Omega 41 (2013) 525–535

Contents lists available at SciVerse ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Commercial territory design planning with realignment and disjoint assignment requirements

Roger Z. Ríos-Mercado [a,*], J. Fabián López-Pérez [b]

[a] Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, AP 111–F, Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico
[b] Graduate Program in Management Science, CEDEEM, Universidad Autónoma de Nuevo León, Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico

ARTICLE INFO

Article history:
Received 23 May 2011
Accepted 16 August 2012
Processed by B. Lev
Available online 23 August 2012

Keywords:
Bottled beverage distribution
Commercial districting
Mixed-integer linear programming
Branch-and-bound method
Heuristics

ABSTRACT

A territory design problem motivated by a bottled beverage distribution company is addressed. The problem consists of finding a partition of the entire set of city blocks into a given number of territories subject to several planning criteria. Each unit has three measurable activities associated to it, namely, number of customers, product demand, and workload. The plan must satisfy planning criteria such as territory compactness, territory balancing with respect to each of the block activity measures, and territory connectivity, meaning that there must exist a path between any pair of units in a territory totally contained in it. In addition, there are some disjoint assignment requirements establishing that some specified units must be assigned to different territories, and a similarity with existing plan requirement. An optimal design is one that minimizes a measure of territory dispersion and similarity with existing design. A mixed-integer linear programming model is presented. This model is unique in the commercial territory design literature as it incorporates the disjoint assignment requirements and similarity with existing plan. Previous methods developed for related commercial districting problems are not applicable. A solution procedure based on an iterative cut generation strategy within a branch-and-bound framework is proposed. The procedure aims at solving large-scale instances by incorporating several algorithmic strategies that helped reduce the problem size. These strategies are evaluated and tested on some real-world instances of 5000 and 10,000 basic units. The empirical results show the effectiveness of the proposed method and strategies in finding near optimal solutions to these very large instances at a reasonably small computational effort.

1. Introduction

The Territory Design Problem (TDP) may be viewed as the problem of grouping basic units into subsets according to specific planning criteria. These subsets are known as territories. There are some other spatial constraints as part of the geographic definition of the problem. Depending on the context of the problem, the concept "territory design" may be used as equivalence to "districting". Districting is a truly multidisciplinary research area which includes several fields such as geography, political science, public administration, and operations research. However, all these problems have in common the task of subdividing the region under planning into a number of territories, subject to some capacity constraints. Indeed, territory design problems emerge from different types of real-world applications. We can mention pick up and delivery applications, waste collection, school districting, sales workforce territory design and even some others related to geo-political concerns. Most public services including hospitals, schools, and postal delivery, are managed along territorial boundaries. We can mention either economic or demographic issues that may be taken in consideration for setup a balanced territory.

The problem addressed in this work is motivated by a real-world application in the bottled beverage distribution industry. As each territory is to be served by a single resource, it makes sense to use some planning criteria to balance the quantity of customers, product demand, and workload required by the dispatchers or truck drivers to cover each territory. Moreover, it is often required to balance the demand among the territories in order to delegate responsibility fairly. To this end, the firm wishes to partition the set of city blocks or basic units (BUs) into disjoint territories that are suitable for their commercial purposes.

This combinatorial optimization problem is NP-hard [1]. To the best of our knowledge, the TDP version studied in this problem has not been tackled before. Related versions have been studied, though. State-of-the-art exact methods can solve instances of some simplified models of around 100–150 BUs.

* Corresponding author. Tel.: +52 81 1492 0383.
E-mail addresses: roger.rios@uanl.edu.mx, roger@yalma.fime.uanl.mx (R.Z. Ríos-Mercado), jesus.lopezpz@uanl.edu.mx (J.F. López-Pérez).

0305-0483/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.omega.2012.08.002

Typical real-world instances are very large and intractable by exact methods. There has been some heuristic approaches for commercial TDPs. For instance, Ríos-Mercado and Fernández [2] developed a Reactive Greedy Randomized Adaptive Search Procedure (GRASP) for a problem similar to ours; however, they measure territory dispersion based on the objective function of a p-Center Problem, and they do not consider the disjoint assignment constraints nor similarity with exiting plan. In our case, we are measuring dispersion by means of a function from a p-Median Problem. This of course leads to a different structure and makes previous approaches inapplicable. In addition, one of the main goals of our work is to develop a tool that can be relatively easy to implement in commercial off-the-shelve modeling languages and optimizers. This is of great value to the company.

Now, when this TDP is modeled as a mixed-integer programming problem, one of the main difficulties is that of the exponential number of connectivity constraints. These simply cannot be written out explicitly. On the other hand, this decision problem can be viewed as a two-level decision problem where at the top level one has to decide where to place territory centers (called location level) and at a second level one has to assign BUs to centers (called allocation level). Location–allocation approaches to TDP have been applied before. In our case, from a practical perspective there is a relatively fair knowledge of reasonable sites to act as territory centers. Therefore, by assuming we have a good representation of these centers and fix them in advance, we focus on the allocation problem.

In this paper, we present a heuristic solution approach based on the iterative resolution of an associated mixed-integer programming model for the TDP aimed at obtaining high quality solutions to large-scale instances. The algorithm consists of iteratively solving a relaxed MILP model (relaxing the connectivity constraints), identifying violated constraints by solving an easy separation problem, and adding these violated cuts to the model. The procedure continues until no more connectivity constraints are needed. This is similar to the exact approach developed by Salazar-Aguilar et al. [1], except that they apply it to the complete model solving instances of up to 100–150 BUs. In our case, we apply this technique to the relaxed model which is solved considerably faster allowing the solution of larger instances. In addition, we have implemented some strategies that allow to fix some binary variables in advance. The solution method and algorithmic strategies were evaluated on a case study from industry. We found that this procedure is successful in finding good quality solutions for large-scale instances (e.g., 5000 and 10,000 BUs) in reasonable times.

The paper is structured as follows. In Section 2 we describe the problem. In Section 3 we present an overview of the most relevant work on models and algorithms for territory design. This is followed by Section 4, where the mathematical framework is presented in detail. The proposed solution approach is fully described in Section 5. In Section 6 we present the empirical evaluation of the proposed approach. We wrap up the paper in Section 7, with some conclusions and final remarks.

2. Problem description

Given a set of city blocks or basic units (BUs) for delivering bottled beverages, we need to partition this set into a given number p of disjoint territories, where the value of p is known. Every BU should be contained in exactly one territory. The goal of the design is to obtain territories that are as compact as possible, that is, the BUs in a given territory must be as close to each other as possible. Territory connectivity is another important planning requirement. Connectivity means that the basic units that conform a territory have to be geographically connected. It is easy to understand that in order to obtain contiguous territories, explicit neighborhood information for the basic units is required. For each BU, the following information is known with certainty: location coordinates (from the firm GIS), and three measurable attributes: (i) number of customers, (ii) product demand measured by the number of 12-bottle boxes, and (iii) workload measured in time (min). The activity measure of a territory is the total sum of the activity measure of its individual basic units. The firm wants to design territories that are balanced (similar in size) with respect to each of the three different activity measures in every BU. That is, the total number of customers, product demand, and workload assigned to each territory should be fairly distributed among the territories. It is interesting to point out that only a few authors consider more than one criterion simultaneously for designing balanced territories (e.g, Deckro [3], Zoltners [4], Zoltners and Sinha [5]).

The firm also seeks some degree of similarity with the existing design. This is achieved by requiring that at least certain given proportion of the current BUs assigned to each territory remain assigned to the same territory. In a similar fashion, there are some predefined pairs of BUs that must be assigned to different territories. We called these *disjoint assignment constraints*. As can be verified, all these features could be easily extended to consider some territories that may already exist at the beginning of the planning process. That means that our method should be prepared to take the existing territories into account and then add additional basic units to them. This modeling feature could be applied to take into account geographical obstacles, e.g., rivers and mountains. The problem can be summarized as follows: partition the set V of basic units into p territories which satisfy the specified planning criteria such as balance, compactness, connectivity, disjoint assignment, and similarity with existing design.

3. Overview of models and solution approaches

The recent paper by Kalcsics et al. [6] is an extensive survey on approaches to TDP that gives an up to date state-of-the-art and unifying approach to the topic. For a more extensive review related to sales districting see Zoltners and Sinha [7]. Another recent survey on districting models is the one by Duque et al. [8]. In a related issue, Mansfield et al. [9] report a preliminary study of several companies investigating the relationships between strategies of diversification and geographic dispersion of territories covered and structural variables relating to the number of structural differentiations in a company, the decentalization of decision-making and functional specialization. In this section, we focus on reviewing the work most related to our research.

As far as commercial territory design is concerned, Vargas-Suarez et al. [10] address a related commercial TDP with a variable number of territories, using as an objective a weighted function of the activity deviations from a given goal. No compactness criterion was considered. A basic GRASP was developed and tested in a few instances obtaining relatively good results. Ríos-Mercado and Fernández [2] studied the problem by considering compactness and contiguity but without joint assignment constrains. They used the objective function of the p-Center Problem for modeling territory dispersion. In that work, the authors proposed and developed a reactive GRASP algorithm for handling large instances. They evaluated their algorithm on 500- and 1000-node instances with very good results. More recently, Salazar-Aguilar et al. [1] develop an exact optimization scheme for solving the TDP with double balancing and connectivity constraints. They used their framework for solving models with both types of dispersion functions: the one based on the p-Center Problem and the one based on the p-Median Problem. They observed that models with an objective function from the p-Median Problem were solved faster than the ones using the objective from the p-Center Problem. Furthermore, they also observed that solutions obtained from the relaxation of the median-based models had a very high degree of connectivity. Still, the largest instance they could solve for the median-based models was about 150 BUs. Our idea is to use a similar framework than the one they used in their work, except that we will be focusing in the allocation level aiming at large instances. More recently, several approaches have been developed for multiobjective versions of the commercial TDP, including both exact optimization approaches [11] and metaheuristic methods [12,13].

Table 1 summarizes this overview. In the first column, the citation is displayed. Column 2 shows the objective to be optimized (D for dispersion, B for balancing). If more than one criterion is shown it represents a multi-objective problem. Column 3 shows the requirement on p (F for fixed, V for variable). Column 4 shows the type of function used for modeling dispersion (p M for median-based function, p C for center-based function, d for diameter function), where "–" means no dispersion was considered. Columns 5–9 show what type of constraints or requirements are present (B: balancing, C: connectivity, JA: joint assignment, S: similarity with existing plan), where Y or N means yes or no, respectively. In addition, the "B" column indicates whether single (S) or multiple (M) balancing constraints are present. The last column indicates the type of method used for solving the problem showing (e) or (h) for exact or heuristic method, respectively.

All of these works address both the location and allocation decisions. Our work focus on the allocation level only. As can be seen from the table, even though our work focus on solving the allocation level only, it is the only work that handles both joint assignment and similarity with existing plan requirements, in addition to the other common constraints. As it will be seen later, our solution methodology is heuristic in nature; however, we have empirically shown the solution is relatively close to the optimal solution, and the model is solved significantly faster which allows to handle very large-scale instances. There are certainly other works in different applications of territory design that have similar structure as that of our problem; however, to the best of our knowledge, none of these address the joint assignment and similarity with existing plan.

There have been some studies on territory realignment that consists of developing territory designs subject to some constraints that attempt to keep an existing plan to the best possible extent. This issue have been studied in the context of political districting [14], school districting [15], and sales territory design [16]. In our problem, there is an interest on having a similarity, at least partially, not with an entire existing design, but with a given set of BUs. To the best of our knowledge, our model is the first to consider this issue within commercial districting. There are other territory alignments problems (e.g., Ronen [17]) that incorporate travel time in the decision process.

4. Modeling framework

The problem is modeled by a graph $G=(V,E)$, where a city block or basic unit (BU) i is associated with a node, and an edge connecting nodes i and j exists in E if blocks i and j are adjacent to each other. Now each node $i \in V$ has several associated parameters such as geographical coordinates (c_i^x, c_i^y), and three measurable activities. Let w_i^a be the value of activity $a \in A = \{1,2,3\}$ at node i, where $a=1$, 2, and 3, refers to the number of customers, product demand, and workload, respectively. A territory is a subset of nodes $V_k \subset V$. The number of territories is given by the parameter p. It is required that each node is assigned to only one territory. Thus, the territories define a partition of V. One of the properties sought in a solution is that the territories are balanced with respect to each of the activity measures. Thus, let us define the size of territory V_k with respect to activity a as: $w^a(V_k) = \sum_{i \in V_k} w_i^a$, $a \in A$. Due to the discrete structure of the problem and to the unique assignment constraint, it is practically impossible to have perfectly balanced territories with respect to each activity measure. To account for this, we measure the balance degree by computing the relative deviation of each territory from its average size μ^a, given by $\mu^a = w^a(V)/p$, $a \in A$. Another important feature 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 V_k must induce a connected subgraph of G. As mentioned before, due to strategic or political reasons, there are some BUs that are required to be assigned to different territories. Let H be set that contains all pairs of units that must be assigned to different territories, that is, $H = \{(j_1, j_2) \in V \times V | j_1$ and j_2 must be assigned to different territories $\}$. This set will be used to represent these disjoint assignment constraints.

The company is also interested in keeping certain similarity with a subset of BUs from an existing plan. The concept of territory realignment [14–16] considers somehow either as a constraint or a term in the objective function a measure of dissimilarity with respect to previous plan. In this particular case, the company wishes to keep a similarity not with an entire existing design but with a subset of BUs. Let F^i denote the prespecified subset of BUs associated to center i from an existing plan. Then the firm wishes that the new plan assigns to the new territory with center in i a significant proportion of the BUs from set F^i taking into account of course the corresponding distance measure. For instance, if two given units, say i and j belong to F^k, preference for assigning either of this to the new territory with

Table 1
Related work on commercial districting.

Reference (1st author)	obj	p	D	B	C	JA	S	Method
Vargas-Suárez [10]	B	V	–	N	Y	N	N	(h) GRASP
Ríos-Mercado [2]	D	F	p C	M	Y	N	N	(h) Reactive GRASP
Caballero-Hernández [20]	D	F	p C	Y	Y	Y	N	(h) GRASP
Ríos-Mercado [21]	D	F	d	Y	Y	N	N	(h) GRASP
Salazar-Aguilar [1]	D	F	pM	M	Y	N	N	(e) Branch-and-cut (h) IQP reformulation
	D	F	pC	M	Y	N	N	(e) Branch-and-cut (h) IQP reformulation
Salazar-Aguilar [11]	D,B	F	pM	S	Y	N	N	(e) ϵ-constraint & B&C
Salazar-Aguilar [12]	D,B	F	pM	S	Y	N	N	(h) GRASP
Salazar-Aguilar [13]	D,B	F	pM	S	Y	N	N	(h) Scatter search

center in k should be given to the unit nearest to k. This may be achieved by introducing a penalty term in the objective function q_{ij}. In addition, it is required that at least certain number of these BUs meet this assignment. This can be achieved by introducing a corresponding constraint. These can be seen in the model below.

Finally, industry demands that in each of the territories, blocks must be relatively close to each other. One way to achieve this is for each territory to select an appropriate node to be its center, and then to define a distance measure such as $D=\sum_{k=1}^{p}\sum_{j\in V_k} d_{c(k),j}$ where $c(k)$ denotes the index of the center of territory k so $d_{c(k),j}$ represents the Euclidian distance from node j to center of territory k. So maximizing compactness is equivalent to minimizing this dispersion function D. All parameters are assumed to be known with certainty. The problem can be thus described as finding a p-partition of V satisfying the specified planning criteria of balancing, connectivity, and disjoint assignment, that minimizes the above distance-based dispersion measure and partial similarity with existing set of BUs.

4.1. MILP formulation

Indices and sets

n	number of blocks (BUs)
p	number of territories
i,j	block indices; $i,j \in V=\{1,2,\dots,n\}$
a	activity index; $a\in A=\{1,2,3\}$
k	territory index; $k\in K=\{1,2,\dots,p\}$
E	edge set of adjacent blocks
H	set of pairs of BUs that must be assigned to different territories
F^i	set of BUs that are assigned to territory with center in i under a current design
N^i	$(=\{j\in V:(i,j)\in E \vee (j,i)\in E\})$ set of nodes which are adjacent to node i; $i\in V$

Parameters

w_i^a	value of activity a in node i; $i\in V$, $a\in A$
d_{ij}	Euclidian distance between i and j; $i,j\in V$
q_{ij}	weight of assigning unit j to center i equal to $0.5d_{ij}$ if $j\in F^i$; 0, otherwise; $i,j\in V$
τ^a	relative tolerance with respect to activity a; $a\in A$, $\tau^a\in[0,1]$

Computed parameters

$w^a(X_k)$	$(=\sum_{j\in X_k} w_j^a)$ size of set X_k with respect to a; $a\in A, X_k\subset V$
μ^a	$(=w^a(V)/p)$ average (target) value of activity a; $a\in A$

Decision variables: In the original problem we are not concerned with territory centers; however, we introduce binary variables based on centers for modeling the dispersion measure

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

Note that $x_{ii}=1$ implies that unit i is a territory center.

Model (*TDP*)

$$\min \quad f(x)=\sum_{i,j\in V} d_{ij}x_{ij}+\sum_{\substack{i\in V\\ j\in F^i}} q_{ij}(1-x_{ij}) \tag{1}$$

$$\text{s. t.} \quad \sum_{i\in V} x_{ij}=1, \quad j\in V \tag{2}$$

$$\sum_{i\in V} x_{ii}=p \tag{3}$$

$$\sum_{j\in V} w_j^a x_{ij} \le (1+\tau^a)\mu^a x_{ii}, \quad i\in V,\ a\in A \tag{4}$$

$$\sum_{j\in V} w_j^a x_{ij} \ge (1-\tau^a)\mu^a x_{ii}, \quad i\in V,\ a\in A \tag{5}$$

$$\sum_{j\in \cup_{v\in S}N^v\setminus S} x_{ij}-\sum_{j\in S} x_{ij} \ge 1-|S|, \quad i\in V,\ S\subset V\setminus(N^i\cup\{i\}) \tag{6}$$

$$x_{ij}+x_{ih}\le 1, \quad i\in V \quad (j,h)\in H \tag{7}$$

$$\sum_{i\in V}\sum_{j\in F^i} x_{ij} \ge \alpha|\cup_i F^i| \tag{8}$$

$$x_{ij}\in\{0,1\}, \quad i,j\in V \tag{9}$$

Objective (1) incorporates a term that measures territory dispersion and a term that favors the assignment of a subset of units from existing plan. Constraints (2) guarantee that each node j is assigned to a territory. Constraint (3) sets the number of territories. Constraints (4) and (5) 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. In particular, the upper bound balancing constraints (4) also assure that if no center is placed at i, no customer can be assigned to it. Constraints (6) guarantee the connectivity of the territories. These constraints, proposed by Drexl and Haase [18], are similar to the constraints used in routing problems to guarantee the connectivity of the routes. These constraints can be explained as follows Let $S\subset V\setminus(N^i\cup\{i\})$, that is a subset whose BUs are not adjacent to center i and its adjacent neighbors N^i. Note that if there is at least one BU j in S that is not assigned to territory center i, then the second term of the left-hand side becomes strictly less than $|S|$, making the constraint redundant. Now, when all BUs in S are assigned to center i, then the first term on the left-hand side must be greater than 1, that is, there must be at least one BU k surrounding S that must be assigned to center i as well. Recursively applying the same rationale to the set $S\cup\{k\}$ results in a territory connected to node i. Note that, as usual, there is an exponential number of such constraints. The disjoint assignment is represented by constraints (7). Constraints (8) assure that at least a minimum number of BUs from existing plan is assigned, where α is a user-specified parameter usually set to 0.10–0.20 in practice.

Computational complexity: This TDP is $\mathcal{NP}$-hard. It is clear that a given solution to our TDP can be checked for feasibility in polynomial time. Now, if we consider a special case of TDP where $F=\emptyset$, for all $i\in V$, and $H=\emptyset$, we are left with the *regular* commercial districting problem, that is, a dictricting problem seeking to minimize a median-based dispersion function subject to balancing and connectivity constraints, which is known to be $\mathcal{NP}$-hard [1]. That is, the regular commercial districting problem is polynomially reducible to our specific TDP. It follows our TDP is $\mathcal{NP}$-hard too.

Allocation model: We have attempted to solve Model TDP with a branch-and-bound method with very limited success. While instances of up to 150-nodes are somewhat tractable, it is no longer possible to solve larger instances within a few hours of CPU. The model has n^2 binary variables and a very weak LP relaxation.

The problem can be decomposed into a two-level hierarchy problem. One can see a location level, which has to do with placing the territory centers, and then an allocation level, which has to do with assigning nodes to centers. Since our aim is to provide solutions to very large instances (in the order of 5000–10,000 nodes), we make the assumption that the set of centers is given and focus our effort in the allocation level. Let V_c be the set of centers. This set can be approximately obtained by means of previous knowledge, a heuristic, or a truncated branch and bound method. The allocation level model becomes

$$\text{(AM) min} \quad f(x) = \sum_{\substack{i \in V_c \\ j \in V}} d_{ij} x_{ij} + \sum_{\substack{i \in V_c \\ j \in F^i}} q_{ij}(1 - x_{ij}) \tag{10}$$

$$\text{s. t.} \quad \sum_{i \in V_c} x_{ij} = 1, \quad j \in V \tag{11}$$

$$\sum_{j \in V} w_j^a x_{ij} \le (1+\tau^a)\mu^a, \quad i \in V_c, \quad a \in A \tag{12}$$

$$\sum_{j \in V} w_j^a x_{ij} \ge (1-\tau^a)\mu^a, \quad i \in V_c, \quad a \in A \tag{13}$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \ge 1 - |S|, \quad i \in V_c, \ S \subset V \setminus (N^i \cup \{i\}) \tag{14}$$

$$x_{ij} + x_{ih} \le 1, \quad i \in V_c \quad (j,h) \in H \tag{15}$$

$$\sum_{i \in V} \sum_{j \in F^i} x_{ij} \ge \alpha |\cup_i F^i| \tag{16}$$

$$x_{ij} \in \{0,1\}, \quad i \in V_c, \ j \in V \tag{17}$$

Model AM has *pn* binary variables. In typical location–allocation methods (Hess et al. [19], Kalcsics et al. [6]), the allocation model to be addressed has single balancing constraints, no contiguity constraints and no disjoint assignment constraints. The way this allocation problem is solved is by replacing the single balancing constraints by a single equation (i.e., making the tolerance parameter equals to zero) and relaxing the integrality restriction of the binary variables. The result is a transportation problem that is solved relatively efficiently. In this solution, which of course has perfect balance, there might fractional variables, i.e, a variable may be partially assigned to two or more centers. This new problem is named the split resolution problem and need to be solved according to certain criteria depending on the specific context. After the split resolution has been solved, the solution obtained may no longer necessarily satisfy the balancing constraints.

In our allocation model, we must deal simultaneously with multiple balancing constraints, connectivity constraints, and disjoint assignment constraints, which makes typical location–allocation procedures not applicable. Thus instead, our goal is to deal directly with the allocation model by developing several strategies within a branch-and-bound framework that would allow us to solve relatively large instances. By relaxing the connectivity constraints (14) from Model AM, we are left with the following relaxed model:

$$\text{(AMR) min} \quad f(x) = \sum_{\substack{i \in V_c \\ j \in V}} d_{ij} x_{ij} + \sum_{\substack{i \in V_c \\ j \in F^i}} q_{ij}(1 - x_{ij}) \tag{18}$$

$$\text{s. t.} \quad \sum_{i \in V_c} x_{ij} = 1, \quad j \in V \tag{19}$$

$$\sum_{j \in V} w_j^a x_{ij} \le (1+\tau^a)\mu^a, \quad i \in V_c, \quad a \in A \tag{20}$$

$$\sum_{j \in V} w_j^a x_{ij} \ge (1-\tau^a)\mu^a, \quad i \in V_c, \quad a \in A \tag{21}$$

$$x_{ij} + x_{ih} \le 1 \quad i \in V_c \quad (j,h) \in H \tag{22}$$

$$\sum_{i \in V} \sum_{j \in F^i} x_{ij} \ge \alpha |\cup_i F^i| \tag{23}$$

$$x_{ij} \in \{0,1\}, \quad i \in V_c, \ j \in V \tag{24}$$

In the following section we describe in detail the solution procedure.

5. Solution approach

In this section we present a solution strategy for solving the Allocation Model (AM) given by (10)–(17), one main difficulty in the exponential number of connectivity constraints (14), which implies it is practically impossible to write them out explicitly. Therefore, we consider instead the relaxation AMR of AM that consists of relaxing these connectivity constraints. The basic idea of our method is to solve model AMR and then check if the solutions obtained satisfy the connectivity constraints. To determine the violated connectivity constraints, a relatively easy separation problem is solved, and these cuts are added to model AMR. This procedure iterates until no additional connectivity constraints are found and therefore an optimal solution to model AM is obtained. This is guaranteed because the separation problem for identifying violated cuts is solved exactly. A general overview of the method is depicted in Fig. 1.

In Step 1, a branch-and-bound method is used (since we are not relaxing the integrality requirements of the binary variables). This approach is motivated by the fact that model AMR can be solved optimally by current branch-and-bound methods relatively fast for quite large instances. For instance, model AMR can be solved in around 30 s of CPU time in a PC for 5000-node instances. In addition, identifying and generating the violated cuts in Step 2 can also be done in polynomial time, such that the overall procedure may be suitable as long as the number of iterations needed to reach optimality is not too large. The algorithm delivers an optimal solution to model AM. Several issues are of particular interest. We would like to investigate the empirical behavior of the method in terms of the number of iterations/cuts required to reach optimality. In addition, the fact we are assuming a fixed set of centers can be further exploited to develop several algorithmic strategies like variable fixing in Step 1. These are further elaborated below.

5.1. Algorithmic strategies for speeding up convergence

- *Variable fixing*: *Eliminating assignments of relatively far units.* We proceed now to reduce the complexity of our problem by eliminating some unnecessary binary variables x_{ij}. This idea is based upon the fact that in an optimal solution, from a practical standpoint it makes no sense to assign a BU that is very far away from a given territory center. Making this assignment will have a very negative impact on the objective function. It is clear that theoretically one can build a pathological instance where this might be the case; however, given the particular data distribution for this problem this never

function method( )

Input: An instance of the TDP problem.

Output: A feasible solution X.

1 Solve model AMR and obtain solution X;
2 Identify a set C of violated constraints of model AM for solution X;
3 If $|C| > 0$, add these constraints to model AMR and go to Step 1;
4 Return X;

end method

Fig. 1. A pseudocode of solution procedure.

happens in practice. Thus, for each BU i we determine a reduced feasible subset R_i, such that we fix $x_{ij}=0$ for all $j \in \overline{R}_i$. For each i we have reduced the number of binary variables from n to $|R_i|$. This is done as follows. First, for each center $i \in V_c$ we sort all the remaining nodes by nondecreasing order of d_{ij}. Let (j) denote the j-th nearest BU to i breaking ties arbitrarily, that is $d_{i(j)}$ denote the distance from BU i to the j-th nearest BU. Then, given a user specified parameter $\beta \in (0,\infty)$ the set R_i is given by $R_i = \{(j) \in V : \sum_{k=1}^{j} w^a_{i(j)} \le \beta\mu^a$ for at least one $a \in A\}$. That is R_i is formed by the nearest BUs to i such that their accumulated sum of weights with respect to all activities do not exceed this threshold for at least one activity. A very large value of β implies $R_i=V$, so no reduction takes place. As β gets smaller, the number of variables fixed at 0 grows. A relatively small value of β means only a few binary variables will be considered (as many will be fixed at 0) over-compromising the optimality of the solution. An issue to investigate is precisely the sensibility and trade-off between solution quality and computing time as a function of β.

- *Variable fixing*: *Preassigning relatively close units*. Applying a similar rationale as in the previous point, it is possible to find a set of relative close units to a given center i such that in any optimal solution, all the units belonging to this set are always assigned to i. Again, while one can build an example where this might not happen, in practice we always see a considerable portion of relative close BUs being assigned to a center i. To this end, we determine a set K_i such that $x_{ij}=1$ for all $j \in K_i$. Given a user-specified parameter γ the set K_i is given by $K_i = \{(j) \in V : \sum_{k=1}^{j} w^a_{i(j)} \le \gamma\mu^a$ for all $a \in A\}$. That is K_i is formed by the nearest BUs to i such that their accumulated sum of weights with respect to all activities do not exceed this threshold for all activities. Here a value of $\gamma=0$ implies $K_i=\emptyset$ and no reduction is applied. The larger γ the larger the number of binary variables will be fixed at 1. Again, it is important to investigate the trade-off between solution quality and time as a function of γ.
- *Strengthening of connectivity constraints*. One way to strength the formulation of the relaxed model is by introducing the following constraints:

$$x_{ij} \le \sum_{q \in N^j} x_{iq}, \quad i \in V_c,\ j \in V \tag{25}$$

These valid inequalities can be interpreted as follows. If BU j is assigned to territory with center in i at least one of its neighbors ($q \in N^j$) needs to be assigned to the same territory as that of BU j. These constraints avoid territories with just a single BU unconnected. The motivation for this stems from the fact that previous research [1] has shown that optimal solutions of the relaxed model contain most of the unconnected subsets S with cardinality equal to 1, that is $|S|=1$. As can be seen, there is a polynomial number of these new constraints (25), thus these can be easily added to the model without imposing a considerable computational burden. Of course, Step 2 of the algorithm still checks for violated constraints with subsets of cardinality $|S|>1$.
- *Finding violated inequalities*. Step 2 of the method can be efficiently done in polynomial time. Let $X=(X_1,\ldots,X_p)$ be a design found in Step 1. For each territory k, there is an associated subgraph induced by X_k given by $G_k=(X_k,E(X_k))$. It is well known that finding all connected components of a graph can be done by breadth first search (BFS) in $O(|E(X_k)|)$. So we apply BFS to graph G_k and find its r connected components $(G^1_k,\ldots,G^r_k)$, with corresponding node sets $(X^1_k,\ldots,X^r_k)$. It is clear that $r=1$ implies G_k is connected; otherwise let us assume without loss of generality that the center of X_k, named $c(k)$ belongs to G^1_k. Clearly, each of the remaining subsets $X^2_k,\ldots,X^r_k$ is disconnected from the center. Each of these corresponds to a violated constraint (14) where set X^q_k plays the role of set S in (14). By repeating this procedure for every set X_k one can efficiently solve this separation problem optimally and add all found violated cuts to set C in Step 3.
- *Forced connectivity strategy for faster convergence*. Step 2 is key for the efficiency of the proposed methods. It has been observed that the number of iterations needed to find a connected solution in instances of up to 5000 BUs is very reasonable. However, for larger instances up to 10,000 BUs the computational effort grows considerable. The main cause for this is that it takes a significant large amount of iterations to converge. This stems from the fact that the combinatorial nature of all possible unconnected subsets makes the algorithm find and add a large number of different cuts. Therefore, we have implemented a heuristic strategy that can be employed as part of the method when faced with large-scale instances.
- To motivate this strategy, it is important to note that if we keep track of a single BU throughout the execution of the algorithm, it can happen that this node may or may not belong to an unconnected subset in the following iteration. In many cases, an oscillating behavior between being part of an unconnected subset and being part of the connected territory is followed by many of these nodes. Therefore to avoid this nasty behavior, instead of solving the AMR model we add a penalty term to the objective function that would favor the assignment of BUs that are already found to belong to a connected territory. The basic idea of this strategy is to take advantage of the connectivity information from a given iteration to attempt to avoid the oscillating behavior expecting to reduce at every iteration the number of BUs that belong to unconnected subsets.
- Let us define Z_t as the number of BUs that are disconnected at iteration t. This parameter is dynamic because the number of disconnected BUs changes at each iteration. In fact, it is expected that this parameter Z_t will tend to zero as the number of iterations grows. Let U^t_i be the set of BUs that are connected to territory with center in i at iteration t. Then we add a term to the objective function (18) as follows:

$$\text{Minimize } g(x) = f(x) + \delta \sum_{i \in V_c} \sum_{j \in U^t_i} r_{ij}(1-x_{ij})/(Z_t+1) \tag{26}$$

- In this added term, a penalty $r_{ij}=d^{\max}-d_{ij}$, where $d^{\max}=\max_{ij}\{d_{ij}\}$ implies closer assignments are preferred, dividing by Z_t+1 avoids division by zero and makes the preference of the assignment of already connected units stronger as the number of iterations grows, and parameter δ allows the user to control the weight of this added term with respect to the original objective function. Naturally, setting $\delta=0.0$ implies deactivation of this strategy.

6. Empirical evaluation

We implement our model on X-PRESS MIP Solver from FICO™ (Fair Isaac, Dash Optimization before). The method was executed on a PC with 2 Intel Cores at 1.4 GHz and Win X64 operating system. For evaluation the proposed method, we use some real-world instances of 5000 and 1000 BUs and 50 territories. In all experiments we set $\tau^a=0.10$ for all $a \in A$ and a relative optimality gap of 0.01% as stopping criterion. These instances are available at: http://yalma.fime.uanl.mx/~roger/ftp/tdp/.

Table 2 shows the effect on problem reduction by using different values of parameters β and γ, discussed in Section 5.

Table 2
Problem reduction effect.

Size (*n*,*p*)	NBV (*np*)	β	γ	RNBV	Reduction (%)
(5000, 50)	250,000	3.0	0.50	7,191	97.1
		3.0	0.25	10,501	95.8
		3.0	0.10	12,428	95.0
		3.0	0.00	13,542	94.6
		4.0	0.50	9,702	96.1
		4.0	0.25	14,612	94.1
		4.0	0.10	17,545	93.0
		4.0	0.00	19,400	92.2
		8.0	0.50	20,484	91.8
		8.0	0.25	30,365	87.8
		8.0	0.10	36,253	85.5
		8.0	0.00	39,755	84.1
		50.0	0.00	250,000	0.0
(10,000, 50)	500,000	3.0	0.50	14,615	97.1
		3.0	0.25	21,227	95.8
		3.0	0.10	25,027	95.0
		3.0	0.00	30,202	94.0
		4.0	0.50	19,968	96.0
		4.0	0.25	29,609	94.1
		4.0	0.10	35,352	92.9
		4.0	0.00	39,244	92.1
		8.0	0.50	41,531	91.7
		8.0	0.25	60,810	87.8
		8.0	0.10	72,214	85.5
		8.0	0.00	79,693	84.1
		50.0	0.00	500,000	0.0

The first two columns reflect the size of the original instance in terms of its number of BUs, number of territories and number of binary variables. The third and fourth column display values of parameters β and γ, respectively. The fifth column (RNBV) displays the number of binary variables after the reduction, and the last column the relative reduction with respect to the original size given by $100\times(\text{NBV}-\text{RNBV})/\text{NBV}$. It can be seen how the number of binary variables in the reduced problem grows as β increases and γ decreases. Note that the case $\beta=50.0$ and $\gamma=0.0$ implies that no reduction is applied. In summary, the strategy we adopt is to decrease the feasible solution space to deal with a reduced problem that can be solved more efficiently without a significant loss on optimality. This trade-off on optimality is evaluated next.

We now apply the proposed method to instances of 5000 BUs with 50 territories. In this experiment we set $\delta=0.0$, that is no forced connectivity strategy applied. The goal of this experiment is to assess the trade-off between solution quality and execution time for different values of β and γ.

Table 3 displays the results for the 5000-BU instance. The first two columns display the values of β and γ used. The third and fourth column show the number of iterations (NI) and CPU time (s). The fifth column shows the best feasible solution found (BestSol) and the last column displays the relative optimality gap between this solution and the optimal solution (corresponding to the row $\beta=50.0$ and $\gamma=0$, which means no reduction applied).

As can be seen the quality of the results is very good reporting relative optimality gaps of less than 0.03% in less than 6 min in all cases. Note that the strategy reported in the first row (corresponding to $\beta=3.0$ and $\gamma=0.50$) took less than a minute and found a solution within 0.02% of optimality. The optimal solution (last row) was found in over 30 min of CPU.

In the following experiment we assess the effect of implementing the forced connectivity strategy for the solution of large instances. Thus we apply the method for different values of the parameters, to an instance with 10,000 BUs and 50 territories, fixing $\beta=3.0$. Table 4 and displays the results for the 10,000-BU instance. For this instance size, it was not possible to obtain an optimal solution. Thus, the "Gap" column reflects the relative optimality gap between the primal solution found under the given strategy and the best known lower bound for this instance. This lower bound was obtained by applying the algorithm to this instance with no reduction strategies into effect (that is, $\beta=50.0$, $\gamma=\delta=0.0$) and time limit of 22,000 CPU second as stopping criterion. As can be seen the introduction of this strategy speeds up the algorithm considerably. The quality of the solution is not over-compromised. In fact, sometimes a better solution was found in less computational effort. For instance, for the ($\gamma=0$) case it was observed how the solution improved from 124.150 to 124.142 when switching from $\delta=0.0$ (no forced-connectivity strategy in action) to $\delta=1.0$. This better solution was obtained in almost 50% of the time employed by the $\delta=0.0$ case. As we penalize more, moving from $\delta=1.0$ to $\delta=5.0$ we can see that the solution deteriorates slightly (less than 0.01%) but it is 90% faster. A similar behavior is observed when we look at the $\gamma=0.1$ and $\gamma=0.25$ cases separately. Here, the best solution 124.122 is obtained when activating the forced-connectivity strategy with $\delta=1.0$. Finally, in the $\gamma=0.50$ case, it was observed for the case $\delta=0.0$, it took a large amount of time to find an optimal solution, thus activating the strategy with $\delta\geq 1.0$ helped obtain feasible designs. Overall the best solution was obtained when using

Table 3
Results for instance (5000, 50).

β	γ	NI	Time	BestSol	Gap (%)
3.0	0.50	25	58	62.5027	0.0168
	0.25	38	118	62.5056	0.0214
	0.10	46	158	62.4972	0.0080
	0.00	50	186	62.4978	0.0089
4.0	0.50	44	146	62.5011	0.0142
	0.25	60	262	62.4986	0.0102
	0.10	48	223	62.4972	0.0080
	0.00	54	264	62.4957	0.0056
8.0	0.50	48	330	62.5101	0.0286
	0.25	63	457	62.5002	0.0128
	0.10	37	305	62.4976	0.0086
	0.00	61	576	62.4956	0.0054
50.0	0.00	54	1930	62.4922	0.0000

Table 4
Results for instance (10000, 50).

β	γ	δ	NI	Time	BestSol	Gap (%)
3.0	0.50	50.0	305	947	124.732	0.54
	0.50	10.0	76	243	124.443	0.31
	0.50	5.0	97	404	124.373	0.25
	0.50	1.0	120	1062	124.296	0.19
	0.50	0.0	488	43,174	124.253	0.15
	0.25	50.0	42	139	124.688	0.50
	0.25	10.0	29	114	124.248	0.15
	0.25	5.0	60	233	124.185	0.10
	0.25	1.0	198	1965	124.122	0.05
	0.25	0.0	156	7442	124.185	0.10
	0.10	50.0	46	161	124.670	0.49
	0.10	10.0	33	110	124.225	0.13
	0.10	5.0	47	203	124.171	0.09
	0.10	1.0	106	1026	124.122	0.05
	0.10	0.0	140	4132	124.168	0.08
	0.00	50.0	87	257	124.467	0.33
	0.00	10.0	41	145	124.244	0.15
	0.00	5.0	52	193	124.165	0.08
	0.00	1.0	136	1516	124.142	0.06
	0.00	0.0	94	3040	124.150	0.07

$\delta = 1.0$ and $\gamma = 0.1$ or 0.25, resulting in a relative optimality gap of 0.05%, quiet remarkable. In general, it was observed than activating the forced-connectivity strategy at $\delta = 1.0$ yielded better solutions than the ones obtained when setting $\delta = 0.0$ in both aspects solution quality (reducing the average relative optimality gap from 0.10% to 0.09%) and time employed (reducing the average CPU time from 14,447 s to 1392 s).

In order to show the behavior of the solution method in terms of solution quality versus computational time we plot the following measures: (i) number of unconnected BUs, (ii) number of unconnected territories, (iii) number of cuts added, and (iv) objective function value as a function of the iterations for several configurations of the parameters. Figs. 2–5 display these results for (β,γ,δ) values of (3.0, 0.25, 50.0), (3.0, 0.25, 35.0), (3.0, 0.25, 25.0), and (3.0, 0.10, 7.0), respectively.

As it can be verified in Figs. 2 and 3, the first two runs with a very high value on parameter δ have a similar behavior. The number of unconnected BUs, unconnected territories, and cuts added to the model decrease with the number of iterations. Something similar happens with the objective function value but in the opposite direction. On the other two cases (Figs. 4 and 5) with a lower value of parameter δ, we have a very different behavior. Particularly, the objective function value moves slowly as the time grows. Indeed, this is the reason why lower objective function values are obtained. Either way, it is important to point out that our methodology presents an MIP model that ensures integral assignments at each iteration. Thus, it is interesting to verify how rapidly our heuristic implemented on the allocation MIP model can evolve and converge on solutions with very high degree of connectivity.

We now evaluate the method efficiency when applied to the solution of a 10,000-BU instance with a smaller territory balance tolerance. For this case, we set $\tau^a = 0.05$. This new value for parameter significantly lower than the previous one of 0.10. Thus, we have a very large-scale instance with a very narrow tolerance for territory balancing. This makes the problem extraordinarily difficult to solve. Our results are presented on Table 5. Again, the "Gap" column reflects the relative optimality gap between the solution found and the best known lower bound for this instance. This lower bound was obtained using the by applying the

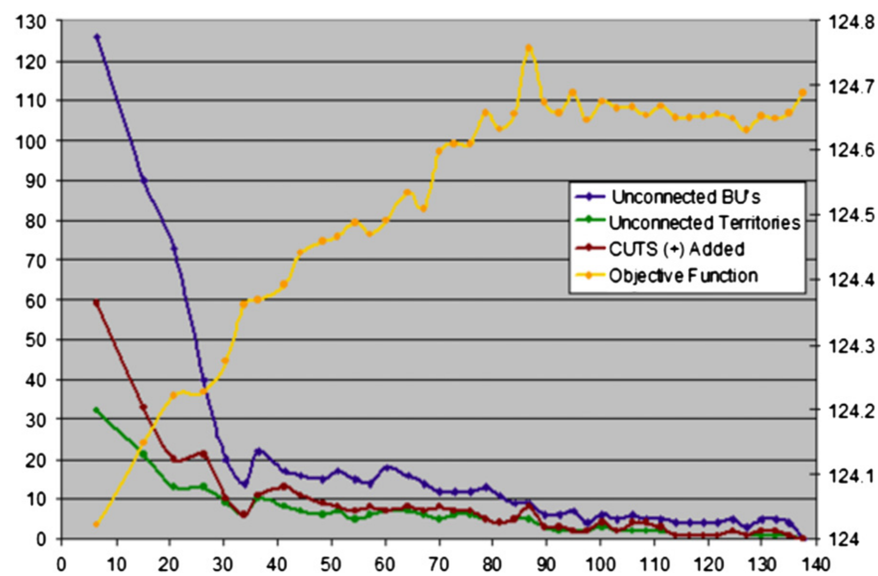

Fig. 2. Algorithm performance for a (10,000,50) instance with $\beta = 3.0$, $\gamma = 0.25$, $\delta = 50.0$.

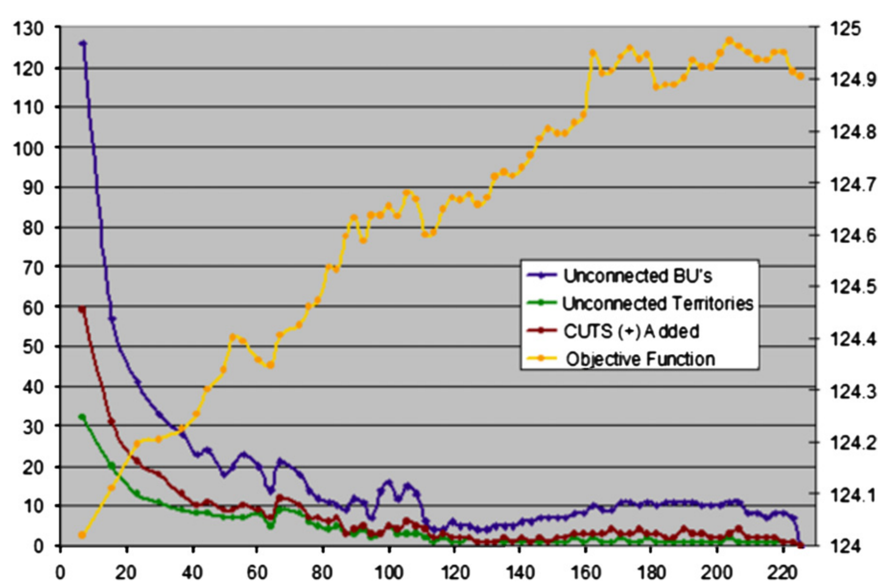

Fig. 3. Algorithm performance for a (10,000,50) instance with $\beta = 3.0$, $\gamma = 0.25$, $\delta = 35.0$.

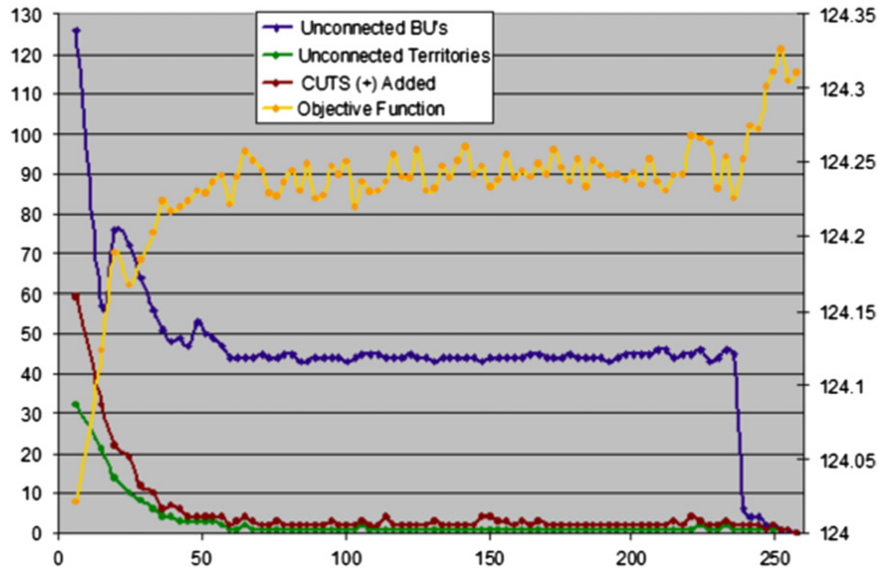

Fig. 4. Algorithm performance for a (10,000,50) instance with $\beta = 3.0$, $\gamma = 0.25$, $\delta = 25.0$.

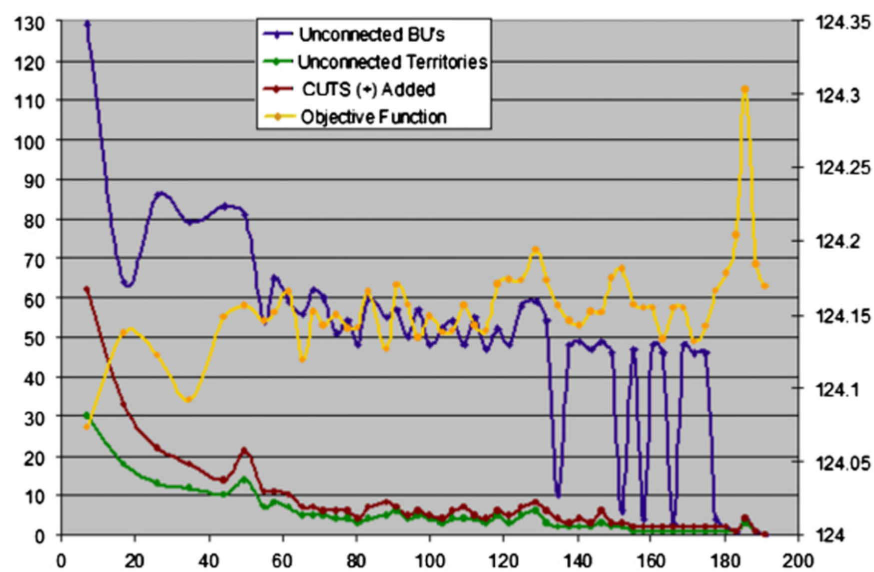

Fig. 5. Algorithm performance for a (10,000,50) instance with $\beta = 3.0$, $\gamma = 0.10$, $\delta = 7.0$.

Table 5
Results for instance (10,000, 50) with $\tau^a = 0.05$.

β	γ	δ	NI	Time	BestSol	Gap (%)
3.0	0.25	15.0	55	424	127.633	0.30
	0.25	10.0	54	689	127.770	0.41
	0.25	7.0	45	800	127.595	0.27
	0.25	5.0	35	615	127.587	0.27
	0.25	3.0	54	874	127.626	0.30
	0.10	15.0	478	1697	127.929	0.54
	0.10	10.0	154	812	127.698	0.35
	0.10	7.0	100	545	127.626	0.30
	0.10	5.0	61	694	127.532	0.22
	0.10	3.0	75	2261	127.543	0.23

algorithm to this instance with no reduction strategies into effect (that is, $\beta = 50.0$, $\gamma = \delta = 0.0$) and time limit of 22,000 CPU sec as stopping criterion.

As can be seen, even in this more difficult case it was possible to obtain feasible designs in very competitive times. The best solution, with a relative optimality gap of 0.22%, was found under the $\gamma = 10.0$ and $\delta = 5.0$ settings showing the success of the introduced strategies for speeding up convergence, and keeping solution quality. Note that all designs obtained fall within 0.6% of optimality.

Finally, Fig. 6 displays the graphical solution of a 5000-BU, 50-territory instance under tolerances of 0.05. This is a feasible solution satisfying all of the planning constraints. The legend besides the graph indicates the number of BUs contained in each territory.

7. Conclusions

In this paper we have addressed a commercial territory design problem motivated by a real-world application in the bottled beverage distribution industry. Planning criteria includes territory compactness, territory balancing with respect to three activity measures, and territory connectivity. In addition, our model incorporates new issues such as disjoint assignment requirements and partial similarity with existing plan. We present a new MILP

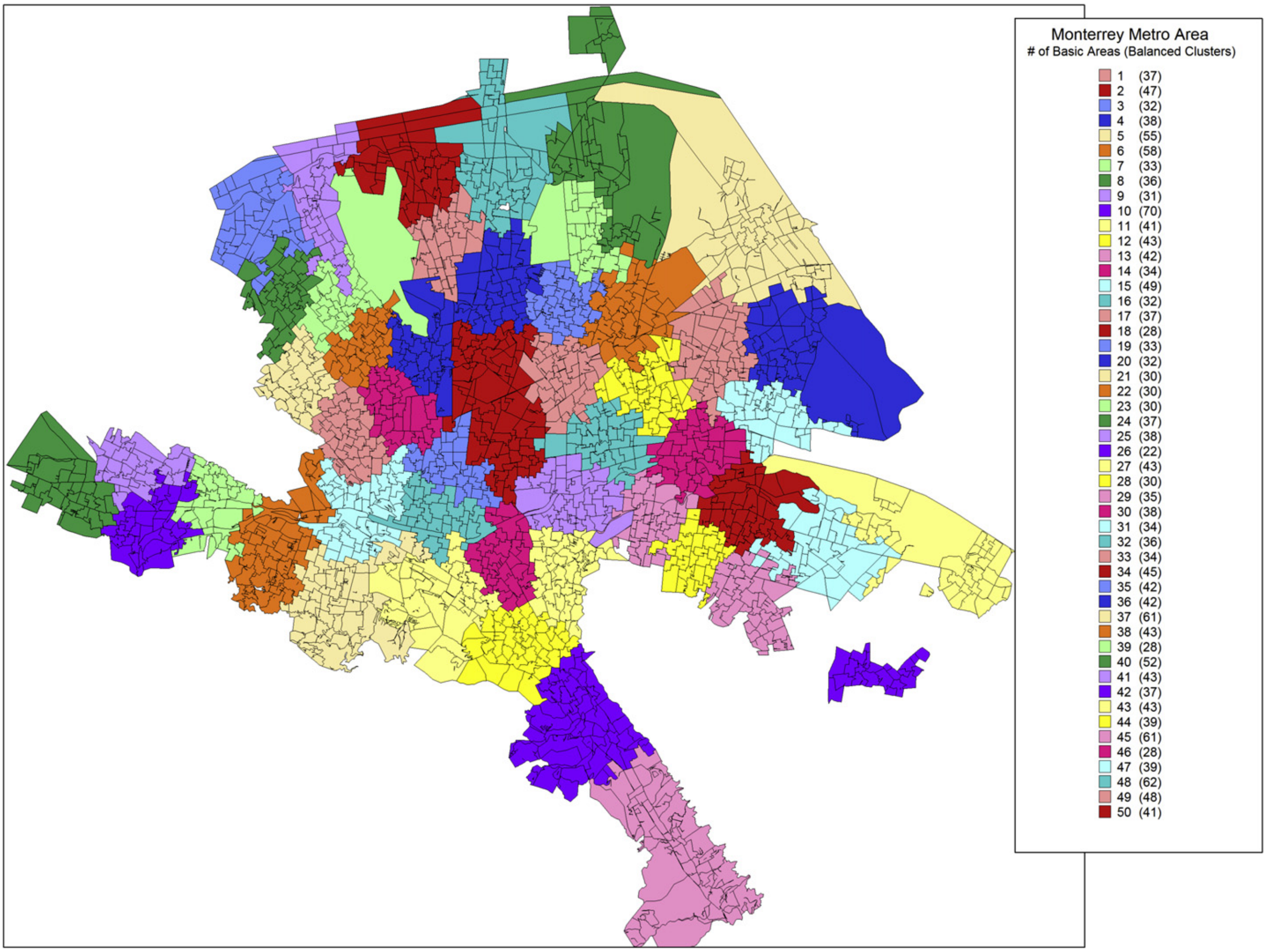

Fig. 6. Visual results of an optimal territory design in Monterrey with 5000 BUs.

model in literature that considers all these issues. A solution framework based on a cut generation strategy within a branch-and-bound algorithm for solving the allocation level for a fixed set of territory centers is developed. This method is intended for solving large-scale instances. The method is enhanced through several algorithmic strategies that help reduce the size of the problem and search space. One added value is a very effective tool which can be relatively easily implemented with off-the-self optimization modeling software such as X-PRESS, GAMS, AMPL. The method and its algorithmic strategies were assessed on large-scale real-world instances. Previous work on heuristics for some related commercial territory design models had reported empirical evidence on instances of up to 2000 BUs in some simplified models. We found the proposed method very successful on handling instance of 5000- and 10,000-BUs, obtaining solutions of very good quality with relative optimality gaps of less than 0.10% and 0.22% for the instances under tolerance levels for the balancing constraints of 0.10 and 0.05, respectively.

There are naturally opportunities for future research. For instance, in this work we focused on solving the allocation problem; however, the results obtained in our research can be used to extend this work to a location–allocation approach where the centers are dynamically updated in an iterative way. This has been done in other similar simpler models. Deriving models and methods for problems with both territory design and routing decisions simultaneously is another very challenging research area. In fact, when one looks at the districting literature in general, one can barely find a very few applications addressing this issue. Finally, in this work we are assuming a deterministic model; therefore the introduction of stochastic models to deal with some parameters such as the product demand becomes a natural extension worthwhile exploring.

Acknowledgments

This research has been supported by the Mexican National Council for Science and Technology (grants SEP-CONACYT CB05-01-48499-Y and CB11-01-166397) and by Universidad Autonoma de Nuevo Leon through its Scientific and Technological Research Support Program (grants UANL-PAICYT CE012-09, CS470-10, and IT511-10). The authors are grateful to three anonymous reviewers for their useful and constructive remarks and suggestions.

References

[1] Salazar-Aguilar MA, Ríos-Mercado RZ, Cabrera-Ríos M. New models for commercial territory design. Networks & Spatial Economics 2011;11(3): 487–507.

[2] Ríos-Mercado RZ, Fernández EA. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research 2009;36(3):755–776.

[3] Deckro RF. Multiple objective districting: a general heuristic approach using multiple criteria. Operational Research Quarterly 1977;28:953–961.
[4] Zoltners AA. A unified approach to sales territory alignment. In: Bagozzi RP, editor. Sales management: new developments from behavioural and decision model research. Cambridge: Marketing Science Institute; 1979. p. 360–376.
[5] Zoltners AA, Sinha P. Sales territory alignment: a review and model. Management Science 1983;29(11):1237–1256.
[6] Kalcsics J, Nickel S, Schröder M. Toward a unified territorial design approach: applications, algorithms, and GIS integration. TOP 2005;13(1):1–56.
[7] Zoltners AA, Sinha P. Sales territory design: thirty years of modeling and implementation. Marketing Science 2005;24(3):313–331.
[8] Duque JC, Ramos R, Suriñach J. Supervised regionalization methods a survey. International Regional Science Review 2007;30(3):195–220.
[9] Mansfield R, Todd D, Wheeler J. Company structure and market strategy. Omega 2003;6(2):133–138.
[10] Vargas-Suárez L, Ríos-Mercado RZ, López F. Usando GRASP para resolver un problema de definición de territorios de atención comercial. In: Arenas MG, Herrera F, Lozano M, Merelo JJ, Romero G, Sánchez AM, editors. Proceedings of the IV Spanish conference on metaheuristics, evolutionary and bioinspired algorithms (MAEB). Granada, Spain; 2005. p. 609–617, [in Spanish].
[11] Salazar-Aguilar MA, Ríos-Mercado RZ, González-Velarde JL. A bi-objective programming model for designing compact and balanced territories in commercial districting. Transportation Research Part C. Emerging Technologies. 2011;19(5):885–895.
[12] Salazar-Aguilar MA, Ríos-Mercado RZ, González-Velarde JL. GRASP strategies for a bi-objective commercial territory design problem. Journal of Heuristics, forthcoming, http://dx.doi.org/10.1007/s10732-011-9160-8.
[13] Salazar-Aguilar MA, Ríos-Mercado RZ, González-Velarde JL, Molina J. Multiobjective scatter search for a commercial territory design problem. Annals of Operations Research, forthcoming, http://dx.doi.org/10.1007/s10479-011-1045-6.
[14] Browdy MH. Simulated annealing: an improved computer model for political redistricting. Yale Law and Policy Review 1990;8:163–179.
[15] Caro F, Shirabe T, Guignard M, Weintraub A. School redistricting: embedding GIS tools with integer programming. Journal of the Operational Research Society 2004;55(8):836–849.
[16] Zoltners AA, Lorimer SE. Sales territory alignment: an overlooked productivity tool. Journal of Personal Selling and Sales Management 2000;XX(3):139–150.
[17] Ronen D. Sales territory alignment for sparse accounts. Omega 1983;11(5): 501–505.
[18] Drexl A, Haase K. Fast approximation methods for sales force deployment. Management Science 1999;45(10):1307–1323.
[19] Hess SW, Weaver JB, Siegfeldt HJ, Whelan JN, Zitlau PA. Nonpartisan political redistricting by computer. Operations Research 1965;13(6):998–1006.
[20] Caballero-Hernández SI, Ríos-Mercado RZ, López F, Schaeffer SE. Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. In: Fernandez JE, Noriega S, Mital A, Butt SE, Fredericks TK, editors. Proceedings of the 12th annual international conference on industrial engineering theory, applications, and practice (IJIE) (Cancun, Mexico); 2007. p. 422–427, ISBN: 978-0-9654506-3-8.
[21] Ríos-Mercado RZ, Salazar-Acosta JC, A GRASP with strategic oscillation for a commercial territory design problem with a routing budget constraint. In: Batyrshin I, Sidorov G, editors. Advances in soft computing. Lecture notes in artificial intelligence, vol. 7095. Heidelberg, Germany: Springer; 2011. p. 307–318.