A Dual Bounding Scheme for a Territory Design Problem

Mónica G. Elizondo-Amaya[1]
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
AP 111-F, Cd. Universitaria
San Nicolás de los Garza, Nuevo León 66450, Mexico
moni@yalma.fime.uanl.mx

Roger Z. Ríos-Mercado
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León, Mexico
AP 111-F, Cd. Universitaria
San Nicolás de los Garza, Nuevo León 66450, Mexico
roger.rios@uanl.edu.mx

Juan A. Díaz
Department of Industrial and Mechanical Engineering
Universidad de las Américas Puebla, Mexico
Sta. Catarina Mártir, San Andrés Cholula
Puebla 72820, Mexico
juana.diaz@udlap.mx

September 2012

[1]Corresponding author

Abstract

In this work, we present a dual bounding scheme for a commercial territory design problem. This problem consists of finding a p-partition of a set of geographic units that minimizes a measure of territory dispersion, subject to multiple balance constraints. Dual bounds are obtained using a binary search over a range of coverage distances. For each coverage distance a Lagrangian relaxation of a maximal covering model can be used effectively. Moreover, empirical evidence show that the bounding scheme provides tigher lower bounds than those obtained by the linear programming relaxation. To the best of our knowledge, this is the first study about dual bounds ever derived for a commercial territory design problem.

Keywords: Commercial territory design; Discrete location; Lagrangian relaxation; Dual bounding scheme.

1 Introduction

Territory design can be viewed as the problem of grouping small geographical areas, called *basic areas*, into larger geographic clusters called *territories* according to specific planning criteria. These problems arise in different applications such as political districting (Garfinkel and Nemhauser [16], Hess et al. [22], Hojati [23], Ricca and Simeone [35], Mehrotra et al. [30], Bozkaya et al. [2]) and sales territory design (Shanker et al. [43], Zoltners and Sinha [45, 46], Fleischmann and Paraschis [13], Drexl and Haase [8], Hess and Samuels [21]) to name the most relevant. An extensive survey on general territory design problems and their approaches can be found in Kalcsics et al. [25] and Duque et al. [9].

The problem addressed in this paper is motivated by a concrete practical application from a local beverage firm. To improve customer supply, the company needs to divide the set of city blocks (or basic units) in the city area into a specific number of disjoint territories. In particular, the planning requirements considered in this problem are territory compactness and territory balancing with respect to two activity measures present at every basic unit. The former criterion means that customers within a territory are relatively close to each other while the latter requirement refers to creating territories of about equal size in terms of both number of customers and product demand. This problem can be classified as a comercial territory design problem (TDP) for which related versions under different requirements have been adressed in literature from both exact and heuristic approaches.

Typically, the problem is modeled as minimizing a dispersion measure subject to satisfy some planning requirements such as connectivity and territory balancing. The connectivity requirement implies that basic units (BUs) that are assigned to the same territory must reach each other by travelling within the territory. Depending on how the dispersion measure objective is chosen, we can further clasify these TDP models as p-median TDPs (PMTDP) and p-center TDPs (PCTDP). The former uses a minisum objective function while the latter uses a minimax objective function. Heuristics methods have been developed for both different versions PCTDPs and PMTDPs.

Ríos-Mercado and Fernández [36] introduced the PCTDP subject to connectivity and multiple balancing constraints. They propose a Reactive GRASP to solve the problem. Their proposed approach obtained solutions of much better quality (in terms of dispersion measure and the balancing requirements) than those found by the company method in relatively fast computation times.

Later, Caballero-Hernández et al. [3] study other version of the commercial PCTDP model that includes additional joint assignment constraints which means that some units are required to belong to the same territory. In that work, the authors develop a metaheuristic solution approach based on GRASP. Experimental results show the effectiveness of their method in finding good-quality solutions for instances up to 500 BUs and 10 territories in reasonably short computation times. Particulary, a very good performance is observed within the local search procedure, which produces an improvement of about 90 percent in solution quality. Ríos-Mercado and Salazar-Acosta [38] address an extension of the TDP that considers requirements about design and routing in territories. In contrast to the TDP variations above described, the authors use network-based distances between basic units (instead Euclidean distances) and a diameter-based function to measure territory dispersion. To solve this problem, the authors proposed a GRASP that incorporates advanced features such as adaptive memory and strategic oscillation. Empirical evidence show that the incorporation of these two components into the procedure had a very positive impact on both obtaining feasible solutions and improving solution quality.

Salazar-Aguilar et al. [39] present an exact optimization framework based on branch and bound and cut generation for tackling relatively small instances of several TDP models. Particulary, they studied both, the PCTDP and PMTDP models. They can successfully solve instances of up to 100 BUs for the PCTDP and up to 150 BUs for the PMTDP. The authors also propose new integer quadratic programming TDP models that allowed to eficiently solve larger instances by commercial MINLP solvers such as DICOPT [27]. For IQPs models, they obtained locally optimal solutions for instances with up to 500 BUs and 12 territories. Ríos-Mercado and López-Pérez [37] and López-Pérez and Ríos-Mercado [28] address a commercial TDP with additional side constraints such as disjoint assignment requirements and similarity with existing plan. In their work, they assume a fixed set of centers, and present several heuristic algorithmic strategies for solving the allocation phase.

Recently, a bi-objective TDP model was introduced by Salazar-Aguilar et al. [40], where an ϵ-constraint method is developed for tackling small- to medium- scale instances from an exact optimization perspective. In that work, two different measures of dispersion are studied, one based on the p-center problem objective and the other based on the p-median objective model. It had shown how the latter has a tighter LP relaxation that allowed to solve larger instances. The proposed method was successful for finding optimal Pareto frontiers on instances from 60 up to 150 BUs and 6 territories. It was also clear that larger instances were indeed intractable, thus justifying the use of heuristic approaches proposed by Salazar-Aguilar et al. in [41] and [42]. In these works, the authors address the development of GRASP and Scatter Search (SS) strategies to handle considerably large instances. These proposed heuristic procedures outperformed two of the well-known and most successful multiobjective algorithms in the field, the Non-dominated Sorting Genetic Algorithm (NSGA-II) by Deb et al. [7] and the Scatter Tabu Search Procedure for Multiobjective Optimization (SSPMO) by Molina et al. [32].

As it can be seen, from literature, practically all of the work on commercial territorial design has focused on developing heuristics for finding good feasible solutions to large instances in reasonable times due the well established NP-completness of both PCTDP and PMTDP [36, 39]. However, thus far, the quality of the solutions obtained by these heuristic methods has not been properly evaluated since the quality of the lower bound provided by the linear programming relaxation of TDP models is very poor. Furthermore, there is no other quality reference to acomplish this issue. To the best of our knowledge, no dual bounding schemes have been developed for any of the comercial TDP models found in the literature. It is worth mentioning that besides being usefull in evaluating the quality of heuristic solutions, dual bounds are also the foundations in the development of exact solution methods.

Therefore, the main contribution of this work is the introduction and development of the first dual bounding scheme for a comercial territory design problem. The TDP addressed here considers balance and compactness requirements. This scheme is motivated by exact solution methodologies already found in literature for related location problems, where the main idea is to generate and solve a set of auxiliary problems. Particulary, Albareda-Sambola et al. [1] propose a successful exact solution method for the capacitated p-center problem (CpCP) that involves a procedure for obtaining lower bounds for this problem. The bounding procedure developed in [1] is not quite applicable for our problem; however, given the strong similarities, one of the goals of this paper is to extend this bounding procedure to handle multiple balancing constraints.

The proposed algorithm performs a binary search over a specific set of covering radii extracted from the distances matrix and solves for each of them a Lagrangian dual problem based on a maximal demand covering problem. The evaluation of this dual problem for a given radious δ can determinate, under certain conditions, whenever such covering radious is a dual bound for TDP. An empirical study was carried out on a collection of data instances. The results show the effectiveness of the developed scheme as it considerably outperforms the linear programming relaxation dual bound.

The paper is structured as follows. Section 2 defines the problem formally and describes the mathematical formulation. Section 3 presents the dual bounding scheme and each of its components. Experimental work is included in Section 4. Finally, conclusions and some final remarks are drawn in Section 5.

2 Problem description

Let V be a set of nodes or basic units (BUs) representing city blocks. Let w_i^a be the measure of activity a in block i, $a \in A = \{1, 2\}$ where $a = 1$ denotes number of customers and $a = 2$ denotes product demand. Let d_{ij} be the Euclidean distance between each pair of basic units i and j. The number of territories is given by p. A territory design configuration is a p-partition of the set V. Let $w^a(V_k) = \sum_{i \in V} w_i^a$ the size of territory $V_k \subseteq V$ with respect to activity a. A solution to this problem must have balanced territories with respect to each activity. Due to the discrete nature of the problem and to the unique assignment constraints, it is practically impossible to get perfectly balanced territories. Thus, in order to address this issue, a tolerance parameter τ^a for each activity a is introduced. This tolerance parameter is user specified and it represents a limit on the maximum deviation allowed from an ideal target allowed. This target value is given by the average size $\mu^a = w^a(V)/p$. Finally, in each of the territories, basic units must be relatively close to each other. To account for this, in this work we use a dispersion function based on the p-center problem objective.

All parameters are assumed to be known with certainty. Therefore the problem can be formally described as finding a p-partition of a set V of basic units that meets multiple balance constraints and minimizes a dispersion measure.

2.1 Integer programming formulation

To state the model mathematically, we define the following notation.

Indices and sets

V := set of BUs,

A := set of BUs activities,

i, j := BUs indices; $i, j \in V = \{1, 2, \ldots, n\}$,

a := activity index; $a \in A = \{1, 2\}$.

Parameters

n := number of BUs,

p := number of territories,

w_i^a := value of activity a in node i; $i \in V$, $a \in A$,

d_{ij} := Euclidean distance between i and j; $i, j \in V$,

τ^a := relative tolerance with respect to activity a; $a \in A$, $\tau^a \in [0, 1]$.

μ^a := $w^a(V)/p$, average (target) value of activity a; $a \in A$.

Although the practical decision does not require to place facilities on centers as it is done in locations problems, we used binary decision variables based on centers because they allowed to model territory dispersion appropiately.

Decision variables

$$x_{ij} = \begin{cases} 1 & \text{if BU } j \text{ is assigned to territory with center in BU } i, \\ 0 & \text{otherwise.} \end{cases}$$

With this notation our commercial TDP can be formulated as the following MILP:

$$
\begin{array}{llrcll}
\text{(TDP)} & \text{Minimize} & f(x) & = & \max\limits_{i,j \in V} \{d_{ij} x_{ij}\} & \qquad (1)\\
& \text{subject to} & \sum\limits_{i \in V} x_{ij} & = & 1 \qquad\qquad j \in V, & \qquad (2)\\
& & \sum\limits_{i \in V} x_{ii} & = & p, & \qquad (3)\\
& & \sum\limits_{j \in V} w_j^a x_{ij} & \geq & (1-\tau^a)\mu^a x_{ii} \qquad i \in V; a \in A, & \qquad (4)\\
& & \sum\limits_{j \in V} w_j^a x_{ij} & \leq & (1+\tau^a)\mu^a x_{ii} \qquad i \in V; a \in A, & \qquad (5)\\
& & x_{ij} & \in & \{0,1\} \qquad\qquad i,j \in V. & \qquad (6)
\end{array}
$$

Objective (1) measures territory dispersion. Constraints (2) guarantee that each basic unit j is assigned to only one territory. Constraint (3) assures the creation of exactly p territories. Constraints (4)-(5) represent the territory balance with respect to each activity measure as they ensablish that the size of each territory must lie within a range (measured by a tolerance parameter τ^a) around its average size (μ^a). Moreover, the upper bound balancing constraints (5) also ensure that if no center is placed at i, no customer can be assigned to it (i.e., $x_{ii} = 0 \Rightarrow x_{ij} = 0, \forall\, i, j \in V$). Finally, constraints (6) define the binary nature of the decision variables.

The model can be viewed in terms of integer programming as a vertex p-center problem with multiple capacity constraints (5) and with additional constraints (4). Given that even the uncapacitated vertex p-center problem is NP-hard [26], it follows that our commercial TDP is also NP-hard. Our model is derived from the model introduced by Ríos-Mercado and Fernández [36] that includes additional planning requirements.

3 Dual bound framework

The bounding framework proposed in this work follows the methodology that underlies a wide range of successful exact and approximate solution approaches for p-center problems. These problems are most often solved through generation and solution of a sequence of *auxiliary problems* that keep a strong structural relation with the p-center problem and assure an optimal solution to the original problem. In this case, the use of an auxiliary problem allows achieve the same goal through simplest equivalent formulations. Different auxiliary problems have been proposed, mostly related to coverage problems such as the *set covering problem* (Toregas et al. [44]) and the *maximal covering problem* (Church and ReVelle [4]). Then, successful techniques for p-center problem use a common principle, to perform an iterative search over a range of coverage distances seeking for the smallest radius such that the optimal solution of the associated auxiliary problem provides a feasible solution to the p-center problem. Representative works for uncapacitated p-center problem can be found in Minieka [31], Daskin [5, 6] and Elloumi et al. [10]. For the capacitated version (CpCP), which has been less studied, Özsoy and Pinar [34] and Albareda-Sambola et al. [1] propose exact solution algorithms where the latter presents the best results so far. In [1], they addressed two auxiliary problems (arising from both set and maximal covering problems) and analize two different strategies for solving exactly CpCP, based on binary search and sequential search, respectively. Given that the CpCP is a substructure of the TDP model, this paper exploits the knowledge generated in [1] for deriving dual bounds for the TDP.

In order to introduce the proposed scheme, we highlight the following remarks from the TDP formulation discused in the previous section.

Remarks

- Let $\bar{D} = \{d_0, d_1, \ldots, d_{k_{\max}}\}$ be the set of the $k_{\max}$ different values of the distance matrix $D = (d_{ij})$ sorted by non-decreasing values ($d_0 < d_1 < \ldots < d_{k_{\max}}$), and let $K = \{0, 1, \ldots, k_{\max}\}$ be the corresponding index set in $\bar{D}$. Given the nature of the objective function, which minimizes the maximum distance between a basic unit and the territorial center to which it is assigned, it can be seen that the optimal value of TDP is an element of $\bar{D}$.

- If d_{k^*} is the optimal value of TDP for some index $k^* \in K$, note that any $d_k \in \bar{D}$ with $k \leq k^*$ ($k \geq k^*$) is a lower (upper) bound of the optimal value d_{k^*}.

Therefore, the algorithm relies on an iterative search procedure that attempts to find the best lower (dual) bound by exploring the set of distances in $\bar{D}$. At each iteration, it sets a threshold distance which is used as the coverage radius of an associated covering problem. This auxiliary problem allows to determine when it is not possible to assign all basic units into p or less territories within such radius, yielding therefore a valid dual bound on the optimal value of TDP. In this section we detail the components of this dual bounding procedure.

3.1 The maximum demand covering problem

From the TDP, we derive an auxiliary problem which gives an answer as to whether we can assign all basic units within a certain radius δ into at most p territories, the *maximum demand covering problem* (MDCP$_\delta$). This problem operates with a fixed maximal distance δ known as *covering radius* and considers the objective of maximizing the total amount of covered demand when at most p territorial centers are located. This auxiliary problem can be seen as an extension of a well-known problem from location optimization literature, the maximal covering location problem (MCLP) [4], as we consider additional capacity constraints (4)-(5).

To formulate the model we will use the following additional notation:

$$\begin{aligned}
I_\delta(j) &= \{i \in V : d_{ij} \leq \delta\}, \\
J_\delta(i) &= \{j \in V : d_{ij} \leq \delta\}, \\
b_i^{(\delta,a)} &= \min \left\{ (1+\tau^a)\mu^a, \sum_{j \in J_\delta(i)} w_j^a \right\}.
\end{aligned}$$

where $I_\delta(j)$ denotes the set of territory centers whose distance to basic unit j does not exceed the radius δ. Similary, for a given territory center i, $J_\delta(i)$ denotes the set of basic units whose distance to i does not exceed the radius δ. Additionally, the parameter $b_i^{(\delta,a)}$ has the purpose of strengthening the model since it fits the upper limit of activity measures for territory balance constraints (5). The maximum demand covering problem henceforth denoted as MDCP$_\delta$ can be formulated as follows:

$$\begin{aligned}
(\text{MDCP}_\delta) \quad W(\delta) = \text{Maximize} \qquad f(x) &= \sum_{i \in V} \sum_{j \in J_\delta(i)} w_j^1 x_{ij} && && (7)\\
\text{subject to} \quad \sum_{i \in I_\delta(j)} x_{ij} &\leq 1 && j \in V, && (8)\\
\sum_{i \in V} x_{ii} &\leq p, && && (9)\\
\sum_{j \in J_\delta(i)} w_j^a x_{ij} &\geq (1-\tau^a)\mu^a x_{ii} && i \in V; a \in A, && (10)\\
4mm] \qquad \sum_{j \in J_\delta(i)} w_j^a x_{ij} &\leq b_i^{(\delta,a)} x_{ii} && i \in V; a \in A, && (11)\\
x_{ij} &\in \{0,1\} && i \in V; j \in J_\delta(i). && (12)
\end{aligned}$$

The objective function (7) maximizes the total amount of demand or product demand (i.e., activity measure $a = 1$) that can be covered. By contraints (8) each customer is assigned to at most one territory. As in TDP model, constraints (10)-(11) conform the terrritory balance constraints, which are referred to as *constraints of minimal and maximal territorial capacity*, respectively. In particular, constraints (11) also guarantee that if no center is placed at i, no customer can be assigned to it. Finally, constraint (9) assure the creation of at most p territories. Then, the maximum demand covering problem consists in maximizing the total demand of basic units that can be satisfied whit at most p territories within a given maximal assignment distance δ.

We investigate now the relation between TDP and MDCP$_\delta$. Let $W_{\text{tot}} = \sum_{j \in V} w_j^1$ be the sum of demand corresponding to activity measure $a = 1$ (i.e., product demand) over all basic units. When solving MDCP$_\delta$ we have the following cases.

Case 1: If for some $k \in K$, the total demand that can be satisfied within a radius $d_k \in \bar{D}$ is at least W_{tot} and p territory centers are selected, then all BUs have been assigned and moreover, the assignment obtained from MDCP_δ is a fasible solution for TDP. Therefore, the radius d_k is a valid upper bound on the optimal value of TDP.

Case 2: The optimal solution to TDP can be obtained by finding the smallest index $k \in K$ such that $W(d_k) = W_{\text{tot}}$. Note that for this case, the number of territories that are generated is allways p as the number of territories required to cover the maximal amount of demand increases when the coverage radius decreases.

Case 3: If for some $k \in K$, $W(d_k) < W_{\text{tot}}$, it can be seen that it is not possible to assign all BUs within such covering radius and therefore the radius d_k is a valid lower bound on the optimal value of TDP.

An advantage of MDCP_δ, is that its objective funcion $W(\delta)$, provides either a bound (dual or primal) or the optimal value for the TDP, depending on the number of basic units that were assigned in the MDCP_δ optimal solution. Also note that, without loss of generality, activity 2 can be alternatively used instead of activity 1 in the objective function $W(\delta)$ and by using $W^2_{\text{tot}} = \sum_{j \in V} w^2_j$ the just described cases still apply.

Given that MCLP is NP-hard [29], it follows that MDCP_δ is also NP-hard. Exact solution methods developed for MCLP are not applicable to MDCP_δ unless they are adapted to handle its specific features. Moreover, even medium size instances of the problem addressed in this work are practically intractable by such solution techniques. Therefore, instead of solving MDCP_δ exactly, a Lagrangian relaxation is considered to obtain a valid upper bound for MDCP_δ from which the following statement can be validated:

Proposition 3.1. *Let $\bar{W}(\delta)$ be an upper bound for $MDCP_\delta$, if $\bar{W}(\delta) \leq W_{tot}$, then the coverage radius δ is a valid lower bound on the optimal value of TDP.*

Proof. Let X_δ be the optimal solution to MDCP_δ with corresponding optimal objective function value given by $W(\delta)$. It is easy to check that $W(d_0) \leq W(d_1) \ldots \leq W(d_{k_{\max}})$, where $d_k \in \bar{D}$, $k \in K$. Now we make more precise relation between the optimal solutions of problems MDCP_δ and TDP.

Let k^* be the smallest index $k \in K$ such that $W(d_{k^*}) = W_{\text{tot}}$, then the $\text{MDCP}_{d_{k^*}}$ optimal solution $X_{d_{k^*}}$ is also the optimal solution for the TDP with optimal value d_{k^*}. Note that territory balance constraints are also present in the MDCP_δ formulation. On the other hand, constraints (8) and (9) in MDCP_δ ensure that each BU is assigned to at most one territory center and the creation of at most p territories, respectively. However, for the optimal solution $X_{d_{k^*}} = \left(x^*_{ij}\right)$ constraints (2) are satisfied since:

$$\sum_{i \in V} \sum_{j \in J_{d_{k^*}}(i)} w^1_j x^*_{ij} = \sum_{j \in V} \left(w^1_j \sum_{i \in I_{d_{k^*}}(j)} x^*_{ij} \right) = W_{\text{tot}}$$

$$\Rightarrow \sum_{j\in V}\left(w_j^1 \sum_{i\in I_{d_{k^*}}(j)} x^*_{ij}\right) = \sum_{j\in V} w_j^1$$

$$\Rightarrow \sum_{i\in I_{d_{k^*}}(j)} x^*_{ij} = 1$$

Notice that $x_{ij} = 0$, $\forall i \notin I_{d_{k^*}}(j)$, then we have that:

$$\sum_{i\in V} x^*_{ij} = \sum_{i\in I_{d_{k^*}}(j)} x^*_{ij} = 1$$

$$\Rightarrow \sum_{i\in V} x^*_{ij} = 1$$

Therefore, TDP constraint of unique assignment (2) is satisfied by $X_{d_{k^*}}$ and all BUs have been assigned. On the other hand, it is easy to see that constraint (3) is also satisfied by $X_{d_{k^*}}$ since the number of territories required to cover all BUs demand tends to increase when the coverage radius decreases. Since d_{k^*} is the smallest coverage radius in $\bar{D}$ for which all TDP constraints can be satisfied, it follows that d_{k^*} is the optimal value of TDP and $X_{d_{k^*}}$ its optimal solution.

Finally, it can be noticed that for all $k \in K$ such that $k \leq k^*$, the radius d_{k^*} is a valid lower bound for TDP and further, $W(d_k) \leq W_{\text{tot}}, \forall k \leq k^*, k \in K$. Notice that in the general case, as $\bar{W}(\delta)$ is an upper bound on the optimal value of MDCP$_\delta$, if $\bar{W}(\delta) \leq W_{\text{tot}}$ implies that $W(\delta) \leq W_{\text{tot}}$ and case 3 holds for any TDP relaxation. □

Next, we detail the relaxation of MDCP$_\delta$ used in order to obtain the upper bound $\bar{W}(\delta)$, for a given coverage radius δ.

3.2 Lagrangian relaxation of MDCP$_\delta$

In this section we propose a relaxation of MDCP$_\delta$ which consists of relaxing the assignment constraints (8) in a Lagrangian fashion, i.e., incorporating them to the objective function with the additional multipliers $\lambda \in \mathbb{R}_+^{|V|}$. For surveys of Lagrangian relaxation, the reader is refered to Guignard [18], Geoffrion [17] and Fisher [11, 12]. Then, the resulting model is:

$$(L_\delta(\lambda)) \quad \text{Maximize} \quad Z_{LR}(\lambda) = \sum_{i\in V}\sum_{j\in J(\delta)} w_j^{a^*} x_{ij} + \sum_{j\in V}\lambda_j\left(1 - \sum_{i\in I(\delta)} x_{ij}\right) = \sum_{j\in V}\lambda_j$$

$$
\begin{array}{rrcll}
 & & & + \max \left\{ \sum_{i \in V} \sum_{j \in J(\delta)} (w_j^{a^*} - \lambda_j) x_{ij} \right\} & \\
\text{subject to} & \sum_{i \in V} x_{ii} & \leq & p, & \\
 & \sum_{j \in J_\delta(i)} w_j^a x_{ij} & \geq & (1-\tau^a)\mu^a x_{ii} & i \in V; a \in A, \\
 & \sum_{j \in J_\delta(i)} w_j^a x_{ij} & \leq & b_i^{(\delta,a)} x_{ii} & i \in V; a \in A, \\
 & x_{ij} & \in & \{0,1\} & i \in V; j \in J_\delta(i).
\end{array}
$$

Then, the Lagrangian problem $L_\delta(\lambda)$ consists of maximizing a weighted sum over the variables $x_{ij}, i, j \in V$, under constraints of minimal and maximal territory capacity and the selection of p territorial centers. Notice that the model $L_\delta(\lambda)$ can be decomposed into $|V|$ independient subproblems, one for each $i \in V$, as follows:

$$
\begin{array}{llrcll}
(\mathrm{TSKP}_i) & \text{Maximize} & v_i(\lambda, x) & = & \sum_{j \in J_\delta(i)} (w_j^{a^*} - \lambda_j) x_{ij} & \qquad (13) \\
 & \text{subject to} & \sum_{j \in J_\delta(i)} w_j^a x_{ij} & \geq & (1-\tau^a)\mu^a x_{ii} \quad a \in A, & \qquad (14) \\
 & & \sum_{j \in J_\delta(i)} w_j^a x_{ij} & \leq & b_i^{(\delta,a)} x_{ii} \quad a \in A, & \qquad (15) \\
 & & x_{ij} & \in & \{0,1\} \quad j \in J_\delta(i). & \qquad (16)
\end{array}
$$

Each of these subproblems can be seen as a knapsack problem with double constraints of minimal and maximal capacity, or as a bidimensional knapsack problem with additional constraints (14). We denote this subproblem as *Two-Sided Knapsack Problem* (TSKP). Hence, to solve $L_\delta(\lambda)$, for each $i \in V$ its corresponding subproblem TSKP_i is solved. Then, in order to meet constraint (9), the indices in the set V are sorted in non-increasing values of TSKP_i, that is,

$$
v_{i_1}(\lambda, x) \geq v_{i_2}(\lambda, x) \geq \ldots \geq v_{i_{|V|}}(\lambda, x).
$$

Then, the first p^* indices are choosen as territorial centers, where p^* is given as follows,

$$
p^* = \min \{ p, \max\{ r \, : \, v_{j_r}(\lambda, x) > 0 \} \}.
$$

The idea behind this is to choose the indices with the best evaluation of its corresponding subproblem $\mathrm{TSKP}_i(\lambda)$. Therefore, the optimal solution of $L_\delta(\lambda)$ consists of the territories with center in $\{i_1, i_2, \ldots, i_{p^*}\}$ and the assignments of the BUs to these territories given by the solution of the p^* associated subproblems $\mathrm{TSKP}_i(\lambda)$.

Thus, for a given vector of multipliers $\lambda \in \mathbb{R}_+^{|V|}$, an upper bound for MDCP$_\delta$ is computed by means of the procedure described above. As it is well known, the best Lagrangian bound is obtained by solving the Lagrangian dual problem,

$$(\text{LD}_\delta) \qquad \bar{W}(\delta) = \min_{\lambda \in \mathbb{R}_+^{|V|}} L_\delta(\lambda).$$

which is solved using subgradient optimization.

Algorithm 1 Subgradient optimization procedure

Input: P:= A TDP instance;
  δ:= Covering radius;
  T:= Stopping criteria;
  t:= Number of iterations without improvement after which the parameter α is halved;
Output: $\bar{W}(\delta)$:= Best upper bound for MDCP$_\delta$;

```
η_lb ← −∞   η_k ← +∞;
λ_j^0 ← random[0, 10];   j ∈ V;
k ← 0;
count ← 0;
Terminate ← false;
while (not Terminate) do
    Solve L_δ(λ^k);
    if (L_δ(λ^k) < η_k) then
        η_k ← L_δ(λ^k);
    else
        count ← count + 1;
        if (count = t) then
            α ← α/2;
            count ← 0;
        end if
    end if
    Apply the primal heuristic to obtain a lower bound lb;
    if (lb > η_lb) then
        η_lb ← lb;
    end if
    s_j^k ← (1 − Σ_{i∈I_δ(j)} x_ij^k);   j ∈ V;
    θ_k ← α_k(η_k − η_lb) / ‖s^k‖^2;
    λ_j^{k+1} ← max{0, λ_j^k − θ_k s_j^k};   j ∈ V;
    k ← k + 1;
    if (Stopping criteria T is not satisfied) then
        Terminate ← true;
    end if
end while
W̄(δ) ← η_(k−1);
return W̄(δ);
```

3.3 Subgradient optimization algorithm

In this phase a classical subgradient optimization is performed [19, 20]. Given an initial vector λ^0, a sequence $\{\lambda^k\}$ is generated by the rule

$$\lambda_j^{k+1} = \max\{0, \lambda_j^k - \theta_k s_j^k\}, \quad j = 1, \dots, n.$$

where s^k is a subgradient at $\lambda = \lambda^k$ and $\theta_k > 0$ is the step size, calculated through the commonly used formula

$$\theta_k = \frac{\alpha_k(\eta^k - \eta_{lb})}{\|s^k\|^2},$$

with α_k being a scalar satisfying $0 < \alpha_k \leq 2$. In practice, this parameter is initialized to $\alpha_0 = 2$ and its value is halved if the upper bound fails to improve after a determinated number of consecutive iterations; η^k is the upper bound at iteration k; η_{lb} is the lower bound available at iteration k usually obtained by applying a primal heuristic for MDCP$_\delta$.

The subgradient vector at iteration k is given by $s^k = \left[s_j^k\right]$, with

$$s_j^k = 1 - \sum_{i \in I_\delta(j)} x_{ij}^* \quad j \in V,$$

where x_{ij}^* is the solution of Lagrangian problem $L_\delta(\lambda)$.

In practice [14, 15, 33], the multipliers vector $\lambda \in \mathbb{R}_+^{|V|}$ is commonly initialized with random values in the range [0,10], while the stopping criteria are the following:

- $\theta \leq 0.00001$
- $\alpha \leq 0.00001$
- $\eta_k - \eta_{lb} < 1$
- If $\lfloor \eta_k \rfloor$ fails to improve after m consecutive iterations.
- Maximum number of iterations.

A summary of the subgradient procedure implemented is depicted in Algoritm 1.

3.3.1 Primal heuristic

Note that, given a vector of multpliers $\lambda \in \mathbb{R}_+^{|V|}$, the solution of $L_\delta(\lambda)$ may not be feasible for MDCP$_\delta$. Since single assignment constraints are relaxed, an $L_\delta(\lambda)$ solution may present multiple assignments of the basic units to the territories whereas other basic units could not been assigned to any territory. Therefore, at the inner iterations of subgradient optimization, primal bounds to MDCP$_\delta$ are heuristically built from $L_\delta(\lambda)$ by repairing infeasibility through the following steps.

Algorithm 2 Primal Heuristic

Input: P:= A TDP instance;
  δ:= Covering radius;
  $I^L = \{i_1, i_2, \ldots, i_{p^*}\}$:= Set of territory centers selected in the Lagrangian solution;
  $X^L = \left\{X_{c(i_1)}, X_{c(i_2)}, \ldots \; X_{c(i_p^*)}\right\}$:= Solution of Lagrangian problem $\mathrm{L}_\delta(\lambda)$;
  U:= Set of unassigned BUs in the solution of $\mathrm{L}_\delta(\lambda)$;

Output: X^f:= Feasible solution (lower bound) for MDCP_δ;

```
X^f ← φ;
X_c(i) := Territory with center in i ∈ V;
for all j ∈ V do
    I_j^L ← {i ∈ V : x_ij = 1};
    if (|I_j^L| > 1) then
        for all i ∈ I_j do
            f_i ← min_{a∈A} {w^a(X_c(i)) − w_j^a − (1 − τ^a)μ^a};
        end for
        if (min_{i∈I_j^L} {f_i} > 0) then
            i* ← arg min_{i∈I_j^L} {f_i};
            for all i ∈ I_j^L such that i ≠ i* do
                X_c(i) ← X_c(i) \ {j};
            end for
        else
            i* ← arg max_{i∈I_j^L} {f_i : f_i < 0};
            for all i ∈ I_j^L such that i ≠ i* do
                X_c(i) ← X_c(i) \ {j};
            end for
        end if
    end if
end for
for all j ∈ U do
    for all i ∈ I do
        r_i ← max_{a∈A} {b_i^(δ,a) − w^a(X_c(i))};
    end for
    i* ← arg max_{i∈I^L} {r_i : w_j^a ≤ b_i^(δ,a) − w^a(X_c(i)) ∧ d_ij ≤ δ,  a ∈ A};
    X_c(i*) ← X_c(i*) ∪ {j};
end for
for all i ∈ I^L do
    if (w^a(X_c(i)) ≥ (1 − τ^a)μ^a,  a ∈ A) then
        X^f ← X^f ∪ X_c(i);
    end if
end for
return X^f;
```

1. This stage eliminates the multiple assignments of BUs (if they exist) by considering the unbalances (with respect to each activity measure) that produces the removal of BUs from the territories. Let $X^L = (X_{c(i_1)}, X_{c(i_2)} \ldots, X_{c(i_p^*)})$ be the optimal solution of $L_\delta(\lambda)$, $\lambda \in \mathbb{R}_+^{|V|}$, where $X_{c(i)}$ represents the set of BUs that belong to territory with center in $i \in V$ and let $I^L = (i_1, i_2, \ldots, i_p^*)$ be the set of territorial centers selected in the Lagrangian solution.

   For each $j \in V$, the set I_j^L denotes the territory centers associated to basic unit j, i.e., $I_j^L = \{i \in I^L : x_{ij} = 1\}$. If $\left|I_j^L\right| > 1$, which means that basic unit j has been assigned to more than one territory, a function f_i is evaluated for each $i \in I_j^L$. This function quantifies the impact on the feasibility with respect to constraints (10), when the basic unit j is subtracted from the territory i and is calculated as follows:

$$f_i = \min_{a \in A} \left\{ w^a(X_{c(i)}) - w_j^a - (1 - \tau^a)\mu^a \right\}, \tag{17}$$

   where $w^a(X_{c(i)}) = \sum_{j \in X_{c(i)}} w_j^a$ is the size of the territory $X_{c(i)}$ with respect to the activity $a \in A$, while w_j^a, τ^a and μ^a are parameters of TDP model described in Section 2. The territory that keeps the basic unit j is selected under the following criteria:

   - If $\min_{i \in I_j^L}\{f_i\} \geq 0$, it means that each territory is feasible when BU $j \in V$ is eliminated from $X_{c(i)}$ and therefore, the territory with the lowest evaluation in function (17) is selected to keep the basic unit j, i.e., $i^* = \arg\min_{i \in I_j}\{f_i\}$.

   - If $\min_{i \in I_j^L}\{f_i : f_i < 0\}$, it means that at least one territory becomes infeasible with respect to the minimal activity size $(1 - \tau^a)\mu^a$ for some activity $a \in A$. Notice that when assigning the basic unit j to a single territory center from I_j^L, those territories that do not satisfiy balancing constraints (10) are not considered in the primal solution of MDCP_δ since they become infeasibles when j is removed from them. Then, is convenient to select the territory that provides the greatest covered demand among those territories for which $f_i > 0$ to keep the basic unit j, i.e., $i^* = \arg\max_{i \in I_j^L}\{f_i : f_i < 0\}$.

2. Once that multiple assignment have been eliminated, we have a feasible solution for MDCP_δ by considering only those territories that meet balance contraints (10). It can be noticed that by subtracting the basic units from territories in the previous phase, it may be that some of them are unbalanced with respect to some activity measure, that is, its size could be less than $(1 - \tau^a)\mu^a$ for some activity $a \in A$ and therefore, such territories could not be included in a feasible solution for MDCP_δ. Additionaly, there may be basic units unassigned from the Lagrangian problem resolution. Hence, a second phase improves the actual feasible solution

and tries to recover feasibility of those territories that do not satisfy balance constraint (10) by assigning the maximum possible number of basic units to them as follows.

- Let U be the set of unassigned basic units in the Lagrangian solution. The idea at each iteration of this stage is to assign each $j \in U$ to that territory with the highest residual capacity among both activity measures. This is performed through the next steps:
  - Territories $X_{c(1)}, \dots, X_{c(p^*)}$ are ranked by non-increasing order according to their residual capacity denoted as r_i, which is calculated as follows:

$$r_i = \max_{a \in A} \left\{ b_i^{(\delta,a)} - w^a(X_{c(i)}) \right\},$$

    being $\{X_{c(i_1)}, X_{c(i_2)} \dots, X_{c(i_{p^*})}\}$ the ordered set in such a way that $r_{(i_1)} \geq r_{(i_2)} \geq \dots \geq r_{(i_p^*)}$.
  - Basic unit j is assigned to the territory in the ordered set with the lowest index i^* that satisfy:

$$w_j^a \leq b_{i^*}^{(\delta,a)} - w^a(X_{c(i^*)}), \quad a \in A. \tag{18}$$

    Relationship (18) assures the compliance of constraints (11). If there is no territory with these characteristics, the basic unit j is not assigned.

At the end of the primal heuristic, we have a feasible solution and therefore, a primal bound to the MDCP$_\delta$, which may sometimes be feasible even for the TDP in the case that all BUs are assigned to exactly p territories which satisfy balance constraints (10). Algorithm 2 summarizes the primal heuristic above described.

3.4 The dual bounding scheme

In this section we present the bounding scheme for the TDP. The idea underlying this procedure is to carry out a search among the elements of the set $\bar{D}$ associated with the distance matrix in order to find the best lower (dual) bound on the optimal value of TDP. The procedure solves a series of Lagrangian duals $\bar{W}(d_k)$ and seeks for the maximal coverage radious d_k^* that statisfy the conditions of Proposition 3.1, thus obtaining the best dual bound from the covering radii candidates.

The proposed LB scheme is based on binary search over the set $\bar{D}$. As a preprocesing step, this set $\bar{D}$ can be further reduce by following test.

- *Elimination by lower bound*: If LB is a valid lower bound for TDP, then the set $\{d_0, d_1, \dots, d_{k_l}\}$, where $k_l \in K$ is the largest index such that $d_{k_l} < LB$, can be discarded.
- *Elimination by upper bound*: If UB is a valid upper bound for TDP, then the set $\{d_{k_u}, d_{k_u+1}, \dots, d_{k_{\max}}\}$, where $k_u \in K$ is the smallest index such that $d_{k_u} > UB$ can be

discarded.

Algorithm 3 summarizes the dual bounding scheme for TDP, which is referred to as DBS.

Algorithm 3 Dual bounding scheme (DBS)

Input: P:= A TDP instance;
  $\bar{D} = \{d_0, d_1, \ldots, d_{k_{\max}}\}$:= Ordered set of covering radii;
Output: LB:= Lower (dual) bound on the optimal value of TDP;
  $a \leftarrow 1$;
  $b \leftarrow k_{\max}$;
  while $(a < b)$ **do**
    $k \leftarrow \left\lfloor \frac{(a+b)}{2} \right\rfloor$;
    Solve LD_{d_k} and evaluate $\bar{W}(d_k)$;
    if $(\bar{W}(d_k) < W_{tot})$ **then**
      $a \leftarrow k + 1$;
    else
      $b \leftarrow k - 1$;
    end if
  end while
  LB$\leftarrow d_a$;
  return LB;

3.5 Pre-processing for DBS

In this section, a pre-processing phase which significantly reduces the computational effort of the binary search by obtaining both initial lower and upper bounds is developed. In addition to this, a relative tolerance ϵ for the size of the exploring interval is used.

To obtain an initial lower bound, a sequential search among the set $\bar{D}$ is performed which solves, at each iteration, the following relaxation of MDCP_δ,

$$
\begin{aligned}
(\mathrm{MDCP}_\delta\text{-R}) \quad \phi(\delta, x) = \quad & \text{Maximize} f(x) = \sum_{i \in V} \sum_{j \in J_\delta(i)} w_j^1 x_{ij}, && (19)\\
\text{subject to} \quad & \sum_{i \in V} x_{ii} \leq p, && (20)\\
& x_{ij} \in \{0,1\}\ i \in V; j \in J_\delta(i). && (21)
\end{aligned}
$$

Once again, it can be noticed that MDCP_δ-R is separable in the set V and it can be easily solved by calculating for each $i \in V$ the maximum demand $c_i(\delta)$ that can be covered from i within a radius δ as follows,

$$c_i(\delta) = \sum_{j \in J_\delta(i)} w_j^1.$$

Finally, to satisfy constraint (20), the indices in V are sorted by non-decreasing order of the values $c_i(\delta)$ and the first p indices are choosen to calculate the amount of effective demand $C_{ef}(\delta)$ that can be covered by p territories within a maximum distance δ, this is:

$$C_{ef}(\delta) = \sum_{r=0}^{p} c_{i_r}.$$

Therefore, the optimal value of MDCP$_\delta$-R is given by $C_{ef}(\delta)$ which, at the same time, is an upper bound for MDCP$_\delta$. Then, using Proposition 3.1 we determine if δ is a valid lower bound for the TDP. The purpose of the sequential search is therefore to find the best initial lower bound (i.e., the largest covering radius for wich $C_{ef}(\delta) \leq W_{tot}$). The procedure for solving MDCP$_\delta$-R is outlined in Algorithm 4.

Algorithm 4 pre_processing $(P, \bar{D})$

Input: P:= A TDP instance;
  $\bar{D} = \{d_0, d_1, \ldots, d_{k_{\max}}\}$:= Ordered set of covering radii;
Output: k_1:= Index of the initial upper bound d_{k_1};

```
t ← 0;
δ ← d_t;
C_ef(δ) ← 0;
c_i(δ) ← 0;  ∀i ∈ V;
while (C_ef(δ) ≤ W_tot) do
    for all i ∈ V do
        c_i(δ) ← Σ_{j∈J_δ(i)} w_j^1;
    end for
    Order the indices in V in such a way that c_{i_1}(δ) ≥ ... ≥ c_{i_|V|}(δ);
    C_ef(δ) ← Σ_{r=0}^{p} c_{i_r}(δ);
    t ← t + 1;
    δ ← d_t;
end while
k_1 ← t − 1;
return k_1;
```

A valid initial upper bound for TDP is obtained from a known heuristic [36]. As it was described in Section 1, the authors propose a reactive GRASP (R-GRASP) for a similar model (TDP-C) with the only difference that they consider additional conectivity constraints. Hence, TDP model can be seen as a relaxation of TDP-C as the first model relaxes the conectivity constraints and therefore an upper bound for TDP-C is also an upper bound for TDP (i.e., $z^*(\text{TDP}) \leq z^*(\text{TDP-C}) \leq ub$, where $z^*(\text{Y})$ is the optimal value of problem Y).

Algorithm 5 states the dual bounding scheme DBS_P, which incorporates the above mentioned components in order to speed up the DBS procedure.

Algorithm 5 DBS_P($P, \bar{D}$)

Input: P:= A TDP instance;
$\bar{D} = \{d_0, d_1, \ldots, d_{k_{\max}}\}$:= Ordered set of covering radii;
Output: LB:= Lower (dual) bound on the optimal value of TDP;

```
k_1 := pre_processing(); {Compute initial lower bound d_{k_1}}
k_2 := R-GRASP();  {Compute initial upper bound d_{k_2}}
a ← k_1;
b ← k_2;
while ((d_b − d_a)/d_a ≥ ε) do
    k ← ⌊(a+b)/2⌋;
    Solve LD_{d_k} and evaluate W̄(d_k);
    if (W̄(d_k) < W_tot) then
        a ← k + 1;
    else
        b ← k;
    end if
end while
LB ← d_a;
return LB;
```

4 Computational evaluation

In this section, we provide computational results for the dual bounding scheme we developed for the TDP. Our overall objective is to assess if DBS is a promising methodology for TDP. More specifically, the following issues are studied:

(1) The effect of the pre-processing stage (providing both dual and primal bounds).

(2) A comparison of the proposed bounding scheme with the LP relaxation.

(3) Optimal solutions are known for medium size instances. The quality of the DBS_P bounds is then assessed for these instances.

All the procedures have been coded in C++ and compiled with the Sun C++ 8.0 compiler. The experimental work was carried out on a SunFire V440 computer under Solaris 9 operating system. CPLEX 11.2 callable libraries [24] were used to solve subproblems TSKP_i.

Randomly generated instances based on real-world data on planar graphs provided by the industrial partner were used. This data set is taken from [36]. In that work, full details on how the instances are generated can be found. A tolerance $\tau^a = 0.05, a \in A$, with respect to each activity measure was considered. The particular characteristics of the instances used are described in each experiment.

In regard to the subgradient procedure for solving $L_\delta(\lambda)$, the algorithmic rules that were considered are the following:

- Start with $\alpha = 2$ and halve its value if the dual bound fails to improve after 15 consecutive iterations.

- Stopping criteria:

  - Maximum iteration number (600 iterations).
  - If the current absolute value of the currrent difference between the upper and lower bounds is less than one unit (i.e., $ub - lb < 1$). As MDCP$_\delta$ is an integer programming problem, a difference less than one indicates that optimality has been achieved since the decision variables coefficients in the objective function are integer-valued. The optimal solution for the problem is given by the current lower bound.
  - If $\lambda_i = 0, \forall i \in V$. An optimal solution for LD$_\delta$ has been obtained, but a duality gap may exist. The best available solution is given by the current lower bound.
  - If $lb = W_{\text{tot}}$. The total assignment of BUs has been achieved by a primal solution of MDCP$_\delta$ then, a feasible solution or upper bound for TDP has been found.
  - If $ub < W_{\text{tot}}$. Preposition 3.1 is met and a valid lower bound for TDP has been found.
  - If $\lfloor ub \rfloor$ fails to improve after 30 consecutive iterations.
  - If $\theta \leq 0.00001$. A duality gap exists and the best available solution is given by the current lower bound wich is provided by the primal heuristic.

4.1 Comparing DBS and DBS_P

The improvement produced when a pre-processing is applied to DBS is first addressed. As stated in Section 3.5, initial upper and lower bounds are easily generated to reduce the initial set of coverage radii to be explored. In addition, the binary search procedure is executed until a relative gap ϵ (i.e. percentage difference) between the lowest and greatest values in the set of candidate radii is reached. In order to balance the tradeoff between solution time and quality we set $\epsilon = 0.001$ (i.e., 0.1%) in our computational study.

Three instance sets defined by $(n, p) \in \{(60, 4), (100, 6), (500, 10)\}$ were generated. For each of these sets, 15 different instances were generated and tested using both binary search schemes. Table 1 compares DBS and DBS_P. The first column indicates the instance size tested. The second and third column display the average CPU time required per instance under each scheme (time required for obtaining initial (lower and upper) bounds for TDP is also included). The fourth column shows the percentage reduction by DBS_P on the total execution time. Similary, the last three columns show the information about the number of radii that were tested.

Results in Table 1 indicate that modified binary search DBS_P has a signicant impact in the execution times, which are reduced up to 74.1%. It can be noticed, that this improvement relies on the number of explored radii, which reaches a decrease of over 50% using pre-processing on tested instances. Although the relative deviation ϵ used in DBS_P can be seen as a pay off for this time improvement, this parameter was fixed considering the order of magnitude of the elements in $\bar{D}$, in such a way that the deterioration of the DBS bound was not significant. It can be concluded that providing initial upper and lower bounds as a preprocessing strategy pays off as a considerable less amount of computational effort is needed. Therefore, is really worthwhile to include the preprocessing procedure within the bounding scheme.

Size	Time (sec)			Explored radii		
(n,p)	DBS	DBS_P	Improvement (%)	DBS	DBS_P	Improvement (%)
(60, 4)	1306.39	513.67	60.7	12	6	50.0
(100, 6)	2812.29	694.75	63.3	12	6	50.0
(500, 10)	11811.97	3058.41	74.1	17	6	64.7

Table 1: Performance of DBS and DBS_P procedures.

4.2 Evaluation of DBS_P bounds

This part of the work is focused on the study of the quality of the obtained bounds. As it was mentioned before, this dual bounding scheme is the first known to date for commercial territory design. For this reason, we make a comparison with bounds based on the LP relaxation. Additionally, the DBS_P bounds are compared with respect to optimal solutions for medium size instances (60 and 100 BUs instances).

4.2.1 Comparison with the LP relaxation

A comparison between DBS_P and the LP relaxation (LPR) lower bounds for TDP is carried out. A set of 30 instances of each size $(n,p) \in \{(500,10),(1000,20),(2000,20)\}$ was tested. First, for solving the LP relaxation, there are several methods available through CPLEX, the most commonly used is the Primal Simplex algorithm, however, we also tested Barrier, Sifting, Dual Simplex and Network Simplex algorithms on 20 500-nodes instances to investigate the computational effort.

As Figure 1 indicates, the most efficient method was the Sifting Algorithm (SA). This method was developed to exploit the characteristics of models with large aspect ratios (that is, a large ratio of the number of columns with respect to the number of rows). In particular, the method is well suited to large aspect ratio models where an optimal solution can be expected to place most variables at their lower bounds. Sifting solves a sequence of LP subproblems where the results from one subproblem are used to select columns from the original model for inclusion in the next subproblem. It starts by solving a subproblem (known as the working problem) consisting of all rows but only a small subset of the full set of columns, by assuming an arbitrary value (such as its lower bound) for the solution value of each of the remaining columns. This solution is then used to

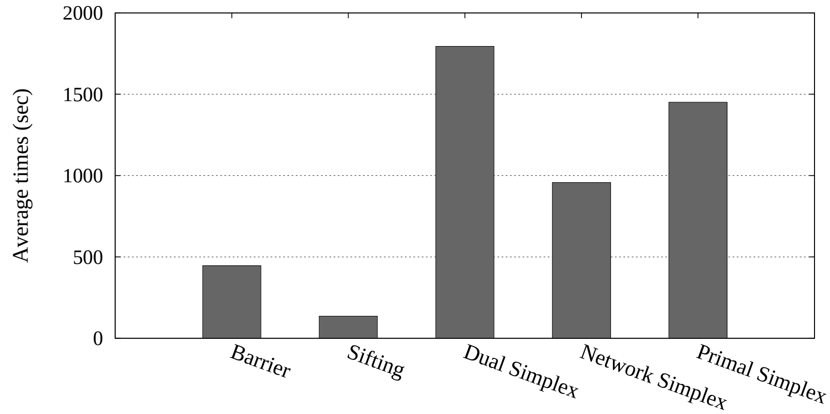

Figure 1: Performance of LP methods for solving the TDP linear relaxation.

re-evaluate the reduced costs of the remaining columns. Any columns whose reduced costs violate the optimality criterion become candidates to be added to the working problem for the next major sifting iteration. When no candidates are present, the solution of the working problem is optimal for the full problem, and sifting terminates (see [24]). Thus, for remainder we use the SA for solving the LP relaxation on this TDP.

Size	RD	Time (sec)	
(n, p)	(%)	LPR	DBS_P
(500,10)	252.46	148.9	2352.4
(1000,20)	259.03	1028.1	5719.8
(2000,20)	346.16	6728.1	13548.3

Table 2: Comparison of LPR and DBS_P bounding schemes.

Results of the empirical comparison are summarized in Table 2 where the first column indicates the instance size, the second column displays the average relative deviation (RD) between the DBS_P and LPR bounds, and the third and fourth columns show the average running times for both LPR and DBS_P bounding schemes respectively. This gap represents the relative improvement of the bound provided by the dual bounding scheme (lb(DBS_P)) with respect to the bound obtained by the linear programming relaxation (lb(LPR)). It is computed as:

$$\mathrm{RD} = 100 \left(\frac{lb(\mathrm{DBS\_P}) - lb(\mathrm{LPR})}{lb(\mathrm{LPR})} \right).$$

As it can be observed, the average computation times of DBS_P are significantly larger than those reported by the resolution of the linear problem. However, it is notable that this effort invested by DBS yields a significant improvement over the LPR bound as it provides a considerably better quality than those reported by the LP relaxation. The average RD ranges from 252.46% to 346.16%

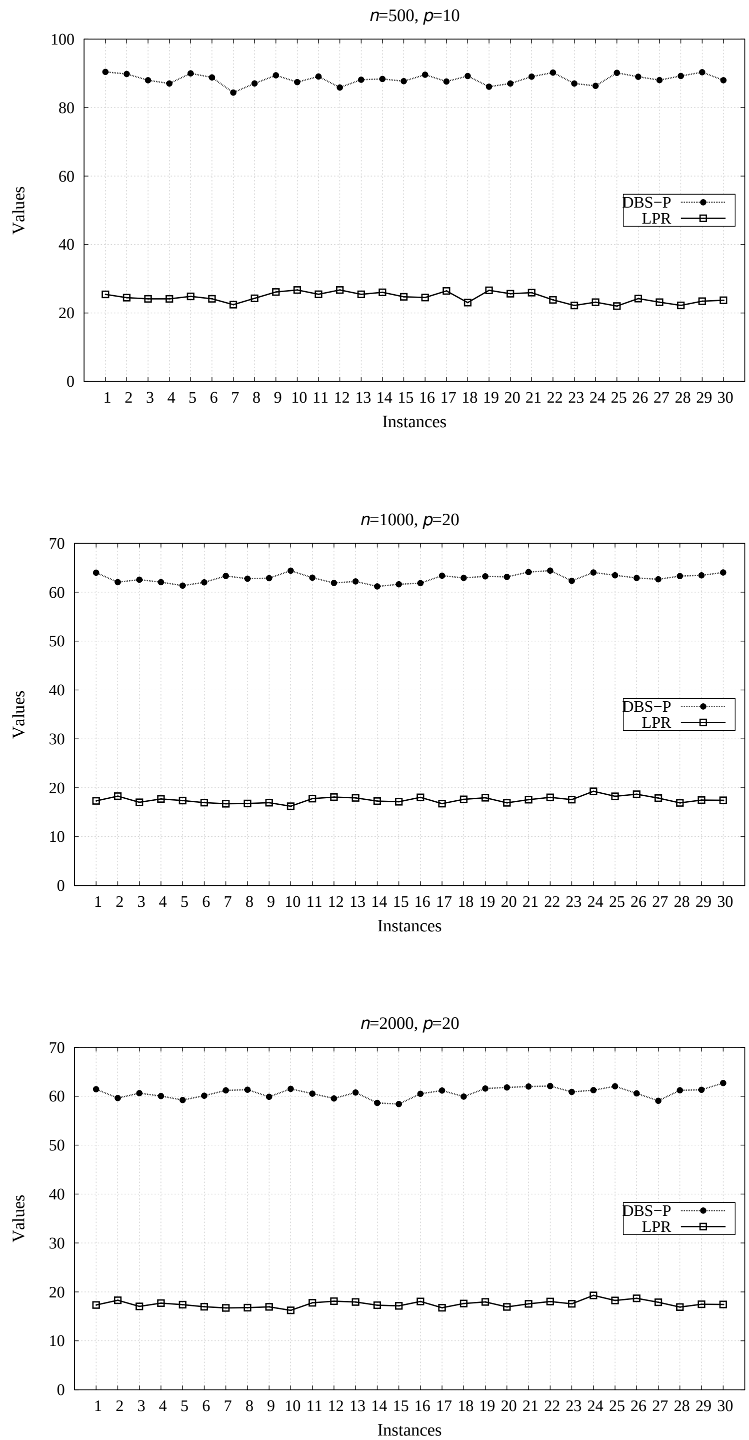

Figure 2: Comparison of LPR and DBS_P lower bounds.

which is remarkably high. This superiority in the quality of the bounds generated by LPR and DBS_P is better apreciated in Figure 2 where the values of both bounds per instance and size configuration are shown.

4.2.2 Comparison with an improved LP bound

From the previous experiment, it was clear that the better quality of the DBS_P bound came at a cost of a higher computational effort. Therefore, we investigate the improvement of the LPR bound when cast in a branch-and-bound (B&B) framework. As it is known, the B&B method iteratively improves its dual and primal bounds until optimality is reached. The main idea behind this experiment is to allow the B&B as much time as the computation of the DBS_P bound task, and make a comparison of the DBS_P and the improved LPR bound (ILPR) under the same computational effort.

Size (n, p)	RD (%)
$(500, 10)$	255.13
$(1000, 20)$	255.30
$(2000, 20)$	342.91

Table 3: Relative improvement of DBS_P with respect to ILPR.

This experiment was carried out on 15 instances of each size configuration $(n, p) \in \{(500, 10), (1000, 20), (2000, 20)\}$. Table 3 indicates the relative deviation (computed as in the previous test) between both ILPR and DBS_P bounds. The most important result in this experiment is that for all tested instances, the B&B method did not improve significantly the dual bound obtained at the root node, that is, the LPR bound. In other words, considering the same execution times for both strategies, the exact solution procedure failed to improve the linear relaxation while the proposed scheme is still better than the ILPR bound showing average relative deviations from 255.13% to 342.91%. Figure 3 shows the individual bounds values, per instance and size configuration, for each bounding scheme.

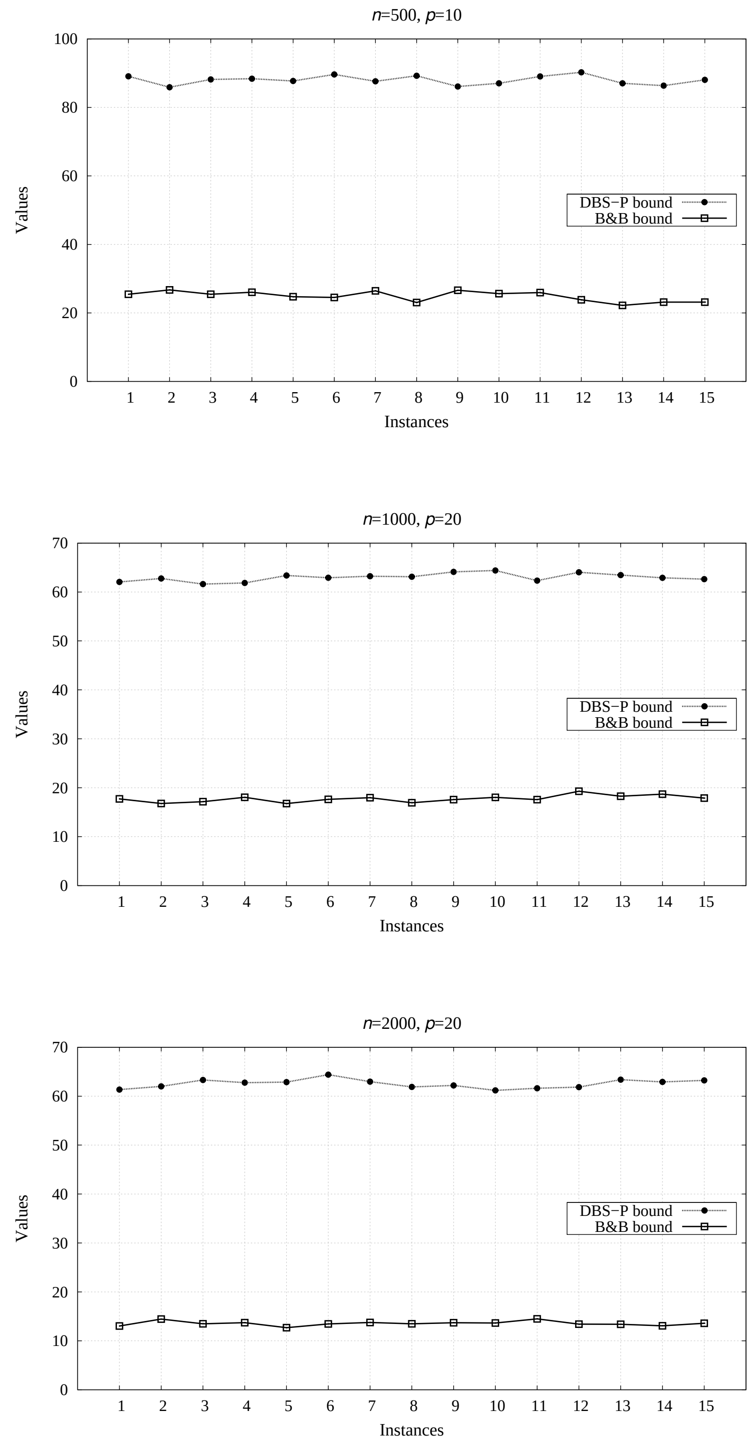

Figure 3: Comparison of B&B and DBS_P bounds.

4.2.3 Comparison with optimal solutions

Finally, the quality of the proposed DBS_P bound is assessed by comparing it with optimal solutions. To this end, we solved 60- and 100-node instances by B&B (20 instances on each set). This is the largest size that can be optimally solved in reasonable times.

Size (n, p)		ROG (%) DBS_P	LPR
$(60, 4)$	Best	0.10	59.94
	Average	5.66	66.59
	Worst	13.15	71.46
$(100, 6)$	Best	2.34	60.84
	Average	10.50	67.62
	Worst	16.58	72.83

Table 4: Comparison of DBS_P and LPR bounds vs. optimal solutions.

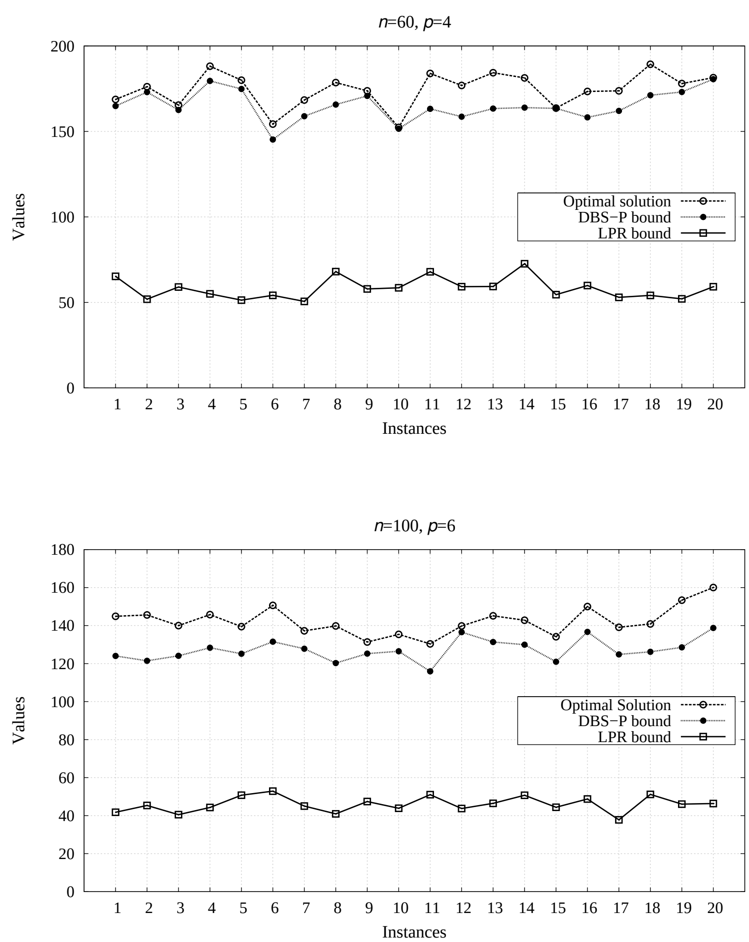

Figure 4: Comparison of DBS_P bounds and optimal solutions.

Results are summarized in Table 4. For each bounding procedure a relative optimality gap is computed. This gap gives the relative deviation on how far is the lower bound (lb) from the optimal solution (opt) and is defined as ROG $= 100\left(\frac{opt-lb}{opt}\right)$. As it can be seen from the table, the DBS_P scheme provides a more attractive choice that its LPR counterpart, confirming the results from previous experiments. In particular, it was observed that 90% of the 60-node instances had optimality gaps of less than 10% under the DBS_P scheme Figure 4 displays the LPR and DBS_P lower bound values as well as the optimal solution values of the different instances in each set (n, p).

5 Conclusions

In this paper we have presented a dual bounding scheme for a territory design problem. This problem includes compactness and balancing among territories as planning criteria. In particular, the problem addressed has been intractable through exact solution methods for real-world intance sizes, therefore different heuristic approaches have been proposed for this problem. However, to the best of our knowledge, there are no previous work on generating dual bounds for the territory design problem. As it is well known, the computation of dual bounds is important for assessing the quality of primal solutions, and moreover, dual bounds can be useful in the design of exact solution methods.

The proposed bounding procedure exploits the similarities of methodologies for solving the well known capacitated p-center problem. In this paper we extended the ideas underlying such methodologies and proposed an adaptation to handle multiple balancing constraints. Lower bounds for TDP are obtained by performing a binary search on the elements on the matrix of distances between basic units. In each iteration of the procedure, the resolution of a Lagrangian dual from a coverage location problem is considered. This allows to evaluate, for a given coverage radius, if it is possible to assign all the basic units in a feasible way into p territories. When this is not met, the explored radius becomes a lower bound for the territory design problem. In addition, a pre-processing technique to speed up the convergence of the procedure was developed by computing initial upper and lower bounds for TDP.

In the computational work, it was observed the positive impact of this simplification reducing up to 64.7% the number of explored radii during the binary search procedure which yields a significantly decrease in computation times. Furthermore, empirical evaluation showed that the proposed dual bound for TDP was of considerably higer quality than those provided by the linear programming relaxation of the model.

There are several extensions to this work that deserve attention. For instance, it was observed that the bottleneck in the overal execution time of the procedure is found at solving the TSKP subproblems derived from de Lagrangian relaxation of the maximum demand covering problem. Therefore, the derivation of efficient solution techniques for TSKP could greatly improve the efficiency of the proposed dual bounding scheme. To the best of our knowledge, this is a variation of the Knapsack Problem that has not been addressed before.

The study of other related location problems that can be used as auxiliary problems in the bounding scheme may also be worthwhile as they could provide different dual bounds for TDP. For instance, the Minimum Set Covering Problem (MSCP$_\delta$), which seeks to minimize the number of territories to create in order to assign all the basic units within a maximal assignation distance δ. Hence, similary to MDCP$_\delta$, it can be determined if a covering radius δ is a valid dual (primal) bound for TDP depending on the optimal value of MSCP$_\delta$, that is, the number of territories created in the optimal solution. In this way, a radius $\delta \in \bar{D}$ is a valid lower (upper) bound for TDP if the optimal value of its corresponding MSCP$_\delta$ is greater (smaller) than p.

A natural extension is to exploit the proposed bounding scheme for developing exact solution methods for TDP. Lagrangian heuristics form a wide family of methods that work well in finding effcient solutions for many integer programming problems. As the DBS_P procedure, these methods uses a Lagrangian relaxation of the problem at hand to obtain easily solved subproblems and approximately solves the Lagrangian dual through an iterative optimization scheme. In this process, some Lagrangian (dual) information is used as an input to guide the construction of feasible solutions. Solutions thus obtained are then submitted to local improvement in an overall procedure that is repeated for every algorithm iteration. The Lagrangian heuristic is then embedded into a branch-and-bound scheme that yields further primal improvements. This B&B scheme can either be an exact method that guarantees the optimal solution of the problem or be a fast heuristic. Although our bounding scheme relaxes an auxiliary problem instead the TDP, the DBS procedure can be extended to a Lagrangian heuristic framework to improve the primal solutions obtained during the subgradient optimization.

Acknowledgements: This work was supported by the Mexican National Council for Science and Technology, grants CONACYT CB-2005-01-48499Y and CB-2011-01-166397, and Universidad Aúnoma de Nuevo León under its Scientific and Technological Research Support Program, grants UANL-PAICYT CE012-09 and IT511-10.

References

[1] M. Albareda-Sambola, J. A. Díaz, and E. Fernández. Lagrangean duals and exact solution to the capacitated p-center problem. *European Journal of Operational Research*, 201(1):71–81, 2010.

[2] 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.

[3] 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.

[4] R. Church and C. ReVelle. The maximal covering location problem. *Papers in Regional Science*, 32(1):101–118, 1974.

[5] M. S. Daskin. *Network and Discrete Location: Models, Algorithms and Applications*. Wiley, New York, 1995.

[6] M. S. Daskin. A new approach to solve the vertex p-center problem to optimality: Algorithm an computational results. *Communications of the Operations Research Society of Japan*, 45(9):428–436, 2000.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H. P. Schwefel, editors, *Parallel Problem Solving from Nature–PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 849–858, Berlin, 2000. Springer.

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

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

[10] S. Elloumi, M. Labbé, and Y. Pochet. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.

[11] M. L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[12] M. L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15(2):2–21, 1985.

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

[14] R. D. Galvão and C. ReVelle. A Lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88(1):114–123, 1996.

[15] R. D. Galvão and C. ReVelle. A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. *European Journal of Operational Research*, 124(2):377–389, 2000.

[16] R. S. Garfinkel and G. L. Nemhauser. Solving optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B485–B508, 1970.

[17] A. M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Studies*, 2:82–114, 1974.

[18] M. Guignard. Lagrangean relaxation. *TOP*, 11(2):151–228, 2003.

[19] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162, 1970.

[20] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25, 1971.

[21] S. W. Hess and S. A. Samuels. Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4):998–1006, 1971.

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

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

[24] ILOG, Inc. ILOG CPLEX 11.0 User's Manual. Mountain View, 2007.

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

[26] 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.

[27] G. R. Kocis and I. E. Grossmann. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13(3):307–315, 1989.

[28] J. F. López-Pérez and R. Z. Ríos-Mercado. Embotelladoras ARCA uses operations research to improve territory design plans. Technical Report PISIS-2011-08, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, Mexico, June 2011.

[29] N. Megiddo, E. Zemel, and S. L. Hakimi. The maximum coverage location problem. *SIAM Journal of Algebraic and Discrete Methods*, 4(2):253–261, 1983.

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

[31] E. Minieka. The m-center problem. *SIAM Review*, 12(1):138–139, 1970.

[32] J. Molina, M. Laguna, R. Martí, and R. Caballero. SSPMO: a scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, 19(1):91–100, 2007.

[33] M. G. Narciso and L. A. N. Lorena. Relaxation heuristics for a generalized assignment problem. *European Journal of Operational Research*, 91(3):600–610, 1996.

[34] F. A. Ozsoy and M. Ç. Pinar. An exact algorithm for the capacitated vertex p-center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.

[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. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.

[37] R. Z. Ríos-Mercado and J. F. López-Pérez. Commercial territory design planning with realignment and disjoint assignment requirements. *Omega*, 2012. Forthcoming, doi: 10.1016/j.omega.2012.08.002.

[38] R. Z. Ríos-Mercado and J. C. Salazar-Acosta. A GRASP with strategic oscillation for a commercial territory design problem with a routing budget constraint. In I. Batyrshin and G. Sidorov, editors, *Advances in Soft Computing: Proceedings of the 10th Mexican International Conference on Artificial Intelligence (MICAI 2011), Part II*, volume 7095 of *Lecture Notes in Artificial Intelligence*, pages 307–318, Heidelberg, Germany, 2011. Springer.

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

[40] 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.

[41] 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*, 2012. Forthcoming, doi: 10.1007/s10732-011-9160-8.

[42] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, J. L. González-Velarde, and J. Molina. Multiobjective scatter search for a commercial territory design problem. *Annals of Operations Research*, 2012. Forthcoming, doi: 10.1007/s10479-011-1045-6.

[43] R. J. Shanker, R. E. Turner, and A. A. Zoltners. Sales territory design: An integrated approach. *Management Science*, 22(3):309–320, 1975.

[44] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

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

[46] A. A. Zoltners and P. Sinha. Sales territory design: Thirty years of modeling and implementation. *Marketing Science*, 24(3):313–332, 2005.