to customer grouping-based logistics distribution operations," *Applied Mathematical Modelling*, vol. 31, no. 6, pp. 1048–1066, 2007.

[26] F. Tavares-Pereira, J. R. Figueira, V. Mousseau, and B. Roy, "Multiple criteria districting problems: the public transportation network pricing system of the Paris region," *Annals of Operations Research*, vol. 154, pp. 69–92, 2007.

[27] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde, "Constructing efficient commercial territory design plans for a beverage distribution firm with a bi-objective programming model," *Transportation Research Part C*, vol. 19, no. 5, pp. 885–895, 2011.

[28] A. G. N. Novaes, J. E. Souza de Cursi, A. C. L. da Silva, and J. C. Souza, "Solving continuous location-districting problems with Voronoi diagrams," *Computers & Operations Research*, vol. 36, no. 1, pp. 40–59, 2009.

[29] R. G. González-Ramírez, N. R. Smith, R. G. Askin, and V. Kalashinkov, "A heuristic approach for a logistics districting problem," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 8, pp. 3551–3562, 2010.

[30] R. G. González-Ramírez, N. R. Smith, R. G. Askin, P. A. Miranda, and J. M. Sánchez, "A hybrid metaheuristic approach to optimize the districting design of a parcel company," *Journal of Applied Research and Technology*, vol. 9, no. 1, pp. 19–35, 2011.

[31] P. A. Miranda, R. G. Gonzalez-Ramirez, and N. Smith, *Districting and Customer Clustering within Supply Chain Planning: A Review of Modeling and Solution Approaches*, Sanda Renko, Ed., In Supply Chain Management - New Perspectives. Intech Publisher, 2011.

[32] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Labs Technical Journal*, vol. 49, no. 1, pp. 291–307, 1970.