Lopez, Fabian
Multi-Objective Territory Design with Capacity & Geographic Constraints
International Journal of Combinatorial Optimization Problems and Informatics
Morelos, México

Available in: http://www.redalyc.org/articulo.oa?id=265219741002
Multi-Objective Territory Design with Capacity & Geographic Constraints

Fabian Lopez
Management School of Business, Universidad de Nuevo Leon, Monterrey, México
fabian.lopez@e-arca.com.mx

Abstract. Territory design can be defined as the problem which is focused on grouping small geographical areas into larger geographic clusters. These clusters may be referenced as territories. On this paper we develop the case where two or more attributes must be considered in order to setup an optimal territory design. We developed a new strategy which is based on a hybrid-mixed integer programming method (HMIP). We introduce the model and present in detail our approach for solving the Territory Design Problem. We start from a large group of basic areas (i.e. blocks) which are characterized individually by two or more attributes. Each attribute has a metric defined by a specific activity measure (i.e. quantity of customers, sales volume, etc). In addition there are some geographic restrictions about contiguity and compactness for the territories that should be constructed from these basic areas. An optimal districting plan is obtained when the constructed territories are well balanced taking in mind each activity measure simultaneously. Our methodology includes a mixed integer model definition. We tested our implementation on a large scale instance that was built from a real world application of more than 4000 basic areas. Some computational results about efficiency and suitability are presented.

Keywords: territory planning; multi-objective; mixed integer programming.

1 Introduction

Depending on the context of the problem, the concept “Territory Design” may be used as equivalent to “Districting”. Districting is a truly multidisciplinary research which includes several fields like 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. In addition some capacity constraints are imposed to the problem.

We can generalize that the Territory Design Problem is common to all applications that operate within a group of resources that need to be assigned in an optimal way in order to subdivide the work area into balanced regions of responsibility. In fact, Territory Design Problems emerge from different types of real world applications. We can mention pick up & 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, postal delivery, and others, are administered along territorial boundaries. We can mention either economic or demographic issues that may be considered for setup a well balanced territory. As each territory would be serviced by a single resource, it makes sense to use planning criteria that should be applied to balance the quantity of customers assigned on each territory. Furthermore, other planning requirements could exist to make some balance in terms of sales volumes or even more take in mind the travel time required to cover the entire territory. In addition, there are some other spatial constraints (contiguity and compactness properties) that must be included as part of the geographic definition of the problem.

2 Problem Definition

A Territory Design Problem can be defined as the process of grouping small geographic areas, i.e. basic areas, into clusters. The new geographic clustered areas are named territories. As it is required, we defined that every basic area should be assigned to just one territory. Moreover, we require compactness and contiguity for the territories constructed. Indeed, contiguity can be defined as a territory that is undistorted geographically. In other words, the basic areas that build a territory have to be geographically connected. It is easy to verify that, in order to obtain contiguous territories is required explicit neighborhood information for the basic areas.
Our problem definition includes three quantifiable attributes for each basic area. These quantifiable attributes are named as activity measures, which in turn, are associated with each of the basic areas. The activity measures for each basic area are: (1) quantity of customers, (2) volume of sales and (3) labor time. We defined labor time as the total amount of time that each point of sale request for service which is very dependent on the characteristics of each customer and its geographical location (i.e. downtown or suburbs). The activity measure of a territory is the total activity of the customers which are sum up from the contained basic areas. As we defined before, all territories should be balanced. In fact, this balancing procedure is taking into account each activity measure individually and simultaneously. It is interesting to point out, that only a few authors consider more than one criterion simultaneously for a balanced territory design problem (Deckro (1977) [1], Ross and Zoltners (1979) [2], Zoltners and Sinha (1983) [3]).

We found the main balancing criteria on real world territory design applications is referred as the selling activity it self. Indeed we argue that time travel is only a mean for the actual work to do. The number of territories T to be constructed is predefined and fixed in advance. Our problem definition includes some prescribed and/or forbidden territories. It means that from the beginning we already have some basic areas which require to be assigned to a specific territory. Furthermore, there are other basic areas which are not allowed to be assigned to the same territory. This modeling feature could be applied to take into account geographical obstacles, e.g. rivers & mountains. As can be verified, all these features could be easily extended to consider some territories that may already exist at the start of the planning process. In other words, our method should be prepared to take the already existing territories into account and then add additional basic areas to them. The problem can be resumed as follows: make an optimal partition of a set of basic areas V into a number of territories T which satisfy the specified planning criteria like balance, compactness and contiguity.

3 Overview of Solution Techniques

In Operations Research (OR) the first work about territory design problems can be traced back to Forrest (1964) [4] and to Garfinkel’s (1968) [5]. The recent paper by Kalcsics, Nickel, and Schröder (2005) [6] is an extensive survey on approaches to territory design that gives an up to date state of the art and unifying approach to the topic. For a more extensive review related to the sales territory design problem see Zoltners and Sinha (2005) [7]. Historically, the first mathematical programming approach was proposed by Hess et al. (1965) [8]. The approach they applied was to decompose the location and allocation procedures into two independent phases. In the location phase, the optimal centers of the territories are calculated. For that purpose they used a capacitated P−median facility location method. Afterwards, on the second allocation phase the basic areas were assigned to these clustering centers. The model formulation take into consideration the capacity of the center locations selected. The objective function is to assign each basic area to a unique location among the candidates, so that the demands of the basic areas are satisfied efficiently. The balancing requirement was modeled as a side constraint. Furthermore, compactness and contiguity were modeled by minimizing the sum of distances between basic areas to the territory centers. Due to its combinatorial complexity, the computational implementation of this model was limited.

In general, for solving large scale problems, the allocation phase can be tackled by relaxing the integrality constraints on the assignment variables (i.e. binary variables). However, this procedure usually assigns portions of basic areas to more than one territory center which is not desired. Hess et al. (1971) [9] proposed a simple rule, which exclusively assigns the so-called split areas to the territory center that “owns” the largest share of the split area. Fleischmann and Paraschis (1988) [10] reported poor results with this heuristic. They found that about 50% of the balancing activity constraints for the territories obtained were violated. Zoltners and Sinha (1983) [7] modeled the allocation problem assigning basic areas to the closest territory center. This procedure yields compact and often connected territories, however, usually not well balanced. There are other types of methods called “Divisional”. The basic idea of these types of methods is to iteratively make a partition on the region under consideration into smaller and smaller sub-problems. Iteration stops if a level has been reached where each sub-problem can be solved efficiently.

We can mention another procedure named “Multi-kernel growth”. This method begins with the selection of a certain number of basic areas as “seeds” (i.e. centers) for the territories. Furthermore, one territory after the other is extended at its boundary through successively adding yet unassigned, adjacent basic areas to the territory. The procedure checks for minimal distance and/or better balancing criterion. This procedure stops until the territory constructed satisfies the activity measures constraints. See e.g. Mehrtra et al. (1998) [11]. Marlin (1981) [12] observed that using squared Euclidean instead of straight line distances produces compact but disconnected territories. Hojati (1996) [13] showed that a good selection for territory centers may impact on the resulting territories.
Lopez, F. / Multi-Objective Territory Design with Capacity & Geographic Constraints. IJCOPI Vol. 1, No. 1, May-Aug 2010, pp. 3-12

Algorithms based on simulated annealing are proposed by Browdy (1990) [14], and D’Amico et al. (2002) [15]. Ricca (1996) [16] developed Tabu-search algorithms. The latter technique has been successfully applied in the recent papers of Bozkaya et al. (2003) [17] and Blais et al. (2003) [18]. Genetic algorithms for solving territory design problems have been introduced recently by Forman and Yue (2003) [19] and Bergey et al. (2003) [20]. They encoded the solution in a similar way as is used to solve the Traveling Salesman Problem (i.e. TSP). The encoding is a path representation through each basic area. As the basic areas are traversed, territories are built on this sequence. Haugland et al. (2005) [21] worked with stochastic data which they argue is frequently present in territory design decisions. They deal with uncertain demand for basic areas.

4 Modeling for Territory Design Problem

We developed a new strategy which is based on a hybrid-mixed integer programming method (HMIP). We will introduce the model and present in detail our approach for solving the Territory Design Problem. Hereby, we introduce some notation that will be used as a “building block” for the construction of the model. We start from a quantity of 36,000 physical blocks (N) which constitutes the entire Monterrey metropolitan area. We proceed to transform this problem of size N in one another of size V. In the following we enumerate some basic steps required to complete this process:

1. For each (N) initial block:
   a. Calculate a geographic center defined by (x, y) coordinates.
   b. Define a reduced set of “k1” neighbor blocks chosen by the nearest in terms of Euclidean distance. We implement (k1= 200).
   c. From this point we should have a total of “N x k1” arcs, i.e. 7.2x10+E6 arcs. Let’s name this total set of arcs as “M1”.
2. From set “M1” we make a sort selecting the first “h” arcs with the minimal distance.
   a. Value “h” is calculated as a proportion of the total quantity of “N” initial blocks. We use (h = 18,000) which encompass for the 50% of the initial blocks.
   b. The activity measures of each “h” arc are aggregated into a new meta-node.
3. For each of the “h” redefined blocks:
   a. We identify a reduced set of “k2” neighbor blocks with the minimal distance, (i.e. the nearest). Again, we implement (k2 = 200).
   b. From this point we should have a total of “h x k2” arcs, i.e. 3.6x10+E6 arcs. Let’s name this total set of arcs as “M2”.

As can be verified, on step 1 we filter out the number of blocks and on step 3 we do the same with the arcs that will be passed to the optimization phase afterwards. By this means we reduce the complexity of our problem by limiting the quantity of blocks and reducing the quantity of arcs. However, we must validate that each relevant block “h” has at least one arc on the set of arcs “M2”. This should be considered to connect each “h” block to the entire network. If there is any block without this connectivity feature, a subset of arcs should be added to the set of arcs “M2”. We proceed now with the first optimization phase that is focused on identify a set of “v” blocks from a set of “h” candidates which in turn will be used as center locations for clustering where “V << h << n”.

4. We implement a P-median location optimal model as a pre-processing phase. The objective function is implemented by taking in mind geographical information only. That is, the pre-processing procedure minimize the total distance for each block “i” to the basic area “j” where is assigned. We define the following model (equations 1 to 6).

\[ \begin{align*}
X_{ij} \begin{cases} 0 \rightarrow \text{block } i \text{ is not assigned to basic area } j \\ 1 \rightarrow \text{block } i \text{ is assigned to basic area } j \end{cases} ,
Y_j \begin{cases} 0 \rightarrow \text{basic area } j \text{ is not defined} \\ 1 \rightarrow \text{basic area } j \text{ is defined} \end{cases} , i \in h, j \in h
\end{align*} \tag{1} \]

\[ \sum_{j \in h} X_{ij} = 1, \forall i \in h \tag{2} \]

\[ X_{ij} \leq Y_j, \forall i \in h, j \in h \tag{3} \]

\[ \sum_{i \in h} X_{ij} \leq MY_j, \forall j \in h \tag{4} \]

\[ \sum_{j \in h} Y_j = V, \forall v \in V \tag{5} \]
min(z) = \sum_{j \in h} \sum_{i \in n} X_{ij} D_{ij} \tag{6}

Where: \(D_{ij} = \text{distance from block } i \text{ to basic area } j\), \(V = \text{number of basic areas}\), \(M = \text{max number of blocks assigned for a basic area}\). As a result we obtain \(\text{"V" geographic regions (basic areas)}\). These final basic areas contain the original \(\text{"n" blocks}\) that were heuristically grouped into a set of \(\text{"h" reduced blocks on step 3 which in turn were also grouped on step 4 by the optimization clustering phase}. Each \(\text{"V" region aggregates all the quantitative data that is coming from the original \(\text{"n" blocks}\). This aggregation procedure sums up each activity measure individually. Something similar happens with the geographic information of the new regions \(\text{"V"}\). New geographic borders are constructed for each region \(\text{"V" as a result of the new polygon defined by the original \(\text{"n" blocks}\) that are being grouped}. For our implementation we defined \(V = 4000\), where \(4k << 18k << 38k\). In fact, the new basic areas \(\text{"V"}, \text{where } V \in V\) are geographical regions that are constructed with the purpose to decrease the dimensionality and the combinatorial complexity of the original problem of size \(N\), (where \(V << N\)). Our problem definition establishes \(\text{"V" as a set of basic areas that are covering the entire Monterrey metropolitan area}\). Each basic area \(\text{"V" is represented by its center with coordinates } (X_V, Y_V)\). We proceed now with the territory design. For each final territory to be designed, a center should be computed. Indeed, all basic areas \(\text{"V" can be used as candidate locations for final territories centers}\).

5. For each basic area \(\text{"V"}:
   \begin{itemize}
   \item[a.] Calculate the geographic centroid (center) defined by \((x, y)\) coordinates.
   \item[b.] Define a reduced set of \(\text{"k3" neighbor basic areas identified as the nearest in terms of Euclidean distance}\). We implement \((k3 = 200)\).
   \item[c.] From this point we should have a total of \(\text{"v * k3" arcs, i.e. 400,000 arcs}. \text{Let’s name this total set of arcs as } \text{"M4"}\).
   \end{itemize}

6. From set \(\text{"M4"} we select the first \(\text{"h3" arcs with the minimal geographical distance}.\)
   \begin{itemize}
   \item[a.] Value \(\text{"h3" is defined as a proportion of the total set of } \text{"M4"}. Again, we can use a factor which encompass for the 50\% of the } \text{"M4"}, (i.e. \(h3 = 200,000\)). \text{Let’s name this reduced set of arcs as } \text{"M5".}
   \item[b.] Again, we must assure that each basic area \(\text{"V" has at least one arc on the set of arcs } \text{"M5" that allows its connectivity to the entire network. If there is any block without connectivity, a sub-set of arcs should be added to } \text{"M5".}\)
   \end{itemize}

We proceed now with the second optimization phase that is focused on identify a set of \(\text{"k" basic areas from a set of } \text{"V" candidates, where } \text{"k << v". As a result we obtain } \text{"k basic areas that are selected as centers for final territories. That is, we start now from } \text{"v = 2k basic areas} to finally \text{"k = 50 final territories". We denote by } \text{"C } \subset V \text{ the set of final territory centers. We should have a total of } \text{"v*k" arcs, i.e. 100k}. \text{Let’s name this set of arcs as } \text{"Div"}, \text{distance between basic area } \text{"V" and the territory center } \text{"i".}\)

7. We use the same P-median location model as was applied on step 4. Basic areas \(\text{"V" should be assigned to the territories centers } \text{"i" in such a way that all } \text{"k" territories constructed must be well balanced}. This balancing procedure is taking in mind each activity measure individually and simultaneously. Indeed, the activity measure of a territory is the total activity measure of the contained basic areas. Formally, the activity measure of \(\text{Ti is defined as follows:}\)
   \[W(\text{Ti}) = \sum_{\text{Wv}}\]
   \[\text{(7)}\]

It is easy to verify that a perfect balanced territory plan cannot be accomplished. This is true, because of the discrete structure of the problem and the unique assignment assumption. In order to calculate a measure of balance among territories, we compute a reference or target for each activity measure. This reference can be defined as an average for each activity measure. Formally we have:
   \[\mu_m = \frac{\sum_{\text{Wv}}\text{Wv}}{k}\]
   \[\text{(8)}\]

We calculate a relative percentage deviation on each territory and for each activity measure. A set of constraints are formulated to ensure that each territory is within a maximal deviation from the target already defined. Moreover, we define a specific constraint for each activity measure in order to set a balance among territories within predefined lower and upper bounds. These bounds are adjusted as a parameter by the decision maker. Formally we have:
   \[\sum_{i \in k} X_{ij} = 1, \forall j \in V; X_{ij} \begin{cases} 0 & \text{basic area } j \text{ is not assigned to territory } i \\ 1 & \text{basic area } j \text{ is assigned to territory } i \end{cases}\]
   \[\mu_m * (1 + \text{Tol}_m) \geq \sum_{j \in V} X_{ij} W_{mj} \geq \mu_m * (1 - \text{Tol}_m), \forall i \in k, m \in M\]
   \[\text{(9)}\]
   \[\text{(10)}\]

Where: \(M = \text{quantity of activity measures to balance}, \text{Tol}_m = \text{maximal \% deviation for activity measure } \text{“m”}\).
Each territory to be constructed defines a partition of the set \( \mathcal{V} \) of basic areas. Let’s define \( T_i \subseteq \mathcal{V} \), where \( T_i \) defines the “\( i \)-th” territory constructed. Then we can generalize:

\[
T_1 \cup T_2 \cup \ldots \cup T_k = \mathcal{V} \quad \text{and} \quad \bigcap_{i \neq j} T_i = \emptyset
\]

(11)

About compactness issues there were proposed on the past some ideas to evaluate this measure, Niemi et al. (1990) [22] and Horn et al. (1993) [23]. Compact territories usually have geographically concentrated operation, therefore we can expect less travel and better service levels because of a more available time to attend the customers. We can calculate a measure of compactness for a region if its area is divided by the perimeter described by the region itself. Although we may extend our cut procedure that will be introduced on the next section to model compactness as geometry measures like perimeter or area; we will not incorporate this feature here. Instead, we model compactness as an objective function. That is, we minimize the total sum up of Euclidean distances from each basic area to the territory centers where they are assigned.

\[
\min(z) = \sum_{i \in k} \sum_{j \in \mathcal{V}} X_{ij} d_{ij}
\]

(12)

Where: \( d_{ij} \) = distance from basic area \( j \) to territory center \( i \). Furthermore, this network \( d_{ij} \) is redefined by eliminating some arcs \((i,j)\) such that \( d_{ij} \) will not be greater than a certain upper bound. The basic idea is to include only a small percentage of arcs as decision variables in our model. This can be implemented by imposing an upper bound for arcs distance. With this in mind, the problem can be solved more efficiently. However, we found it quite problematic to estimate in advance the number of arcs necessary for each basic area and for each territory in order to keep the problem feasible. For that reason, we propose a different strategy on step 8 in order to produce very fast an initial reduced network which can be modified afterwards by the algorithmic procedure depending on the results produced.

8. Now we proceed to construct a set of binary decision variables in order to allocate a reduced network. Our computational implementation includes a parameter that controls an upper bound for the activity measure size that each territory may request to allocate a sub set of decision variables. Our procedure calculates this parameter “\( F_1 \)” as a percentage of the quantity of “\( k \)” new territories to be constructed. As a result “\( F_1 \)” factor identify the quantity of basic areas that will be available to assign for each territory. In addition, we implement a parameter “\( F_2 \)” to identify from the available basic areas, which basic areas can be fixed as a kernel for each territory. To construct the kernel on each territory we implement two different types of constraints. The first type of constraint is named as “soft” because is designed to be modified afterwards by the algorithmic procedure depending on the results produced. No further modifications occur on hard constraints. Formally we have.

\[
F_1 = 10\% \times k = 5 \quad \text{! Reduced binary matrix factor}
\]

\[
F_2 = 25\% \quad \text{! Kernel territory allocation factor}
\]

INPUT : activity . measure (m) sorted by : territory (i), \( d_{ij} \)

for each territory \((i)\) \( \forall i \in k \)

do while activity . measure (m) for territory \((i)\) \( \leq F_1 \times \mu_m \)

define Binary \((X_{ij})\)

if basic . area \((j)\) not allocated then

if activity . measure (m) for territory \((i)\) \( \leq F_2 \times \mu_m \) then

allocate basic . area \((j)\) on territory \((i)\)

if \( d_{ij} = 0 \) then

\( X_{ij} = 1; \quad \alpha = j \quad \text{! Hard constraint} \)

else \( X_{ij} = X_{i \alpha} \quad \text{! Soft constraint} \)

9. In order to obtain contiguous territories, explicit neighborhood information for the basic areas is required to be considered on the optimization phase. Our computational implementation incorporates this graph information based on a cut generation phase. That is, iteratively we add some relevant cuts on the primal problem. The basic idea of our method is to recursively check for the contiguity constraints that are required to impose on each territory. On each iteration, the procedure evaluates if all the territories obtained satisfy the contiguity constraint. For each territory that violates this condition, an algorithmic procedure formulates additional geographic constraints in order to setup a new
incremental mathematical model. This procedure iterates until no additional contiguity constraints are needed and therefore territory design problem is finally solved. Formally we have:

```
sub cut.procedure()
    perturbation = 0; contiguity.ok = true
    for each territory(i) ! \( \forall i \in k \)
        if fail contiguity on territory(i) then
            contiguity.ok = false
            for each not allocated(basic.area(j)) and not adjoin on territory(i)
                look for best territory (g) to assign basic.area(j)
                Rule1: not allocated(basic.area(h)) adjoin to basic.area(j)
                Rule2: territory(basic.area(h)) \( \neq i \) and exists \( \forall j \) territory(basic.area(h)) \( \neq j \)
            for each not allocated(basic.area(h)) adjoin to basic.area(j)
                and territory(basic.area(h)) = g
            define new.cut (Xg,j = Xg,h)
            if not exists(new.cut) on constraints.pool then
                ! check inconsistency on new.cut before add new.cut
                for each cut(z) on pool where territory(cut(z)) \( \neq g \)
                    if basic.area1(cut(z)) = j and basic.area2(cut(z)) \( \neq h \) or
                        basic.area1(cut(z)) \( \neq j \) and basic.area2(cut(z)) = h then
                        delete cut(z) on pool
                end for
            end if
            add new.cut on pool; z = z + 1
            if not new.cuts.added for basic.area(j) then
                if basic.area(j) is adjoin on territory(i) then
                    perturbation = perturbation − Xij
                if not hard.allocated(basic.area(j)) then
                    deallocate.basic.area(j)
                end if
            end if
        end if
    end for
end sub

main()
do while true
    minimize: \[ \sum_{i \in k} \sum_{j \in V} X_{ij} \cdot d_{ij} + \text{perturbation} \]
goto cut.procedure
    if contiguity.ok then end
```
5 Results and discussion

We present some computational results indicating the efficiency of our method for solving large scale instances around 4000 basic areas. The number of territories we select for our test was 50, such that on average 40 basic areas comprised each territory. The computational results are presented on table 1. We test some combinations on network reducing factor (F1) and kernel factor (F2). For each combination we measure the computational time (minutes) that was required to reach optimality. CPU configuration used for the experiment was Win x64, 2 processor Intel Core at 1.4GHz.

<table>
<thead>
<tr>
<th>NETWORK REDUCING FACTOR (F1)</th>
<th>PERFORMANCE EXECUTION TIME TO OPTIMALITY (MINUTES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8%</td>
<td>65%  60%  50%  30%  15%  0%</td>
</tr>
<tr>
<td>10%</td>
<td>12.6  8.6  1.7  1.8  16.0  9.2</td>
</tr>
<tr>
<td>12%</td>
<td>16.2  15.4 1.7  13.0  33.7  11.0</td>
</tr>
<tr>
<td>14%</td>
<td>16.1  26.4 1.8  16.2  31.5  13.6</td>
</tr>
<tr>
<td>16%</td>
<td>16.9  27.7 1.8  17.6  36.5  15.7</td>
</tr>
</tbody>
</table>

As can be seen on table 1, we verify that small values on F1 factor and large numbers on F2 factor results on a better computational performance. However, as long as we use a larger factor on F1 we found not useful anymore to apply a large value on F2 kernel factor. We observe that the algorithm take a lot of unnecessary time to calibrate the “optimal” kernel factor (F2). As follows on table 2 we expose some computational metrics that can be useful to describe the behavior of our algorithm. We use 8% as a network reducing factor (F1). We report the number of iterations that was required to optimality, initial and final size of the kernel (F2) and the number of added and deleted constraints that are maintained on the cut pool.

We found that the number of iterations and the computational time are related. As long as we have a larger kernel the algorithm requires a larger quantity of iterations to converge. However, a larger kernel speeds up the algorithm efficiency on each single iteration, in other words, we can process more iterations by unit time. The difference between initial & final kernel factor is related to the computational time we spend to set the kernel required to converge. Something similar exists with cut added and cut deleted statistics. A cut degradation coefficient is obtained as a division of number of cuts deleted divided by cuts added. We believe that in general, we obtain better results with a small measure on this coefficient. Finally, we observe that when we don’t use the kernel factor at all, i.e. F2 = 0%, the quantity of binary arcs we require to expand the original network is close to zero. Something very different happens when we use a larger kernel factor, this is true because we need to add more arcs to assure problem feasibility.
Our computational results are shown only to give some insight about the effectiveness that we have seen during our experiments. This lack of information will be overcome in a subsequent paper. We finish our report showing the geographical output that is obtained for our territory planning instance. This procedure is considering Monterrey metropolitan area that is constructed from 4,000 basic areas. As we pointed out before, our experiment was performed to make up a well balanced territory planning taking in mind three activity measures at the same time (1) quantity of customers, (2) volume of sales and (3) labor time. On figure 1 we graph 50 clusters. Each cluster corresponds to a single territory. Moreover, each territory complies within (+/-) 10\% tolerance (i.e. from 90\% up to 110\%) in reference to the goal we define for each activity measure. Contiguity and compactness properties can be verified for each territory obtained. Although compactness constraint was not explicit modeled, however we take advantage from the objective function implemented. That is, our HMIP implementation assigns basic areas to clusters seeking to minimize the total Euclidean distance from each basic area to the centers defined for each cluster.
The legend besides the graph indicates the quantity of basic areas contained on each cluster (i.e. territory). As expected, we didn’t found any balancing representation regarding this measure because it was not considered a component of the problem definition.

6 Conclusions

This paper is focused on solving efficiently special instances for the Territory Design Problem. Some modeling elements include balance, compactness and contiguity. We found that, contiguity is typically hard to represent mathematically. Within Operations Research various algorithmic approaches have been proposed, some based on integer linear programming, others on classical heuristics and, more recently, on some meta-heuristics. All previous research works indicated that this NP-Hard problem is not suitable for solving on large-scale instances. In fact, this problem has a MILP formulation with $O(n^2)$ binary variables and a very weak LP relaxation, so TDP is NP-hard since we can reduce it to the well-known Partition Problem. We detailed out our methodology and then we proceed to its computational implementation. The preliminary results are satisfactory.

Our methodology presents a technique which ensures integral assignments all the time at each iteration. Furthermore we can deploy our model to handle several activity measures simultaneously. In fact, our method is not focused on minimize the total activity measure deviation for each territory. Instead, our objective function is focused on minimize the total geographic distance that exists on each territory (i.e. compactness). The proposed approach can extend the basic model to address different specific business rules or additional planning criteria. This can easily be modeled as activity measures on the basic areas. In addition, our model is suitable to handle different values on lower and upper tolerances for each activity measure, which is very common in real applications.
Our model contribution can be used as a frame to generate specialized mathematical cuts for different and more complex planning constraints. The methodology does not start from a set of single customers. In fact, demand points are first aggregated into small groups (i.e. basic areas) that serve as basis for the construction of final territories. As a result, depending on the level of detail or aggregation, we can reduce the mathematical complexity of the problem. It is clear the convenience to integrate a Geographic Information System environment (GIS) with the algorithmic process to complete a Territory Design framework. Finally, we conclude that all districting problems are multicriteria in nature. Depending on the specific application we can define which attributes may be modeled as hard constraints and which should be optimized.

References