An Improved GRASP with Path Relinking for a Commercial Territory Design Problem

Hugo Jair Escalante
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
hugojair@yalma.fime.uanl.mx

Roger Z. Ríos-Mercado[1]
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
roger.rios@uanl.edu.mx

July 2011

[1]Corresponding author

Abstract

This paper presents an improved GRASP with path relinking for the commercial territory design problem. We face the problem of grouping basic commercial units into larger geographic territories subject to dispersion, connectivity, and balance requirements. The problem is motivated by a real-world application from the bottled beverage distribution industry. For solving this particular territory design problem we propose an improved GRASP that incorporates a novel construction procedure where territories are formed simultaneously in two main stages using different criteria. The GRASP is further enhanced with two variants of forward-backward path relinking, namely static and dynamic. Both path relinking strategies resulted very helpful for further improving solutions with respect to those obtained with a former approach. The proposed algorithm, called GPR_CTDP, has been extensively evaluated over a wide set of data instances. Experimental results revealed that the construction mechanism improves the corresponding procedure from previous work. Also the two variants of path relinking implemented in GPR_CTDP allowed us to obtain better solutions than those obtained when using straight local search. Compared to previous work, it was also observed that the proposed method outperformed the existing method in terms of solution quality. The ideas and component of the developed method can be further extended to other districting problems under balancing and connectivity constraints.

Keywords: Commercial territory design; Metaheuristics; GRASP; Path relinking

==

1 Introduction

The territory design problem (TDP) may be viewed as the problem of grouping small geographic basic units (BUs) into larger geographic clusters, called territories, in a way that the territories are acceptable (or optimal) according to relevant planning criteria. Territory design or districting has a broad range of applications such as political districting [3, 4, 17, 25, 35], sales territory design [10, 42, 43], school districting [7, 40], power districting [1], public services [2, 8, 26], to name a few. The reader can find in the works of Kalcsics, Nickel, and Schröder [22] and Duque, Ramos, and Suriñach [11] state of the art surveys on models, algorithms, and applications to districting problems.

The problem addressed in this paper is a commercial territory design problem (CTDP) motivated by a real-world application from the bottled beverage distribution industry. The problem, introduced by Ríos-Mercado and Fernández [36], considers finding a design of p territories with minimum dispersion subject to planning requirements such as exclusive BU-to-territory assignment, territory connectivity, and territory balancing with respect to three BU attributes: number of customers, product demand, and workload. Other works in commercial territory design have focused on models with additional side constraints such as joint-assignment [6], i.e., additional requirement that state that some specific pairs of BUs must belong to the same territory, models that incorporate several optimization criteria [38, 39].

In this work, we present an improved GRASP with Path Relinking for the CTDP (GPR_CTDP) with several features that enhances the previous work of [36]. In our proposed GRASP we develop a procedure that builds exactly p territories at once simultaneously, that is, we start with p node seeds and start associating nodes to the seeds until all of them are assigned. By growing the territories simultaneously rather than one at a time as done previously, one expects that the violation of the balancing constraints be considerably lower. In addition, we develop two path relinking strategies, one dynamic and one static, motivated by the work of Resende et al. [28], who successfully applied it to the max-min diversity problem. In our work, these PR strategies rely on finding a "path" between two different territory designs. To this end, an associated assignment subproblem for finding the best match between territory centers is solved. The solution to this problem provides a very nice way of generating the trajectory between two given designs. This idea is novel in any districting of territory design application to the best of our knowledge.

To assess its efficiency, the proposed proposed GPR_CTDP has been extensively evaluated over a wide set of data instances. We found that that the proposed GPR_CTDP compares favorably in performance with respect to previous work. More specifically, we found that the construction procedure improves the corresponding procedure from previous work. Hence growing territories simultaneously proved to be an effective mechanism for obtaining very competitive solutions. The two PR variants implemented in GPR_CTDP allowed us to obtain better solutions than those obtained when using straight local search; although, the dynamic strategy resulted more helpful. When compared to the previous approach, it was also observed that the proposed method outperformed in solution quality the existing technique. The main algorithmic ideas incorporated in the developed algorithm can be extended so as to handle other districting problems with similar structure.

The paper organized as follows. In Section 2 we describe the problem in detail and present a combinatorial optimization model. Section 3 gives an overview of relevant previous related work. Section 4 describes in detail the components of the proposed heuristic, and Section 5 presents the empirical evaluation of the method. We end the paper in Section 6, with some conclusions and final remarks.

2 Problem Description

Let $G = (V, E)$ denote a graph where V is the set of city blocks or basic units (BUs), and E is the set of edges representing adjacency between blocks, that is, $(i, j) \in E$ if and only if BUs i and j are adjacent blocks. Let d_{ij} denote the Euclidean distance between BUs i and j, with $i, j \in V$. For each BU $i \in V$ there are three associated parameters. Let w_i^a be the value of activity a at node i, where $a = 1$ (number of customers), $a = 2$ (product demand), and $a = 3$ (workload). The number of territories is given by the parameter p. A p-partition of V is denoted by $X = (X_1, \ldots, X_p)$, where $X_k \subset V$ is called a territory of V. Let $w^a(X_k) = \sum_{i \in X_k} w_i^a$ denote the size of territory X_k with respect to activity $a \in A = \{1, 2, 3\}$. The balancing planning requirements are modeled by introducing a user-specified tolerance parameter τ^a that measure the allowable relative deviation from the target average size μ^a, given by $\mu^a = w^a(V)/p$, for each activity $a \in A$. Another planning requirement is that all of the nodes assigned to each territory are connected by a path contained totally within the territory. In other words, each of the territories X_k must induce a connected subgraph of G. Finally, we seek to maximize territory compactness or, equivalently, minimize territory dispersion, where dispersion is given by $\max_{k=1,\ldots,p} \max_{j \in X_k} \{d_{c(k),j}\}$, where $c(k)$ denotes the index of the center of territory X_k, and is determined by

$$c(k) = \arg \min_{j \in X_k} \min_{i \in X_k} \{d_{ij}\}.$$

This is the same function used to measure dispersion in the p-Center Problem (pCP).

Let Π be the collection of all p-partitions of V. The combinatorial optimization model is given as follows.

Model (CTDP)

$$\min_{X\in\Pi} \quad f(X) = \max_{k\in K}\max_{i\in X_k}\left\{d_{c(k),i}\right\} \quad (1)$$

$$\text{subject to} \quad \frac{w^a(X_k)}{\mu^a} \in [1-\tau^a, 1+\tau^a] \quad k\in K,\ a\in A \quad (2)$$

$$G_k = G(V_k, E(V_k)) \text{ is connected} \quad k\in K \quad (3)$$

Objective (1) measures territory dispersion. Constraints (2) represent the territory balance with respect to each activity measure as it establishes that the size of each territory must lie within a range (measured by tolerance parameter τ^a) around its average size. Constraints (3) guarantee the connectivity of the territories, where G_k is the graph induced in G by the set of nodes X_k. Note that there is an exponential number of such constraints. An equivalent MILP formulation is given in [36].

The model can be viewed as partitioning G (the contiguity graph representing the BUs) into p connected componentes (contiguous districts) under the additional side constraints on bakancing product demand, number of customers, and workload of each territory, and minimizing a dispsersion measure of the BUs in a territory. The basic contiguity graph model for the representation of a territory divided into elementary units has been adopted in political districting [21, 27, 35]. This CTDP is NP-hard [36].

3 Related Work

Territory design or districting has a broad range of applications such as political districting [4, 17, 3, 25, 35], sales territory design [10, 42, 43], school districting [7, 40], power districting [1], public services [2, 8, 26], to name a few. The reader can find in the works of Kalcsics, Nickel, and Schröder [22] and Duque, Ramos, and Suriñach [11] state of the art surveys on models, algorithms, and applications to districting problems. Zoltners and Sinha [43] present a survey focusing on sales districting.

Here we discuss the related work on commercial territory design which is mostly related to our proposal. A first model was introduced by Vargas-Suárez et al. [41]. They consider a CTDP with multiple balancing and connectivity requirements, aiming at maximizing territory balancing. They did not consider the compactness issue. They used a variable number of territories. They develop a GRASP for obtaining feasible designs for large instances with relative success.

Later, Ríos-Mercado and Fernández [36] extended this model by incorporating a territory compactness criterion, and a fixed number of territories p. They seek to maximize this compactness criterion subject to planning requirements such as exclusive BU-to-territory assignment, territory connectivity, and territory balancing with respect to three BU attributes: number of customers, product demand, and workload. In their work, the authors consider as a minimization function a dispersion function based on the p-Center Problem [23] objective function. After establishing the NP-completeness of the problem, the authors propose a Reactive GRASP for obtaining high-quality solutions to this problem. The core of their GRASP is a three-phase iterative procedure composed by a construction phase, an adjustment phase, and a local search phase. In the construction phase a solution with q territories, where q is usually larger than p, satisfying the connectivity constraints is built. Then an adjustment phase based on a pairwise merging mechanism is applied to obtain a solution with p territories. Afterwards, a local search phase attempting both to eliminate the infeasibility with respect to the balancing requirements and to improve the dispersion objective function is applied. One interesting observation is that the construction and adjustment phases produce solutions with very high degree of infeasibility. This is very nicely repaired by the local search, at a very high computational cost though. The reason for this is that attempting to merge two territories into one in the adjustment phase may result in a high violation of the upper bound of the balancing constraints.

Caballero-Hernández et al. [6] study a similar CDTP with additional joint assignment constraints, that is, a requirement that given pairs of customers must both be assigned to the same territory. They develop a GRASP for instances of 500- and 1000-nodes based on a preprocesing stage where a k-shortest path algorithm is used for assuring the joint assignment constraints are met.

Aguilar-Salazar et al. [37] present an exact optimization framework for tackling relatively small instances of several CDTP models. They studied two linear models that differ in the way they measure dispersion, one model uses a dispersion function based on the objective of the p-Median Problem (MPTDP) and the other is based on the p-Center Problem (CPTDP). They can successfully solve instances of up to 100 BUs for the CPTDP and up to 150 BUs for the MPTDP. This concludes that center-based dispersion measures yield more difficult models as they have weaker LP relaxations that the median-based models.

More recently, CTDP has been addressed from a multiobjetive optimization perspective. Salazar-Aguilar et al. [38] present an exact optimization method for obtaining Pareto fronts for relatively small instances. Salazar-Aguilar et al. [39] develop heuristic methods for addressing larger instances.

In this work, we address the CTDP model as presented by [36] and present an improved GRASP with Path Relinking (GPR_CTDP) with several features that enhances the previous work obtaining designs of considerably better quality. The details of the proposed procedure are discussed in Section 4.

4 Proposed Heuristic

This section introduces the proposed GRASP heuristic with path relinking for the commercial territory design problem (GPR_CTDP). GRASP is a well known meta-heuristic based on greedy search and random construction mechanisms [14]; it has been successfully used for solving many combinatorial optimization problems [29], including CTDP [36]. We propose a GRASP improved with path relinking (PR) that compares favorably in performance with respect to previous work. The improvements comprise a new construction procedure and a very effective PR mechanism. The construction procedure improves considerably the corresponding procedure from previous work, while the PR formulation allows us to obtain better solutions than those obtained when using straight local search, see Section 5. The rest of this section describes in detail the components of the GPR_CTDP approach, which receives as input an instance of the CTDP and a set of parameters as described below.

4.1 GRASP

A GRASP is an iterative process in which each major iteration consists of two phases: construction and local search [14, 29]. The construction phase attempts to build a feasible solution and the local search phase attempts to improve it. This process is repeated for a fixed number of iterations and the best overall solution is returned as the result. GRASP incorporates greedy search and randomization mechanisms that allow it to obtain high quality solutions to combinatorial problems in acceptable times. Despite the simplicity of this multi-start heuristic it has proved to be very effective in a wide range of problems and applications. We refer the reader to the following references for complete surveys on GRASP [15, 16, 29, 31]. Previous work on GRASP for the CTDP is presented in Section 3. In this paper we propose procedure GPR_CTDP, which is in essence a GRASP augmented with PR mechanisms, accordingly, in this section we describe the particular construction and local search procedures of the GRASP and the next subsection presents the PR strategies.

Construction phase

At a given iteration, the construction phase consists of building p territories, $X_1, \ldots, X_p$, simultaneously in such a way that connectivity is always satisfied while infeasibility in terms of dispersion and balance is allowed to some extend. Each territory X_k is formed by a subset of BUs or nodes such that $\cup_{k=1,\ldots,p} X_k = V$ and $X_k \cap X_l = \emptyset$, for all $k \neq l$. Under the proposed procedure each territory X_k is associated to a center, $c(k)$. This is not a requirement of the problem but a feature of the proposed formulation that was adopted for convenience when measuring dispersion of territories.

```
Procedure 1 grasp_construction( δ, L, α )
Input: δ: fraction of nodes assigned by the distance criteria;
  L: interval for updating centers;
  α: RCL quality parameter;
Output: X: A p-partition of V;
  (c(1), ..., c(p)) ← max_disp( p ); {Compute p initial centers}
  {Stage 1}
  i ← 0;   V̄ ← V;
  while ( n − |V̄| ≤ δn ) do
      for all ( k ∈ {1, ..., p} ) do
          N_q(X_k) ← q nearest (unassigned) neighbors of X_k;
          X_k ← X_k ∪ N_q(X_k);   V̄ ← V̄ \ N_q(X_k);
      end for
      i ← i + 1;
      if ( i module L = 0 ) then
          c(k) ← min(max d_{v,w}), ∀v, w ∈ X_k, k = 1, ..., p; {Update centers}
      end if
  end while
  {Stage 2}
  open(k) ← TRUE, k = 1, ..., p;
  while ( |V̄| > 0 and ∃k such that open(k) == TRUE ) do
      for all ( k = 1, ..., p ) do
          if ( open(k) == TRUE ) then
              N(X_k) ← Set of neighbors of of X_k;
              Compute φ_k(v) in Eq. (4), ∀ v ∈ N(X_k);
              Φ_min ← min{φ_k(v)}; Φ_max ← max{φ_k(v)};
              RCL ← {h ∈ N(X_k) : φ_k(h) ≤ Φ_min + α(Φ_max − Φ_min)};
              Choose v ∈ RCL randomly;
              X_k ← X_k ∪ {v}; V̄ ← V̄ \ {v};
              if ( N(X_k) = ∅ or w^a(X_k) > (1 + τ^a) for any a ) then
                  open(k) ← FALSE; {Close this territory}
              end if
          end if
      end for
  end while
  {Postprocessing step}
  if (|V̄| > 0) then
      for all ( v ∈ V̄ ) do
          X_v ← Nearest territory to node v;
          X_v ← X_v ∪ {v}; V̄ ← V̄ \ {v};
      end for
  end if
  return X = {X_1, ..., X_p};
```

Procedure 1 presents the construction phase of the proposed GPR_CTDP. $\bar{V}$ denotes the set of nodes that have not been assigned to any territory and $n = |V|$ the number of

BUs. The process starts by selecting p centers, $c(1), \dots, c(p)$, which are the first nodes assigned to each territory; that is, $c(k) \in X_k$, $k \in \{1, \dots, p\}$. Territories are then built iteratively in two main stages followed by a postprocessing stage. In the first stage q BUs are iteratively assigned to each territory X_k. For each territory X_k, we iteratively assign the q (unassigned) nearest neighboring nodes of that territory, $v \in N_q(X_k)$. The BUs in $N_q(X_k)$ that are assigned to X_k must be connected by an edge to a BU already assigned to X_k. The latter process is iterated until a fraction (δ) of the total of BUs have been assigned to one of the p territories, where the centers $c(1), \dots, c(p)$ are updated every L iterations.

Figure 1 shows the BUs assigned after stage one of the construction phase for an instance of the CTDP considered for experimentation. From this stage the p territories have been simultaneously built by using a neighborhood criteria completely ignoring the balance constraints. The rationale behind this proposal is that nodes that belong to the same territory must be close to each other, hence a portion of nodes can be assigned with a closeness criterion. The remaining nodes will lie at boundaries among territories, therefore, balance and dispersion information is taken into account for assigning those nodes.

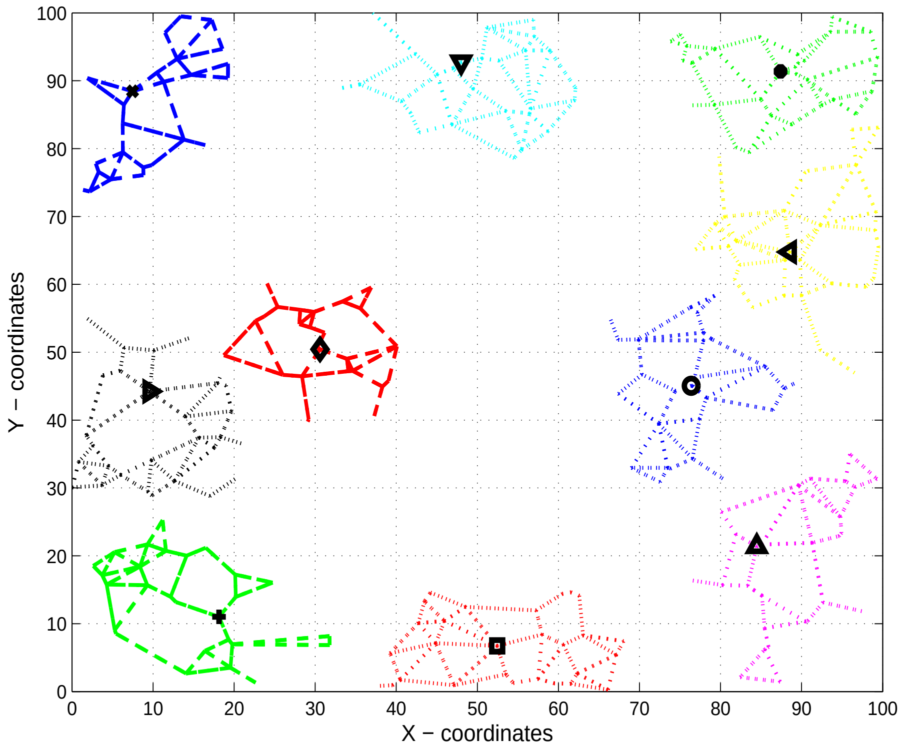

Figure 1: First stage of the proposed construction procedure for an instance of the CTDP.

A crucial aspect of stage one is that of selecting seed centers. Clearly, randomness must be considered for this process as we want to generate fairly different centers at each iteration of the GPR_CTDP approach. Nevertheless, a purely random approach could lead to obtain inappropriate seed centers (e.g., centers close to each other), as every node $v \in V$

would have the same chances to be selected as center. To this end, we view the problem of choosing an appropriate set of p initials seeds as a p-Dispersion Problem (Erkut et al. [12]), which is a combinatorial optimization problem that places p points in the plane as far way of each other as possible by using an appropriate measure for maximizing dispersion. In our procedure, we used an approach that selects centers randomly with a maximum dispersion criteria. The particular strategy starts with a randomly selected node as the center for the first territory and the rest of centers are obtained by using a greedy heuristic for the $p-$dispersion problem [12]. In our procedure, we used a particular strategy that starts with a randomly chosen node as the center for the first territory and the rest of centers are obtained by using a greedy heuristic for the p-Dispersion Problem.

The second stage of the construction phase consists of assigning the remaining $n - \delta n$ nodes that were not assigned in stage one. For this stage BUs are assigned to territories using a greedy randomized adaptive procedure that takes into account both balance and dispersion constraints. For each territory X_k, the cost of assigning every neighboring node $v \in N(X_k)$ to X_k is evaluated according to Equation (4). Then a restricted candidate list (RCL) is formed, from which a single BU is randomly selected and assigned to the current territory X_k. This RCL is restricted by a quality parameter α, that is, RCL is formed by those BUs whose greedy function evaluation falls within α percent form the best evaluation. Equation (4) determines the cost incurred when assigning node v to a territory X_k. This cost is determined by a linear combination of the weights assigned to nodes in territory $X_k \cup \{v\}$, as determined by the term $G_k(v)$, and the dispersion of those nodes, as estimated by the term $F_k(v)$. With $G_k(v)$ and $F_k(v)$ are defined in Equations (5) and (6), respectively

$$\phi_k(v) = \lambda F_k(v) + (1-\lambda) G_k(v), \quad (4)$$

$$G_k(v) = \sum_{a \in A} g_k^a(v), \quad (5)$$

$$F_k(v) = \left(\frac{1}{d_{\max}}\right) f(X_k \cup \{v\}) = \left(\frac{1}{d_{\max}}\right) \max\left\{ f(X_k), \max_{i \in X_k} d_{vi}) \right\}, \quad (6)$$

where $f(X_k) = \max_{i,j \in X_k} d_{ij}$ is the dispersion measure (as dictated by the objective function) and $g_k^a(v) = \frac{1}{\mu^a} \max\{w^a(X_k \cup \{v\}) - (1+\tau^a)\mu^a, 0\}$ accounts for the sum of relative infeasibilities for the balancing constraints. Here $d_{\max} = \max_{i,j \in V}\{d_{ij}\}$, the maximum distance between any pair of nodes, is used for normalizing the objective function. One should note that $g_k^a(v)$ represents the infeasibility with respect to the upper bound of the balance constraint for activity a. Both factors dispersion and balancing are weighted by a parameter λ in expression (4). The process is repeated for every territory k. If a territory exceeds the expected average weight for a territory it is considered *closed* (i.e., *open*$(j) = false$) and no further node can be assigned to it. The latter process iterates until either every node has been assigned to a territory or every territory is considered closed. Since stage

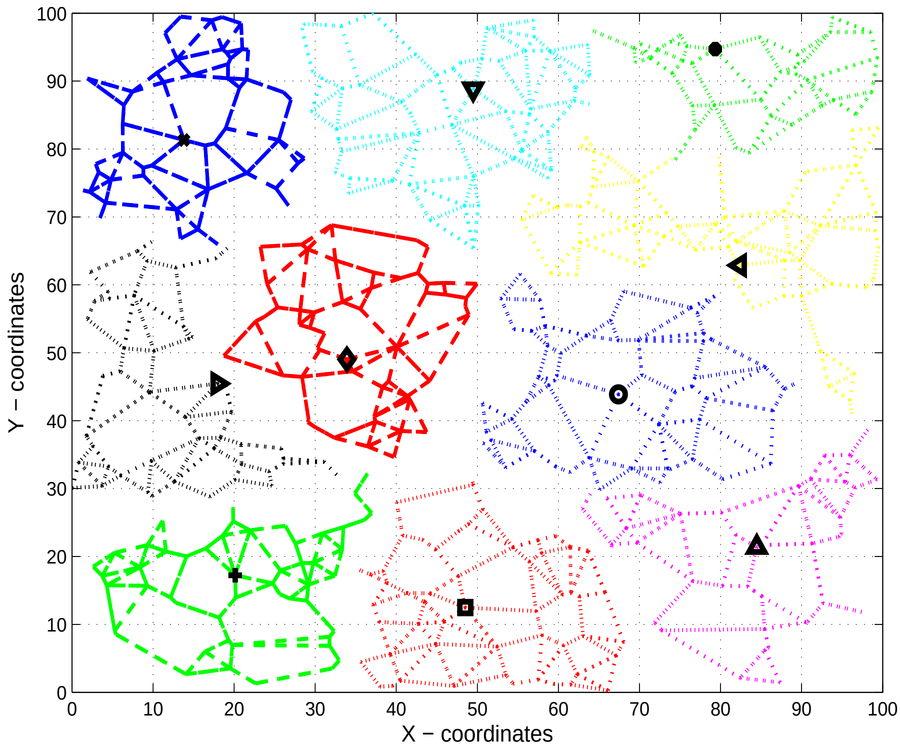

Figure 2: Second stage of the proposed construction procedure for an instance of the CTDP.

two does not guarantee that all nodes will be assigned to a territory, a postprocessing step is applied in which unassigned nodes are assigned to their nearest territory. Figure 2 shows the distribution of BUs for an instance of the CTDP after stage two of the construction procedure.

Local search

After a solution is build a postprocessing phase consisting of local search is performed. The goal in this phase is to improve the objective function value and recovering feasibility (if violated) in the constructed solution, X. In this local search, a merit function that weights both the infeasibility with respect to balancing constraints and the objective function value is used. This function is indeed similar to the greedy function used in the construction phase with the exception that now the sum of relative infeasibilities take into consideration lower and upper bound violation of the balancing constraints. Specifically, the merit function for a given territory design $X = \{X_1, \ldots, X_p\}$ is given by

$$\psi(X) = \lambda F(X) + (1 - \lambda)G(X) \tag{7}$$

where

$$F(X) = \left(\frac{1}{d_{\max}}\right) \max_{k=1,\ldots,p} \max_{h,i \in X_k} \{d_{hi}\}, \tag{8}$$

and

$$G(X) = \sum_{k=1}^{p} \sum_{a \in A} g^a(X_k), \tag{9}$$

with $g^a(X_k) = \frac{1}{\mu^a} \max\{w^a(X_k) - (1+\tau^a)\mu^a, (1-\tau^a)\mu^a - w^a(X_k), 0\}$ being the sum of the relative infeasibilities of the balancing constraints. The quality of solutions is then determined by Expression (7), we now describe the mechanism for exploring solutions around the constructed territory design. Let $t(i)$ denote the territory node i belongs to, $i = 1, \ldots, n$. A move $move(i,j)$ is defined as moving a node i from its current territory to a territory $t(j)$, where $t(j) \neq t(i)$. Only moves $move(i,j)$ where $(i,j) \in E$ and $t(i) \neq t(j)$ are allowed. Thus, $move(i,j)$ transforms a solution $X = (X_1, \ldots, X_{t(i)}, \ldots, X_{t(j)}, \ldots, X_p)$ into $X^T = (X_1, \ldots, X_{t(i)} \setminus \{i\}, \ldots, X_{t(j)} \cup \{i\}, \ldots, X_p)$. If connectivity must be kept, only moves where $X_{t(i)} \setminus \{i\}$ remains connected are allowed. Note that in general $move(i,j)$ is asymmetric.

Procedure 2 local_search(X)

```
Input: X: A solution to the CTDP;
Output: X: Improved solution to the CTDP;
  nmoves ← 0;    local_optima ← FALSE;
  k ← 1; {starting territory}
  while ( nmoves ≤ limit_evals AND ¬local_optima ) do
      improvement ← FALSE;
      while ( |N(X_k)| > 0 and ¬improvement) do
          move(i, j) ← Choose valid move from N(X_k);
          N(X_k) ← N(X_k) \ {(i, j)};
          Evaluate ψ(X^T) using Expression (7);
          if ( ψ(X^T) < ψ(X) ) then
              X ← X^T; {perform move}
              nmoves ← nmoves + 1;
              improvement ← TRUE;
              kend ← k;
              k ← (k + 1) mod p;
          end if
      end while
      if ( ¬improvement ) then
          k ← (k + 1) mod p;
      end if
      if ( k = kend ) then
          local_optima ← TRUE;
      end if
  end while
  return X
```

The basic idea of the local search is to start the search with a given territory, say territory k, and then consider first the moves emanating from territory k, that is, if we let $N(X_k)$ denote the feasible moves $move(i, j)$ with $t(i) = k$ evaluate first all the moves in $N(X_k)$, and take the best that improves the current solution, if any. If none found, proceed with territory $(k + 1) \mod p$. As soon as a better move is found, perform the move, and restart the search from this new solution X^T but setting $k + 1$ as the starting territory, where k was the last territory examined, that is, in a new move the starting territory is $k + 1$ and the final territory to be examined is k. By using this cyclic strategy for starting territory we avoid performing many unnecesary move evaluations. A move is performed using a different territory each time until no improvements can be found. In practice an additional stopping criterion: the maximum number of allowed evaluations of the fitness function (*limit_evals*), is added to avoid performing an extensive search for long periods of time. Therefore, the postprocessing step stops when either a local optima is found or the number of moves exceeds *limit_evals*. The postprocessing phase is described in Procedure 2.

4.2 Path relinking

Path Relinking (PR) was originally proposed by Glover and colleagues as a way of incorporating intensification and diversification strategies in tabu search [18, 19]. PR consists of exploring the path of intermediate solutions between two selected solutions called starting (X^S) and target (X^T) with the hypothesis that some of the intermediate solutions can be either better than X^S and X^T (intensification) or comparable but different enough from X^S and X^T (diversification). Intermediate solutions are generated by performing moves from the starting solution in such a way that these moves introduce attributes that are present in the target solution. Successful applications of PR in the context of Tabu and Scatter Search are reported in [19, 20, 32].

Despite the fact that PR was originally proposed for Tabu and Scatter search, it has been successfully used with GRASP as well [24, 30, 31, 28]. In the context of GRASP, PR can be considered as a way of introducing memory into the search process. Different variants have been proposed so far each having benefits and limitations in terms of efficiency and efficacy. In this work we consider two variants of forward-backward PR, namely static and dynamic, that have proved to be very effective in related problems [28]. For excellent surveys on applications of GRASP with PR we refer the reader to the work of Resende and Ribeiro [30, 31].

The so called, forward-backward PR strategies explore the paths between X^S and X^T in two different ways (i.e., from X^S to X^T and viceversa) [30]. The main benefit of these strategies is that more and different solutions can be generated, although it has been found that there is little gain over one-way strategies [34]. This can be due to the greediness of usual PR methods, which evaluate every possible solution that can be generated by making a move from a initial solution and choose the move that results in the best intermediate solution [34, 28]. Thus, these methods explore a large number of solutions and, therefore, forward-backward PR does not help to improve the quality of final solutions. In this work we select moves in such a way that a single move is evaluated for generating intermediate solutions. This form of PR is more efficient at the expense of sacrifying the benefit of greedy strategies. Nevertheless, we believe that in the considered setting the use of a forward-backward PR strategy is advantageous.

Besides the direction of the search, there are other aspects that make PR strategies different [30, 31, 28]. For example, greedy-randomized PR methods form a RCL with candidate moves and select a move randomly as in GRASP [13]. Truncated PR techniques explore partially the trajectory between X^S and X^T. Evolutionary PR consists of evolving a reference set of solutions in a similar way as the reference set is evolved in scatter search [33]. In this work we developed static and dynamic PR strategies that resulted very effective for the CTDP. Both strategies have been successfully used in other combinatorial optimization problems [28]. The rest of this section describes the PR strategies incorporated in GPR_CTDP.

Recall each solution of the CTDP is an assignment of every node $i \in V$ to one of p territories $X_1, \ldots, X_p$. Let $t(X, i) \in \{1, \ldots, p\}$ denote the index of the territory to which node i is assigned according to solution X. Given two particular solutions X^S and X^T, PR aims at generating intermediate solutions or p-partitions in the path starting at X^S and finishing at X^T. In GPR_CTDP intermediate solutions are created by changing $t(X^S, i)$, the territory to which node i is assigned in solution X^S into the corresponding territory $t(X^T, i)$. Because both X^S and X^T solutions are created independently, and the territory ordering may be arbitrary, it is not clear what territory in X^S corresponds to what territory in X^T. Hence, a correspondence between territories must be obtained before starting the search process. The problem of finding the bast match between territories can be set as an Assignment Problem (AP) by considering the territory centeres only. Let $C(X)$ the set of p node centers corresponding to solution X. Then a complete bipartite graph is formed with sets $C(X^S)$ and $C(X^T)$, where the cost between node $i \in C(X^S)$ and $j \in C(X^T)$ is given by d_{ij}. The AP can be solved in polinomial time by using. We use one of the most recent implementations of the Hungarian algorithm [5]. A solution to the AP represents a minimum cost assignment between territory centers, and therefore a match between territories. Let M be the solution to AP given by $M = \{(i_i, j_1), \ldots, (i_p, j_p)\}$. The idea of the PR is then to "transform" each territory $X_{t(i_k)}$ to territory $X_{t(i_k)}$ for each $(i_k, j_k) \in M$. The rationale for this matching stems from the fact that it is expected that relatively close territories (fron different designs) will have many BUs in common. This scheme is illustarted in Figure 3.

Once that a correspondence between territories has been established it is possible to perform moves from one solution X^S to another X^T. As a consequence, in order to arrive at solution X^T starting from X^S, every nodes in X^S such that $t(X^S, i)) \neq t(X^T, i)$ must be moved to its associated territory in X^T. We define a PR move, $move_{PR}(X^S, X^T, i)$, as a function that moves or reassigns a node i from territory $t(X^S, i)$ to territory $t(X^T, i)$. The move is valid as long as $t(X^S, i) \neq t(X^T, i)$ and the resulting p-particition remains conected, that is, if and only if $X_{t(X^T,i)} \cup \{i\}$ is connected and $X_{t(X^S,i)} \backslash \{i\}$ remains connected. One should note that moves are always made between boundary nodes as it is not possible to exchange a non-boundary node from one territory to another territory in a single move because loss of connectivity.

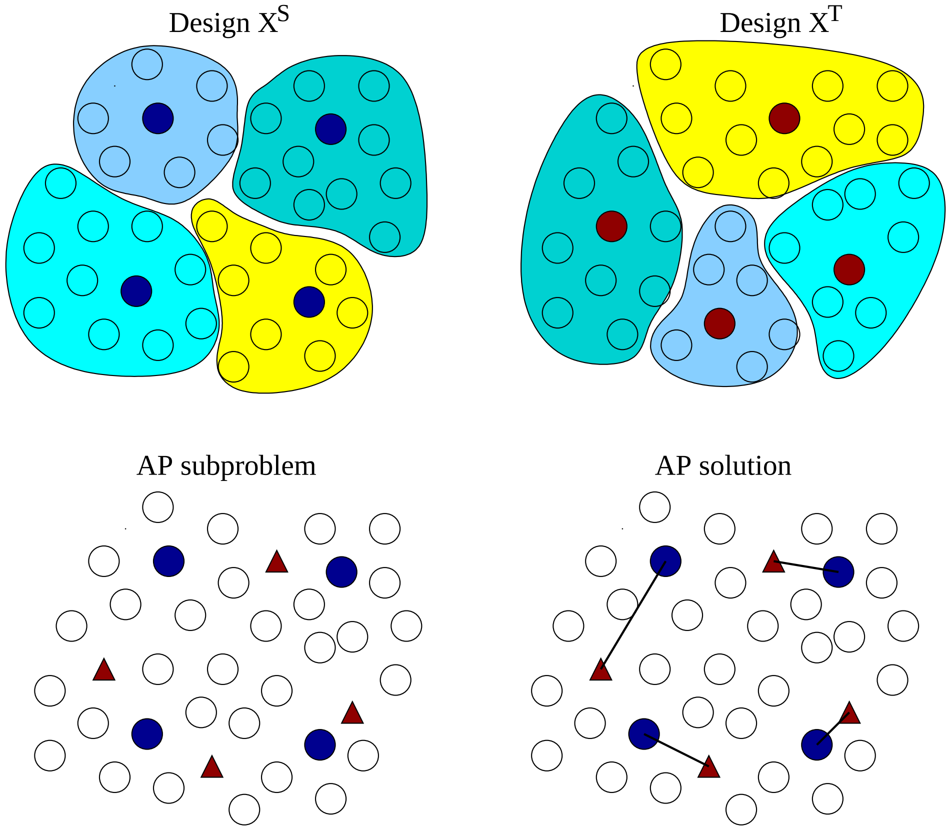

Figure 3: Illustration of how to set up a search trajectory from two given designs (top) by solving an associated Assignment Problem (bottom).

Intermediate solutions between X^S and X^T are generated by making moves from X^S to X^T and updating the solution X^S accordingly. Clearly, the order in which nodes i are selected may give rise to different trajectories between X^S and X^T. In this work we chose nodes i in lexicographical order, we also tried a random node selection approach although no difference in performance was obtained. After an intermediate solution is created it is evaluated using Formula (8). The generation-evaluating process is repeated for every node with $t(X^S, i) \neq t(X^T, i)$ and the process stops when $t(X^S, i) = t(X^T, i)$ for all $i \in V$. Thus, the PR procedure receives as input a pair of solutions X^S and X^T, generates and evaluates all of the intermediate solutions from X^S to X^T and the best intermediate solution X^R is returned as output. In the following we denote with $PR(X^S, X^T)$ the application of PR starting at solution X^S and finishing at solution X^T.

Procedures 3 and 4 present the static and dynamic variants of PR implemented in GPR_CTDP, respectively. Both static and dynamic variants maintain a set of b elite solutions $B = \{B_1, \dots, B_b\}$. B is initialized by running the construction and local search procedures for b times. Solutions in B are always kept sorted in ascending order of their objective function value estimated with Equation (8).

Procedure 3 grasp_pr_static($i_{\max}$)

```
Input: i_max: number of global iterations;
Output: X^best: A p-partition of V;
  for all ( i ∈ {1,...,b} ) do
      X ← grasp_construction();
      B_i ← local_search( X );
  end for
  Sort B from best to worst;
  for all ( iter = 1,...,i_max ) do
      X^S ← grasp_construction();
      X^S ← local_search( X^S );
      if ( (ψ(X^S) < ψ(B_1)) or (ψ(X^S) < ψ(B_b) and d_μ^sol(X^S, B) > θ) ) then
          E_j ← closest solution to X^S in B with ψ(X^S) < ψ(B_j);
          E_j ← X^S;
          Update B;
      end if
  end for
  X^best ← B_1;
  for all ( i ∈ {1,...,b−1} ) do
      for all ( j ∈ {i+1,...,b} ) do
          Apply PR(B_i, B_j) and PR(B_j, B_i) and let X^S ← best solution found;
          X^S ← local_search( X^S );
          if ( ψ(X^S) < ψ(X^best) ) then
              X^best ← X^S;
          end if
      end for
  end for
  return X^best;
```

4.2.1 Static GPR_CTDP

In the static variant, PR is performed at the end of $i_{\max}$ iterations of a typical GRASP. In each iteration of the GRASP a solution is constructed and improved with local search, X^S. This solution is compared with the solutions in B. If X^S is better than the best solution in B (i.e., B_1) or if X^S is better than the worst solution in B (i.e., B_b) and is at a distance larger than a given threshold θ from solutions in B, then the most similar solution to X^S in B is replaced by X^S. Solutions in B are then sorted from best to worst. After $i_{\max}$ iterations the static PR starts. Every path between solutions in B is evaluated and the best solution is returned. The distance between X^S and solutions in B is estimated as $d_\mu^{sol}(X^S, B) = \frac{1}{b}\sum_{i=1}^{b} g(X^S, B_i)$, where $g(X^S, B_i)$ is the fraction of nodes in X^S and B_i that are assigned to corresponding territories; that is, $d_\mu^{sol}(X^S, B)$ is the average number of nodes assigned to common territories in X^S and B_i. $\theta \in [0, 1]$ is a scalar that is set empirically. The pseudocode is shown in Procedure 3.

4.2.2 Dynamic GPR_CTDP

The dynamic PR variant differs from the static one in that in each iteration of the GRASP the solution X^S is compared to a randomly selected solution from B, say B'. The intermediate solutions between X^S and B' are evaluated, and the best solution found in the path is denoted X^R. Then if X^R is better than B_1 or if X^R is better than B_b and it is at a distance of at most θ from the solutions in B, then the closest solution in B to X^R is replaced with X^R. Then solutions in B are sorted from best to worst. After $i_{\max}$ iterations the best solution, namely B_1, is returned. The pseudocode is shown in Procedure 4.

Procedure 4 grasp_pr_dynamic($i_{\max}$)

```
Input: i_max : number of global iterations;
Output: X^best; A p-partition of V;
  for all ( i = {1,...,b} ) do
      X^S ← grasp_construction();
      B_i ← local_search( X^S );
  end for
  Sort B in ascending order;
  for all ( iter = 1,...,i_max ) do
      X^S ← grasp_construction();
      X^S ← local_search( X^S );
      Randomly select B' from B;
      Apply PR(X^S, B') and PR(B', X^S) and let X^R ← best solution found;
      if ( (ψ(X^R) < ψ(B_1)) or (ψ(X^R) < ψ(B_b) and d_μ^sol(X^R, B) > θ) ) then
          B_j ← closest solution to X^R in B with ψ(X^R) < ψ(B_j);
          B_j ← X^R;
          Update B;
      end if
  end for
  return X^best ← B_1;
```

A number of parameters are associated with GPR_CTDP in both variants, namely δ the fraction of nodes assigned with a distance criterion, k the number of neighbors that are considered for building a territory, λ the tradeoff parameter of the objective function, α the GRASP quality parameter for the RCL, *limit_evals* the maximum number of evaluations for the local search, b the number of solutions in the elite set B and θ the distance threshold in

PR. In this work we have fixed all of these parameters based on preliminary experimentation. The next section reports experimental results with the proposed GPR_CTDP.

5 Computational experiments

This section reports experimental results obtained with GPR_CTDP. The proposed method was implemented in Matlab and the code is publicly available for research purposes at *http://yalma.fime.uanl.mx/∼hugojair/code/gpr-tdp/*. All of the experiments were run in a 64-bit iMac® computer with a CoreDuo processor at 3.06Ghz and 4 GB in RAM, with Mac OS X 10.6.7 operating system.

5.1 Experimental setting

For the experiments we used the data base from [36]. These are randomly generated instances based on real-world data. Data sets DS and DT are considered for experimentation The former generate the BU weights from a uniform distribution and the latter uses a triangular distribution. Data set DT more closely resembles of the real-world instances. The data sets contains instances of many sizes as a function of n and p. These data sets are fully described in [36].

For all of the instances in both DS and DT data sets we use a tolerance level $\tau^a = 0.05$, $a \in A$. Recall that τ^a measures the allowable relative deviation from the target average size μ^a for activity a. Hence, a value of $\tau^a = 0.05$ implies that instances are tightly constrained in all activities and therefore the problem is more difficult to solve than instances that use a larger value of τ^a. In previous work [36], experiments have been reported with other values for $\tau^a \in [0.05, 0.30]$. Here we focus on the more difficult instances.

The process of randomly generating an instance is affected by two sources of information. One is the generation of the map (i.e., the graph topology), and the other one is the generation of node attributes. Instances were generated in this way because of the following reasons. On the one hand, as far as the topology is concerned, due to proprietary reasons the beverage firm has not allowed us yet to use its city map. On the other hand, the generation process allows us to test our method in a more general scenario and not making it too problem specific. In addition, since our basic geographical units are city blocks and given that one can argue that blocks are somewhat uniformly distributed over an urban setting, the generation of city blocks by an uniform distribution is a reasonable assumption. Regarding the node attribute generation, using an uniform distribution generates instances with a larger variance, which reflects the expected variance in real scenarios, and, again, this allows us to test the proposed method in a more general situation.

Both DS and DT data sets have been used in previous work [41, 36], hence the use of these data allows us to directly compare the performance of GPR_CTDP to the Reactive GRASP of Ríos-Mercado and Fernández [36] (referred to as RF heuristic), which is the best method developed for this problem to the best of our knowledge. For each of DS and DT data sets, 20 different instances of size $n = 500$ and $p = 10$ were generated. Throughout the evaluation, the GRASP is run with $i_{\max} = 500$. This number of iterations is half the number of GRASP iterations considered in the RF heuristic. Based on preliminary experimentation for fine-tuning the algorithmic parameters for GPR_CTDP, we will use the values reported in Table 1. Showing the fine-tuning of these parameters is out of the scope of this paper.

Table 1: Summary of values used for the algorithmic parameters of GPR_CTDP.

Parameter	Value	Description
δ	0.5	Fraction of nodes assigned with a distance criterion.
k	3	Number of neighbors that are considered for growing a territory.
λ	0.7	Weight parameter in the meritfunction.
α	0.3	RCL quality parameter.
$limit\_evals$	$1,000$	The maximum number of fitness function evaluations in the local search.
b	20	The number of solutions in the elite set E.
θ	0.6	The distance threshold in PR.
$i_{\max}$	500	Number of global iterations for GPR_CTDP.

In the following sections we report the obtained experimental results. We have divided experimental results in three sections that aim at assessing different aspects of the GPR_CTDP.

5.2 Construction strategy and local search (GRASP)

This section describes results of experiments designed to evaluate the effectiveness of the proposed construction and local search mechanisms in GPR_CTDP. Table 2 shows the performance of these mechanisms for both DT and DS data sets. The table shows the improvements of the local search over the construction procedure; for $\Psi(S)$ and $F(S)$ it is shown the percentage of relative improvements while for $G(S)$ we only show the actual difference. From this table we can see that the average of the sum of relative infeasibilities is maintained low in the construction procedure for both data sets. This result shows that the proposed procedure is able to obtain acceptable solutions in terms of the degree of satisfaction of the balance constraints despite the fact part of the construction procedure is based on a purely distance-based criterion.

As expected, after applying local search to the constructed solutions, the three measures $\Psi(S)$, $F(S)$ and $G(S)$ are improved. For the DT data set the objective function is improved by an average of 20.74%, while for the DS data set the improvement is of 94.90%. The fact that lower improvements are obtained for the dispersion term ($F(S)$) in both data sets gives evidence that the distance-based criterion of the construction mechanism resulted very helpful for finding competitive solutions. Interestingly, for the DS data set some solutions

Table 2: Evaluation of the construction and local search procedures of GPR_CTDP.

Measure		DT	DS
Weighted objective $\Psi(S)$	Best	8.04%	58.19%
	Average	20.74%	94.90%
	Worst	41.36%	154.20%
Objective (dispersion) $F(S)$	Best	6.02%	-8.95%
	Average	14.12%	5.49%
	Worst	31.22%	19.98%
Sum of relative infeasibilities $G(S)$	Best	$0.00E+00$	$2.50E-01$
	Average	$2.69E-02$	$4.18E-01$
	Worst	$6.12E-02$	$6.30E-01$

are not improved in terms of $F(S)$ after applying local search, showing that this procedure sacrifyces dispersion to reduce balance infeasibilities for some solutions.

Table 3: Comparison of the proposed construction procedure in GPR_CTDP with the one in the RF heuristic.

Measure		DT	DS
$\Psi(S)$	Best	62.05%	-36.90%
	Average	143.88%	-12.41%
	Worst	217.84%	12.81%
$F(S)$	Best	-10.01%	-4.52%
	Average	17.83%	17.87%
	Worst	63.36%	48.16%
$G(S)$	Best	$3.78E-01$	$4.26E-01$
	Average	$6.19E-01$	$2.08E-01$
	Worst	$8.27E-01$	$1.90E-02$

Table 4: Comparison of the proposed local search procedure in GPR_CTDP with the one in the RF heuristic.

Measure		DT	DS
$\Psi(S)$	Best	2.17%	-4.41%
	Average	6.33%	3.41%
	Worst	11.92%	24.50%
$F(S)$	Best	2.17%	-5.45%
	Average	6.40%	3.57%
	Worst	11.92%	24.50%
$G(S)$	Best	$-2.52E-03$	$-2.01E-02$
	Average	$-2.71E-04$	$-5.94E-04$
	Worst	$0.00E+00$	$1.96E-02$

Tables 3 and 4 compare the performances of the construction and local search procedures of GPR_CTDP with those introduced in [36]. In these tables, we report the relative improvement of our method over the RF heuristic. Table 3 shows that in DT, GPR_CTDP improves dramatically the construction procedure from previous work (an average of 143% in terms of $\Psi(S)$). Although for DS the construction procedure in [36] outperforms that of GPR_CTDP (an average of -12.41% in terms of $\Psi(S)$); note that GPR_CTDP outperforms previous work by an average of 17.87% in terms of dispersion, but it obtained larger values of $G(s)$. This result reflects again the suitability of the distance-based criterion in the construction mechanism, which reduces dispersion of solutions.

Table 4 compares the local search procedures in GPR_CTDP with the RF heuristic. Improvements of the proposed local search method are rather small. Again, larger improvements are observed for the dispersion term, while there is a slight decrease in terms of the sum of relative infeasibilities. This reduction in terms of balance constraints is alleviated with the PR strategies incorporated in GPR_CTDP.

5.3 GRASP vs. GPR_PR

This section reports experimental results on the improvements of the PR strategies over the straight GRASP implementation described in Section 4.1 and the RF heuristic [36]. Tables 5 and 6 show the performance of GPR_CTDP under both static and dynamic PR strategies for DT and DS data sets, respectively. In these tables, we compare the performance of GPR_CTDP when using PR and when only local search is adopted. We show the relative deviation between the best solution obtained with each method and the best known solution for each instance.

As we can see, for the DT data set (Table 5) the improvements obtained with PR over local search are small yet non-negligible. We believe this result can be due to the fact that we are approaching to the global optimum for this data set and since the local search procedure provides very competitive solutions by itself the improvements due to PR are rather small. However, it is important to emphasize that both PR strategies optimize balance infeasibilities for all of the instances in the data set, whereas it also improves the dispersion of territories. For this data set the dynamic PR strategy outperformed the static one by about 1% in terms of the objective function.

Table 5: Improvement obtained by GPR_CTDP with the static and dynamic PR variants over the GRASP for the DT data set.

Measure		Local-Search	Static PR	Dynamic PR
$\Psi(S)$	Best	0%	0%	0%
	Average	3.55%	1.48%	0.38%
	Worst	13.97%	5.42%	2.5%
$F(S)$	Best	0%	0%	0%
	Average	3.4%	1.48%	0.38%
	Worst	13.57%	5.42%	2.5%
$G(S)$	Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Average	$5.78E-04$	$0.00E+00$	$0.00E+00$
	Worst	$5.05E-03$	$0.00E+00$	$0.00E+00$

For the DS data set (Table 6) the improvements due to PR are larger. GPR_CTDP with static PR outperforms the results of local search by an average of 10% in terms of the objective function, whereas the dynamic strategy outperforms local search by 11%. Both, static and dynamic PR achieve important improvements in terms of the dispersion objective ($F(S)$), improvements in terms of balance infeasibility are small.

Table 6: Improvement obtained by GPR_CTDP with the static PR variant over the GRASP for the DS data set.

Measure		Local-Search	Static PR	Dynamic PR
$\Psi(S)$	Best	1.19%	0%	0%
	Average	13.83%	3.07%	2.35%
	Worst	50.34%	10.98%	10.26%
$F(S)$	Best	0%	0%	0%
	Average	13.06%	2.51%	2.75%
	Worst	41.92%	10.98%	10.26%
$G(S)$	Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Average	$6.05E-03$	$4.29E-03$	$1.39E-04$
	Worst	$7.75E-02$	$6.84E-02$	$2.79E-03$

Figures 4, 5, and 6 show the territories obtained with local search, static GPR_CTDP and dynamic GPR_CTDP, respectively, for an instance of the DT data set. These figures illustrate the advantages of GPR_CTDP over local search. It can be seen that territories generated with local search (Figure 4) are more disperse than those generated with GPR_CTDP. Two of the territories generated with local search are particularly disperse (circle and down-facing triangle). Both versions of GPR_CTDP improve the dispersion of that territory, although dynamic GPR_CTDP obtains territories that are better distributed.

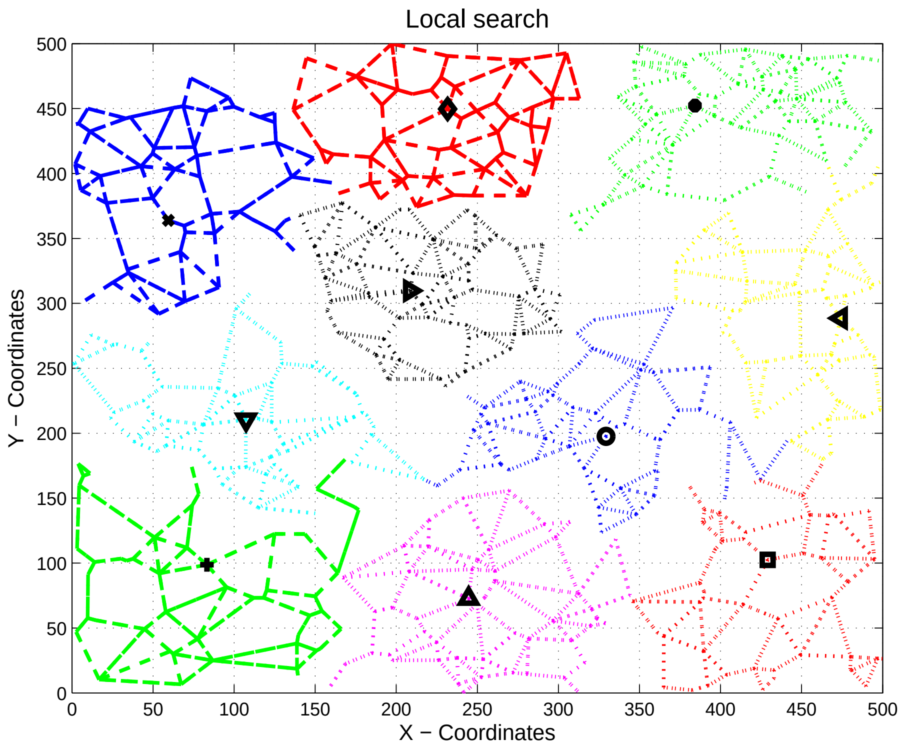

Figure 4: Solution for a particular instance of the DT data set. This is the best solution generated with the construction + local search procedures of GPR_PR.

Tables 7 and 8 compare the performance of GPR_CTDP with static (GPR_CTDP-ST) and dynamic (GPR_CTDP-DY) PR to that obtained by the RF heuristic (reactive

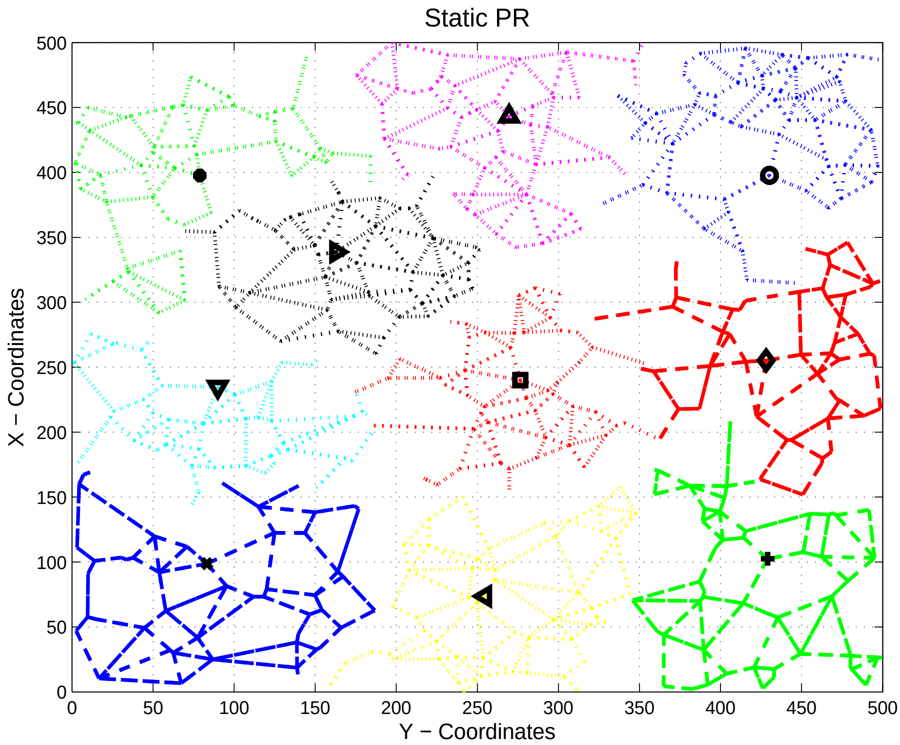

Figure 5: Solution for a particular instance of the DT data set. This is the best solution obtained after applying the static variant of GPR_PR.

GRASP). These tables show the relative deviation between the best solution obtained with each method and the best known solution for each instance. Table 7 shows the results obtained for the DT data set. The improvements over previous work for DT are of 7.44% and 8.54% for the static and dynamic PR strategies, respectively. The reactive GRASP from previous work obtained solutions that violated feasibility in terms of balance constraints ($G(S)$), while GPR_CTDP obtained feasible solutions for all of the instances. Besides the dispersion of solutions was reduced as well with GPR_CTDP.

Table 7: Comparison of GPR_CTDP with RF heuristic for the DT data set.

Measure		RF	GPR_CTDP-ST	GPR_CTDP-DY
$\Psi(S)$	Best	3.87%	0%	0%
	Average	8.92%	1.48%	0.38%
	Worst	14.54%	5.42%	2.5%
$F(S)$	Best	3.87%	0%	0%
	Average	8.91%	1.48%	0.38%
	Worst	14.54%	5.42%	2.5%
$G(S)$	Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Average	$1.84E-05$	$0.00E+00$	$0.00E+00$
	Worst	$3.67E-04$	$5.05E-03$	$0.00E+00$

Table 8 shows the results obtained for the DS data set. The improvements of GPR_CTDP over previous work are larger for this data set. The static variant of GPR_CTDP outper-

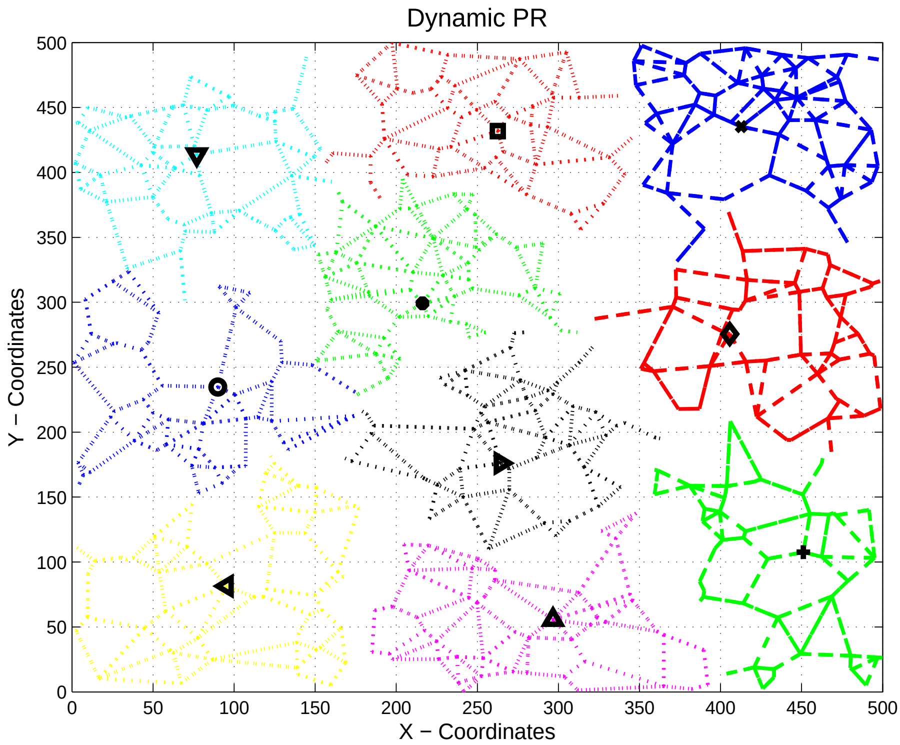

Figure 6: Solution for a particular instance of the DT data set. This is the best solution obtained after applying the dynamic variant of GPR_PR.

forms previous work by 8.98% while the dynamic strategy outperforms it by 9.70%. Static GPR_CTDP obtained better solutions than those obtained in previous work in terms of dispersion, although the sum of relative infeasibilities was larger. On the other hand, the dynamic GPR_CTDP outperformed previous work in both terms $F(S)$ and $G(S)$.

Table 8: Comparison of GPR_CTDP with RF heuristic for the DS data set.

Measure		RF	GPR_CTDP-ST	GPR_CTDP-DY
$\Psi(S)$	Best	1.15%	0%	0%
	Average	12.05%	3.07%	2.35%
	Worst	42.55%	10.98%	10.26%
$F(S)$	Best	1.15%	0%	0%
	Average	11.96%	2.51%	2.75%
	Worst	42.55%	10.98%	10.26%
$G(S)$	Best	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Average	$2.75E-03$	$4.29E-03$	$1.39E-04$
	Worst	$2.50E-02$	$6.84E-02$	$2.79E-03$

5.4 Static GPR_PR vs. dynamic GPR_PR

This section elaborates on the difference in performance between the static and dynamic PR variants of GPR_CTDP. From Tables 5 and 6 we can see that the improvements of static and dynamic GPR_CTDP over local search are of 2.07% and 3.17% for the DT data set and of 10.76% and 11.48% for the DS data set (in terms of the objective function). Thus, despite the fact both strategies resulted effective, the use of the dynamic one is advantageous. We think this can be due to the fact that in dynamic GPR_CTDP each of the $i_{\max}$ solutions are subject to PR which increases the probability of finding effective solutions via PR. Static GPR_CTDP, on the other hand, explores the paths between elite solutions at the end of the search process, exploring only a portion of the solutions that are explored in the dynamic variant. This result is consistent with previous work with static and dynamic PR [28].

Figures 7 and 8 show the relative deviation of the solutions found with each tested method and the best known solution for each instance for DT and DS data sets, respectively. These figures give us more insight into the performance of the different methods across the instances, it is rather clear that the dynamic PR strategy obtained the best solutions for most of the instances (those instances for which the relative deviation is zero), followed by the static PR approach. These plots also confirm that the improvements over the method described in previous work [36] are considerable.

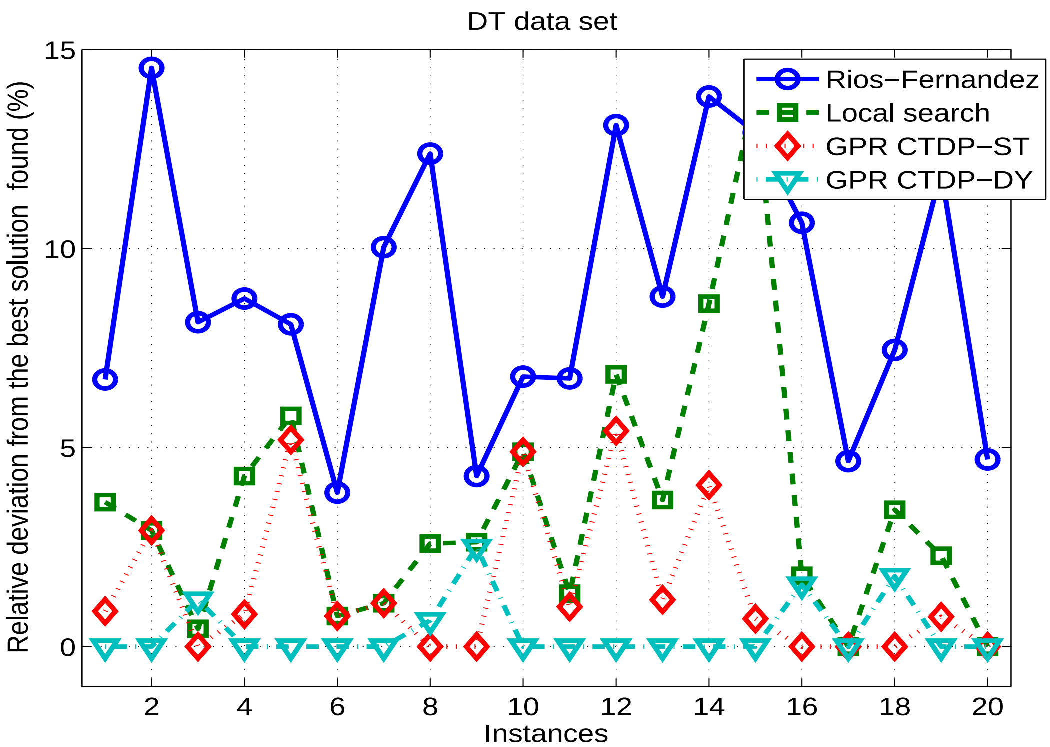

Figure 7: Relative deviation in terms of the weighted objective $\Psi(S)$ for the considered methods and for each instance of the DT data set.

Table 9 reports the processing time for each variant of GPR_CTDP and for each data set. Since more PR evaluations are performed under dynamic GPR_CTDP processing time is higher for this strategy. While it is true that the results from the existing approach were obtained in the order of 20-25 minutes on average (2000 GRASP iterations), it also

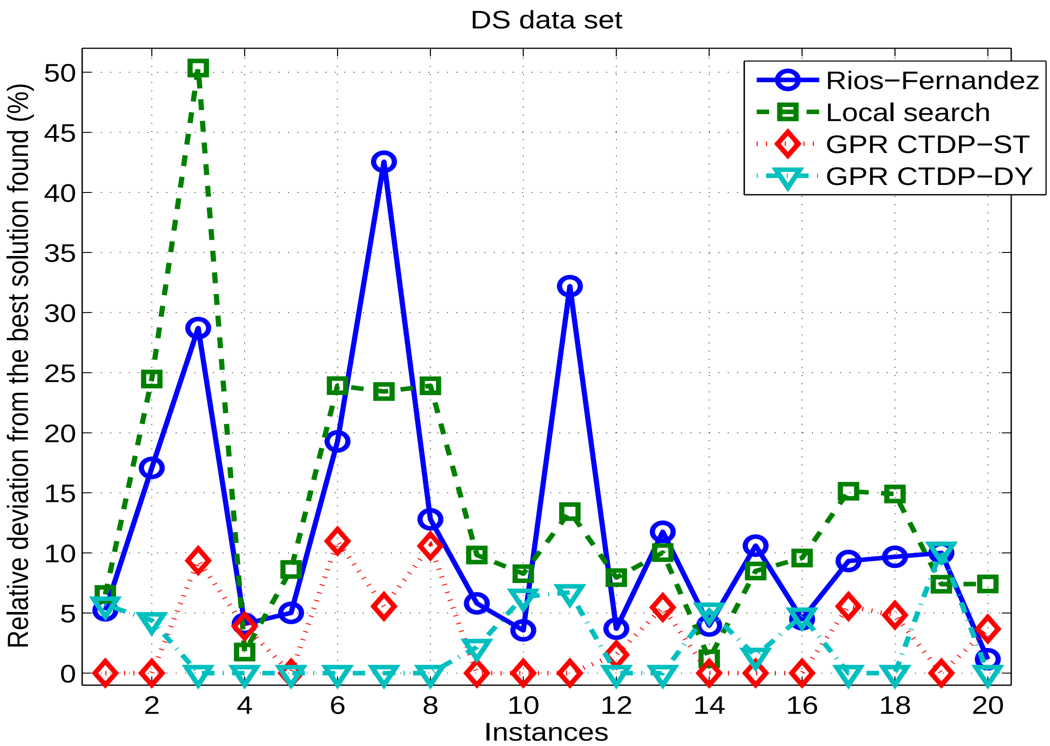

Figure 8: Relative deviation in terms of the weighted objective $\Psi(S)$ for the considered methods and for each instance of the DS data set.

Table 9: CPU time (min) comparison for static and dynamic GPR_CTDP.

	DT		**DS**	
	GPR_CTDP-ST	**GPR_CTDP-DY**	**GPR_CTDP-ST**	**GPR_CTDP-DY**
Best	188.28	197.87	234.33	250.99
Average	194.14	203.35	265.60	275.94
Worst	204.30	210.61	299.68	301.97

true that that algorithm reports very small or no improvement afterwards. The results obtained by either variant of our proposed method significantly improves the quality of the previously obtained solutions, making this extra effort worthwhile indeed. Furthermore, from the practcal standpoint, this decision is taken every 3-4 months, therefore one can certainly afford to run a few more iterations of the proposed method for quality's sake if needed.

6 Conclusions

We have described an improved GRASP with path relinking for the commercial territory design problem (GPR_CTDP). The problem, motivated by a real-world application, consists of grouping commercial units into geographic territories subject to dispersion, connectivity and balance constraints. A novel construction procedure was developed and two variants of path relinking (PR) were explored in GPR_CTDP, namely, static and dynamic PR. The components of GPR_CTDP were evaluated and compared extensively in instances that are known to be very challenging from previous work.

Experimental results show that the proposed construction procedure is able to construct very competitive solutions, mainly in terms of the dispersion criterion. The local search of the GPR_CTDP improves solutions in terms of both dispersion and balance requirements. Both versions of PR improve the performance of the application of the construction and local search mechanisms, confirming previous work on the combination of GRASP and PR. In particular we found that with the dynamic PR variant better solutions can be obtained for the TDP. This can be due to the fact that more solutions are subject to PR under dynamic PR. A consequence of the latter feature of dynamic PR is that processing time is larger for this strategy, although processing time lies in reasonable ranges. In any event the proposed method signifcantly outperformed the best existing method for this problem.

We have identified several future work directions in the context of GPR_CTDP. In particular we would like to explore other variants of PR that are known to be very effective, for example, evolutionary PR. Further, we are interested in the development of an adaptive filtering step that allows us to identify pairs of solutions that can be potentially improved by applying PR. This is in addition to the rules used for updating the set of elite solutions. We think that such a filtering strategy will have a very positive impact in the efficiency of GPR_CTDP. Since we found evidence that maintaining a set of elite solutions can be beneficial for TDP, we would like to explore the use of other "population-based" metaheuristics like scatter search.

It is important to note that the method developed in this work can also be extended and applied to other districting problems under balancing and connectivity constraints. The presence of the connectivity constraints make the path relinking process more challenging. For instance, path relinking has been applied in a different manner in related partitioning problems such as capacitated clustering [9]. In this particular work, we have successfully exploited the problem structure by solving an associated Assignment Problem whose solution will guide the relinking process in a more intelligent fashion. To the best of our knowledge this PR idea is novel and worthwhile for further exploration in other districting or clustering problems under connectivity constraints.

Acknowledgements: This work was supported by the Mexican National Council for Science and Technology under grant SEP-CONACYT 48499–Y, Universidad Autónoma de Nuevo León under its Scientific and Technological Research Support, grant UANL-PAICYT CE012–09.

References

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

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

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

[4] M. H. Browdy. Simulated annealing: An improved computer model for political redistricting. *Yale Law and Policy Review*, 8:163–179, 1990.

[5] R. E. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* SIAM, Philadelphia, 2009.

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

[7] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: Embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–849, 2004.

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

[9] Y. Deng and J. F. Bard. A reactive GRASP with path relinking for capacitated clustering. *Journal of Heuristics*, 17(2):119–152, 2011.

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

[11] J. C. Duque, R. Ramos, and J. Suriñach. Supervised regionalization methods: A survey. *International Regional Science Review*, 30(3):195–220, 2007.

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

[13] H. J. Faria, S. Binato, M. G. C. Resende, and D. J. Falcao. Transmission network design by a greedy randomized adaptive path relinking approach. *IEEE Transactions on Power Systems*, 20(1):43–49, 2005.

[14] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[15] P. Festa and M. G. C. Resende. An annotated bibliography of GRASP – Part I: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009.

[16] P. Festa and M. G. C. Resende. An annotated bibliography of GRASP – Part II: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.

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

[18] F. Glover. Tabu search and adaptive memory programming – Advances, applications, and challenges. In R. S. Barr, R. V. Helgason, and J. L Kennington, editors, *Interfaces in Computer Science and Operations Research*, chapter 1, pages 1–75. Kluwer, Dordrecht, 1996.

[19] F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, 1997.

[20] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39:653–684, 2000.

[21] P. Grilli Di Cortona, C. Manzi, A. Pennisi, F. Ricca, and B. Simeone. *Evaluation and Optimization of Electoral Systems*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia, 1999.

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

[23] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. I: the p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[24] M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11(1):44–52, 1999.

[25] A. Mehrotra, E. L. Johnson, and G. L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1113, 1998.

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

[27] B. Nygreen. European assembly constituencies for Wales – Comparing of methods for solving a political districting problem. *Mathematical Programming*, 42(1–3):159–169, 1988.

[28] M. G. C. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path relinking for the max-min diversity problem. *Computers and Operations Research*, 37(3):498–508, 2010.

[29] M. G. C. Resende and C. C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 8, pages 219–250. Kluwer, Dordrecht, 2003.

[30] M. G. C. Resende and C. C. Ribeiro. GRASP with path-relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, chapter 2, pages 29–63. Springer, Berlin, 2005.

[31] M. G. C. Resende and C. C. Ribeiro. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, chapter 10, pages 283–319. Springer, Berlin, 2010.

[32] M. G. C. Resende, C. C. Ribeiro, F. Glover, and R. Martí. Scatter search and path-relinking: Fundamentals, advances, and applications. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, chapter 4, pages 87–107. Springer, Berlin, 2010.

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

[34] C. C. Ribeiro, E. Uchoa, and R. F. Werneck. A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS Journal on Computing*, 14(3):228–246, 2002.

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

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

[37] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and M. Cabrera-Ríos. New models for commercial territory design. *Networks and Spatial Economics*, 2011. (Forthcoming, doi: 10.1007/s11067-010-9151-6).

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

[39] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, and J. L. González-Velarde. GRASP strategies for a bi-objective commercial territory design problem. *Journal of Heuristics*, 2011. (Forthcoming, doi: 10.1007/s10732-011-9160-8).

[40] M. N. Scott, R. G. Cromley, and E. K. Cromley. Multi-objective analysis of school district regionalization alternatives in Connecticut. *The Professional Geographer*, 48(1):1–14, 1996.

[41] L. Vargas-Suárez, R. Z. Ríos-Mercado, and F. López. Usando GRASP para resolver un problema de definición de territorios de atención comercial. In M. G. Arenas, F. Herrera, M. Lozano, J. J. Merelo, G. Romero, and A. M. Sánchez, editors, *Proceedings of the IV Spanish Conference on Metaheuristics, Evolutionary and Bioinspired Algorithms (MAEB)*, pages 609–617, Granada, Spain, September 2005. In Spanish.

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

[43] A. A. Zoltners and P. Sinha. Sales territory design: Thirty years of modeling and implementation. *Marketing Science*, 24(3):313–331, 2005.