An Adjusted Variable Neighborhood Search Algorithm Applied to the Geographical Clustering Problem

Beatriz Bernábe[1], Maria Osorio[2], Javier Ramírez[3], José Espinosa[4] and Ricardo Aceves[5]

[1,2,4] BUAP, Benemérita Universidad Autónoma de Puebla, Puebla, México
{Bernábe, Osorio, Espinosa}
Beatriz Bernabe, Facultad de Ciencias de la Computación
Maria Osorio, Facultad de Ingeniería Química
José Espinosa, Facultad de Ciencias Físico Matemáticas
[3] UAM, Universidad Autónoma Metropolitana, Departamento de Sistemas, México
[1,5] UNAM Universidad Nacional Autónoma de México, Departamento de Sistemas, México

Abstract. This paper describes the application of the Box-Behnken experimental design technique in the response surface methodology to find the best value for the parameters in the Variable Neighborhood Search algorithm for the geographical clustering problem. The solution of this problem demands a zone classification process where each zone is made of objects that best fulfill the objective, usually the minimum accumulated distance from the objects to the centroid in each zone: informally this process is named the geometric compactness. This well known application is an NP hard combinatorial problem. In this paper, we present the use of Variable Neighborhood Search VNS that has proven to be one of the best methods in the heuristic resolution of combinatory problems [9,10], but as a heuristic methodology, the conflict is centered in evaluating the quality of the solutions obtained and their corresponding parameters value [1].

Keywords: Geographical clustering Problem, Variable Neighborhood Search, Box-Behnken, Experimental Design.

1 Introduction

The geographic clustering problem (GCP) consists in the classification of objects in geographic units that fulfill a certain objective, mainly the geometric compactness [7,18,19]. The geographic units that have been considered correspond to AGEBs (Basic Geostatistic Areas) of the metropolitan zones in Toluca Valley MZTV [20].

The GCP problem belongs to the Territorial Design TD category and it is understood as the problem of grouping small geographic areas (basic areas) in greater geographical clusters called territories, in such a way that the acceptable grouping is the one that fulfills certain predetermined criteria [19]. The criteria or properties to fulfill in GCP problems depend on the space restrictions as continuity and geometric com-

Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 113-125

Received 15/03/09
Accepted 27/04/09
Final version 06/05/09

pactness [5,6,7,13,14,19]. The NP-hard condition of the GCP, implies solving a great number of geographic tasks that emphasizes the classification process directed toward the fulfillment of an objective.

Therefore, this problem is usually explained with a description oriented towards an optimization objective modeled mathematically as a cost function accompanied by the characteristics of the problem expressed as constraints. The NP nature of this classification problem, justify the utilization of a heuristic methodology to obtain a solution approximated to the optimal one [15] . The GCP is a special case of the classic clustering problem [7], but under the fulfillment of compactness, connectedness and/or homogeneity in some cases [19].

There are interesting works on classification under the criteria of minimal distances that have partially supported this paper [7,16] but they do not offer systematic methods to help to fit the parameters in a heuristic procedure according to the quality of the solutions offered [1,2,3]. In order to solve the problem, we selected PAM (Partitioning Around Mediods) [11] because is an exact and good partitioning algorithm that can be easily implemented and applied to the AGEBs, in order to obtain optimal solutions for small problems. We used the optimal solutions obtained and compared them with the solutions generated by Variable Neighborhood Search VNS for the GCP and used those results in the Box Bhenken experimental design developed for this paper.

To solve the GCP we developed our own partitioning algorithm that minimizes the distances between objects in order to obtain compactness between the AGEBs. However, the primary target of this research goes beyond revealing the solutions generated by the VNS, in this sense that we have applied a statistical methodology of Box Behnken experimental design to find the parameters that best directs the heuristic procedure to obtain high quality solutions because of its proximity to the optimal solution.

Since there are not clear methodologies to determine the right parameters for heuristic procedures as the VNS, our main contribution is centered exactly in this point, in the search and control of the statistical properties of the parameters in a VNS procedure, under a systematic process that allows us to observe the quality of the results for every combination in the design. In this context, this work presents a precise way to choose the correct parameters that lead to the generation of solutions of good quality.

This document is organized in 4 sections. The introduction is presented in section 1, in section 2 we present the Mathematical model for the geographical clustering problem and the solution and the Variable Neighborhood Search algorithm is presented in section 3. Section 4 describes the results obtained with the model and the validation of the parameters variation; and finally the conclusions are in section 5.

2 A Mathematical Model for the Geographic Clustering Problem

Many approaches have been used to solve the geographic clustering problem (CGP). The method utilized in this research to solve the AGEBs conglomerate design is simi-

lar to the method presented in [7], where the authors implemented a genetic algorithm for a similar zone design problem.

In the Geographic Clustering Problem solved here, the AGEBs are geographical units where each AGEB is separated by different distances of non uniform geometric structure, because the AGEBs are spatial data [14] and its geographical localization is given by latitude and longitude, that made easier the calculation of the distances between them. The AGEBs are clustered in a way that the AGEBs composing such groups are very close geographically, in order to minimize distances between them.

Basically, the strategy is to randomly choose AGEBs as centroids to identify the groups. Those AGEBs that are not centroids and have the shortest distance to a specific centroid-AGEB are members of a group or a cluster. This informal idea is the definition of geometrical compactness.

We did not define compactness in a formal way before, but the definition of compactness of geographic units is included in Definition 1:

Definition 1 **Compactness**

Let Z={1, 2, ..., n} be the set of n objects to classify; the objective is to divide Z in k groups $G_1, G_2, \ldots, G_k$ with $k<n$, such that:

- $\cup_{i\ 1,k}\ G_i = Z$
- $G_i \cap G_j = \varnothing$ $\quad i \neq j$
- $|G_i| \geq 1$ $\quad i = 1,2, \ldots, k$

A group G_m with $|G_m| > 1$ is compact, if for every object $t \in G_m$ satisfies:

$$\underset{t \in Gm,\ i \neq t}{Min}\ d(t, i) < \underset{j \in Z\text{-}Gm}{Min}\ d(t, j)$$

A group G_m with $|G_m| = 1$ is compact only if its object t satisfies:

$$\underset{i \in Z - \{t\}}{Min}\ d(t, i) < \underset{j,l \in Gf,\ \forall f \neq m}{Min}\ d(j, l) \qquad (1)$$

The neighborhood criterion between objects needed to achieve the compactness is given by the pairs of distances described in (1). Using this definition of compactness we will proceed to describe the model for the Geographic Clustering Problem (GCP).

2.1 Model for the Geographical Clustering Problem (GCP)

Data

UG= total of AGEBS

Let the initial set of n geographical units be

UG=$\{x_1, x_2, \ldots, x_n\}$, where

x_i is the i^{th} geographical unit (i=UG index)

k is the number of the zone (group)

The following variables are defined to refer to the different groups:
Z_i is the set of geographical units that belong to the i^{th} zone
n is the number of geographical units
C_i is the centroid
$d(i,j)$ is the euclidean distance from node i to node j (from one AGEB to another)

Constraints
$Z_i \neq \emptyset$ for $i = 1,\ldots, k$ (nonempty groups)

$Z_i \cap Z_j = \emptyset$ for $i \neq j$ (The same AGEBs cannot be in different groups)

$\cup_{i=1,k} G_i = U_g$ (The union of all the groups are all the AGEBSs

Objective Function Once the number of centroids (k) is decided (C_i with $i = 1,\ldots k$), the centroids will be randomly selected and the AGEBs will be assigned to the nearest centroids. Each AGEB i is assigned to the nearest centroid C_i. Then, for each AGEB i:

The objective function is the minimum of the sum of the distances between the centroids (for each k) and the AGEBs assigned to them. Each AGEB is assigned to the closest centroid (C_i).

$$\min_{i=1,\ldots,k} d(i, C_i)$$

For every k (where $k=1,\ldots,n$) the sum of the distances from every AGEBS assigned to each centroid is calculated and the minimum is selected. Therefore the objective function can be written as:

$$\min_{k=1,\ldots,n} \left\{ Min \left\{ \sum_{i=1}^{k} \sum_{i \in c_i} d(i, c_i) \right\} \right\} \qquad (2)$$

3 The Variable Neighborhood Search (VNS)

The Variable Neighborhood Search (VNS) metaheuristic, proposed by Hansen and Mladenovic [9,10] is based in the observation that local minima tend to cluster in one or more areas of the searching space. Therefore when a local optimum is found, one can get advantage of its contained information. For example, the value of several variables may be equal or close to their values in the global optimum. Looking for better solutions, VNS starts exploring, first the nearby neighborhoods of its current solution, and gradually the more distant ones. There is a current solution Sa and a neighborhood of order k associated to each iteration of VNS. Two steps are executed in every iteration: first, the generation of a neighbor solution of Sa, named $Sp \in Nk$ (Sa), and second, the application of a local search procedure on Sp, that leads to a new solution Sol. If Sol improves the current solution Sa, then the searching proce-

dure will start now from G using $k = 1$. Otherwise, $k = k + 1$ and the procedure is repeated from Sa. The algorithm stops after a certain number of times that the complete exploration sequence $N_1;N_2; \ldots ;N_{kmax}$ is performed. The following algorithm shows how the solutions are obtained.

Two steps are executed in every iteration: first, the generation of a neighbor solution of Sa, named $Sp \in Nk\ (Sa)$,, and second, the application of a local search procedure on Sp, that leads to a new solution Sol. If Sol improves the current solution Sa, then the searching procedure will start now from G using $k = 1$. Otherwise, $k = k + 1$ and the procedure is repeated from Sa. The algorithm stops after a certain number of times that the complete exploration sequence $N_1;N_2; \ldots ;N_{kmax}$ is performed. The following algorithm shows how the solutions are obtained.

***Procedure** Variable Neighborhood Search (VNS)*

```
BEGIN
/*  Nk : k = 1, ..., kmax, neighborhood structures */
/*  Sa  : current solution */
/*  Sp  : neighbor solution of  Sa */
/*  Sol: local optima solution */

REPEAT UNTIL (End) DO
    k ← 1
    REPEAT UNTIL (k ← kmax) DO
/*  Generate neighbor Sp of the k^th neighborhood of
        Sa (Sp ∈ Nk (Sa))*/

        Sp ← GetNeighbor (Sa, Nk);
        Sol ← LocalSearch (Sp);
        IF (Sol is better than Sa) THEN
            Sa ← Sol;
        ELSE
            k ← k + 1
        ENDIf
    ENDDO
ENDDO
END
```

Partitioning algorithms match the objective function (2) in order to minimize the distance of the objects to their centroids. The geographic clustering algorithm is following the same objective with the use of VNS. The following pseudocode is commented in order to highlight the performance of both cycles and the search of the minimum distance in the objective function.

3.1 VNS Algorithm for the Geographical Clustering Problem (GCP)

Let n be the number of objects to classify
UG_{ij} denotes that the object i is assigned to the centroid j for i=1,..,n; j=1,...,k
Let M={M_1,M_2,,M_k} be a solution of K centroids

MaxVNS /*maximum number of iterations to go over all the neighborhood search */

MaxLS /*number of iterations of Local Search (LS) for each neighborhood */

1. Initialization

```
/* Get an initial solution */
Generate initial random centroids M = {M1, M2, ..., Mk}
/* Any AGEB can be a randomly obtained centroid */
BEGIN
       Current _cost ← Cost (M)
/* Another solution is generated and compared with the current so-
lution.  The  best solution is stored  */
cont←1
WHILE cont < MaxVNS DO
    BEGIN
       k-neighborhood ← 1
       /* Control variable */
       WHILE kneighborhood <> n DO
         BEGIN
              C ← Generates a random solution with a k-
                 neighborhood
              /* Gets a neighbor of k-neighborhood */
              Sol_neighborhood← LocalSearch (C);
              IF(Cost (Sol_neighborhood)<current_cost) THEN
                     M ← Sol_neighborhood;
              ELSE k-neighborhood ← k-neighborhood +1;
         ENDWHILE
/*Go to the next neighborhood only if the current
  solution(M) is not improved */
    ENDWHILE
       Cont ← cont+1
END
Return (M)  /* Solution with the minimum cost */
```

2. Cost Function (Sol)

```
/* Determine the quality of the solution  SOL, i.e. how much the
objective is minimized */

BEGIN
i ← 1
/* Initialize the first object */
 cost ← 0
WHILE (i ≤ n) DO
 BEGIN
/* For each object in Ug do */
       IF (Ugi is not a centroid) THEN
       BEGIN
           dmin ← dist(Sol1 , Ugi )
/* Represents the distance between the object and the Sol1 (first
centroid where Sol represents the set of centroids). The distance
between each object and its nearest centroid is calculated */
           j ← 2
/* Go to the second centroid  */
             WHILE (j ≤ k) THEN
            BEGIN
                    IF (dist (Solj , Ugi) < dmin) THEN
```

```
/* Calculate the distance between the object i and the Sol_j
   (another centroid) */
                        dmin ← dist (Sol_j , Ug_i)
                    ENDIF
                    j ← j + 1
/* Go to the next centroid  */
                ENDWHILE
               cost ← cost + dmin
        ENDIF
       i ←i + 1
 ENDWHILE
Cost(Sol) ← cost
END
```

The Local Search (LS) algorithm improves the current solution searching in its neighborhood. It can finish finding a better solution or reaching the maximum number of iterations. The maximum number of iterations avoids cycling in the case that a better solution cannot be found.

4 Experimental Design

This section presents the necessary conditions to diminish the compactness between AGEBs, used as a function cost in the CGP solved with the VNS heuristic. The control variables used are the neighborhood structures (NS), the local search iterations (LS) and the number of groups (G) to consider. The quality of the results is systematically evaluated to identify the influence of the control parameters on the cost function and to model the dependency, exploring its influence in the obtaining of local optima solutions.

The experiments were performed in a computer with an Intel Centrino® processor, speed of 1.4 Ghz 768MB in RAM memory and 80 GB of Hard Disk memory. We tested 171 variables in 473 AGEBs. Each AGEB includes data of 55 blocks, in average.

An experimental design of answers' surface with a set of tests that deliberately change some variables with other remaining fixed in the system allowed us to observe the changes in the output variables and to explore the effects described in the previous paragraph.

4.1 Response Surfaces

The methodology of response surfaces is a combination of techniques of design and analysis of experiments that used in a sequential way, allow researchers, the determination of the operation conditions that produces solutions near to the optima [12].

A smooth complex function can come near locally (in "small" zones of the operation region) using polynomials of low order. If the zone where the local approach is made is "far" from the zone where the maximum is found, a polynomial of first order is a good approach. However, if the zone is "close" to the zone where the maximum is, it is necessary to use a polynomial of second order to describe the function.

A systematized analysis can be developed using a Box-Behnken design type. This type of design, due to its characteristics is easy to carry out for defined and adapted

levels of the design parameters; besides, a rotary design with equal variance can be made for all points in the experiment that are equidistant to the center of the design region. On the other hand it is possible to make sequential experiments in zones that we pruned in order to study the individual effects of the control parameters and the combination of the synchronized effects [12].

Another advantage of this design is that the results can be modeled with a second order function and an analysis of the behavior of the cost function can be modeled using the methodology of the response surfaces.

A Box-Behnken design for five parameters of control is used in an experiment with 15 combinations and four central points. The results obtained with the heuristic method are used for choosing the levels of the parameters and the definition of the experimentation region. The parameter levels used in the experiment can be seen in Table 1. The nomenclature used in the tables 1 and 2, is: NS (neighborhood structures), LS (Local Search), G (Groups), FC (Cost Function).

The Box-Behnken's matrix of parameters design are: Factors = 3, Replicates = 1, Base runs = 15; Total runs = 15; Base blocks = 1; Total blocks = 1; Center points = 3. With these levels and parameters, 15 experimental combinations were tested.

In the test with Standard Order of 8 we observed that with 24 groups and parameters of LS = 530 and NS = 640 the cost function is 10.8371. This is the closest value to the optimal objective of 9.279 obtained with PAM. In contrast, PAM managed to get that solution in 27 hours and our VNS algorithm reduced considerably the computational time to 13 minutes, with 616529 iterations and 16 accepted solutions. The behavior of the objective value, represented with the cost function, according to the number of iterations for the VNS heuristic can be seen in Fig. 1.

Table 1. Levels used during the experimentation.

Parameter	High Level	Center Level	Low Level
LS	848	530	212
NS	640	400	160
G	24	18	12

Table 2. Experimental tests for BB.

Std Order	Groups	LS	NS	FC
15	18	530	400	12.6586
2	24	212	400	10.9011
4	24	848	400	10.8866
1	12	212	400	15.4535
5	12	530	160	15.3206
7	12	530	640	15.3221
14	18	530	400	12.5667
6	24	530	160	11.0177
13	18	530	400	12.5597
10	18	848	160	12.4411
11	18	212	640	12.6957
8	24	530	640	10.8371
3	12	848	400	15.1598
12	18	848	640	12.4726
9	18	212	160	12.8541

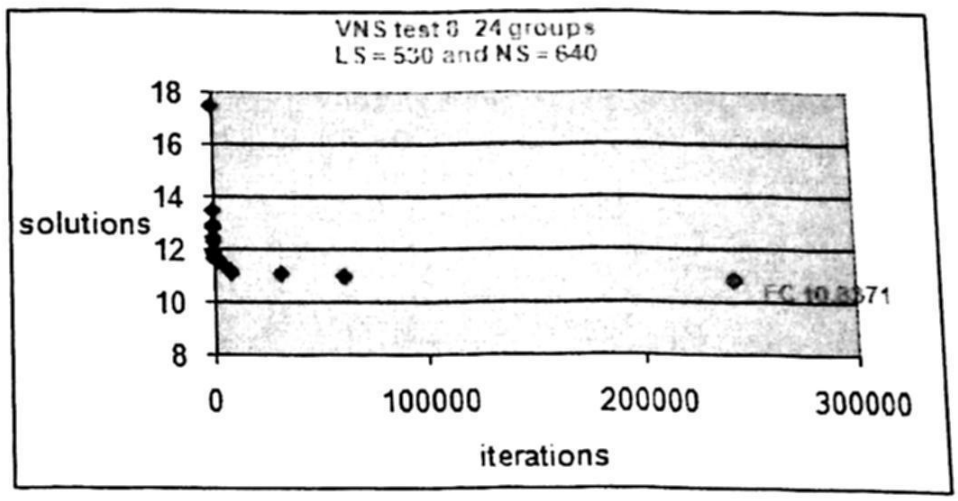

Fig. 1. Objective (Cost Function) vs. Number of Iterations for VNS Std. Order = 8, Groups =24, LS = 530, NS = 640, FC = 10.8371

This instance has been chosen as a representative example of the experiment designed, because it was found that 24 groups is an observed turning point in our multivariate statistical study.

4.2 Model for the Cost Function and Verification of the Experimental Model

Fig. 2 shows the residual plots for the experimental model. It can be concluded that the data behave normally and a second order model is well adapted.

The data obtained in the experiments were used in a second order regression model in order to get a prediction equation. The estimated regression coefficients are presented in Table 3.

Table 3 shows that the interaction effects between parameters are less important.

4.2 Predictions for the Cost Function

Figure 3 presents the response surface plots that show the effects of the variation of LS, NS and G on the cost function. It can be seen that the number of groups in a value of 24, generates a minimum in the function cost for high values in LS and NS.

An analysis of the surface plots allowed us to observe that the cost function tends to have reduced values for a greater number of groups, the NS value should be large and the LS value high. This analysis allowed us to limit the magnitude of the control parameters in the search of the minimum value in the cost function.

Contour plots were generated for regions in the surface plots with control parameters that generated cost function values near to the optimal.

It can be seen in Fig. 4 that the best cost function of 10.85, is reached for several values of BL and NS, fixing G in 24, and obtaining a contour of 10.8378, shown in the upper right of the graph. Otherwise for LS = 848, NS = 640 and G = 24 the objective function has a minimum value, as can be seen in Fig. 5. For 24 groups the

optimal value obtained with PAM is 9.279 and the cost function obtained with the VNS is 10.8378 with 24 groups. LS = 848 and NS = 640 as an example. These parameters were used for the Regression with a Second Order model (See Fig. 5).

All solutions obtained by the VNS needed less than 13 minutes.

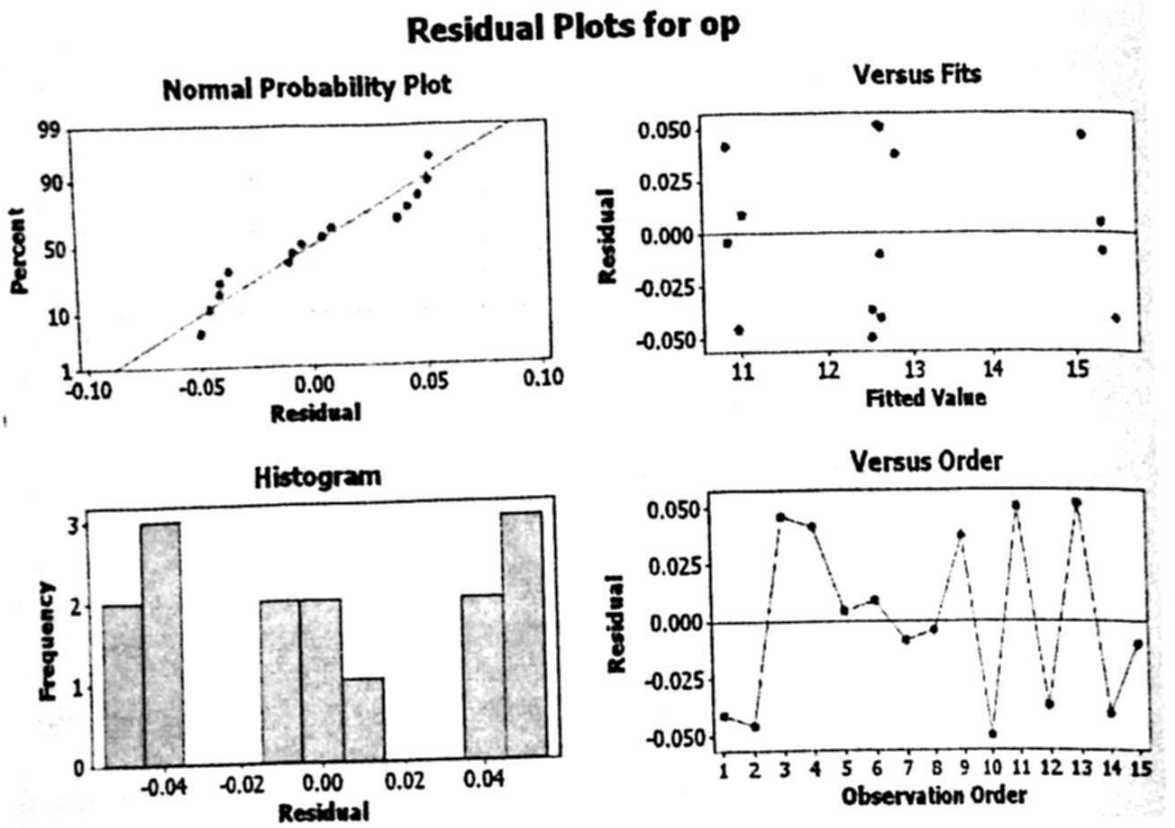

Fig. 2. Graphs of normal probability, residual and histogram of results

Table 3. Estimated Regression Coefficients

Term	Coef	SE Coef	T	P
Constant	24.1966	0.400207	60.460	0.000
G	-0.8701	0.035285	-24.658	0.000
LS	-0.0011	0.000489	-2.280	0.072
NS	-0.0002	0.000648	-0.280	0.791
G*G	0.0138	0.000912	15.175	0.000
LS*LS	-0.0000	0.000000	-0.315	0.766
NS*NS	0.0000	0.000001	0.573	0.592
G*LS	0.0000	0.000017	2.055	0.095
G*NS	-0.0000	0.000022	-1.443	0.209
LS*NS	0.0000	0.000000	1.505	0.193

S=0.06308, R-Sq=99.9%, R-Sq(adj)=99.9%

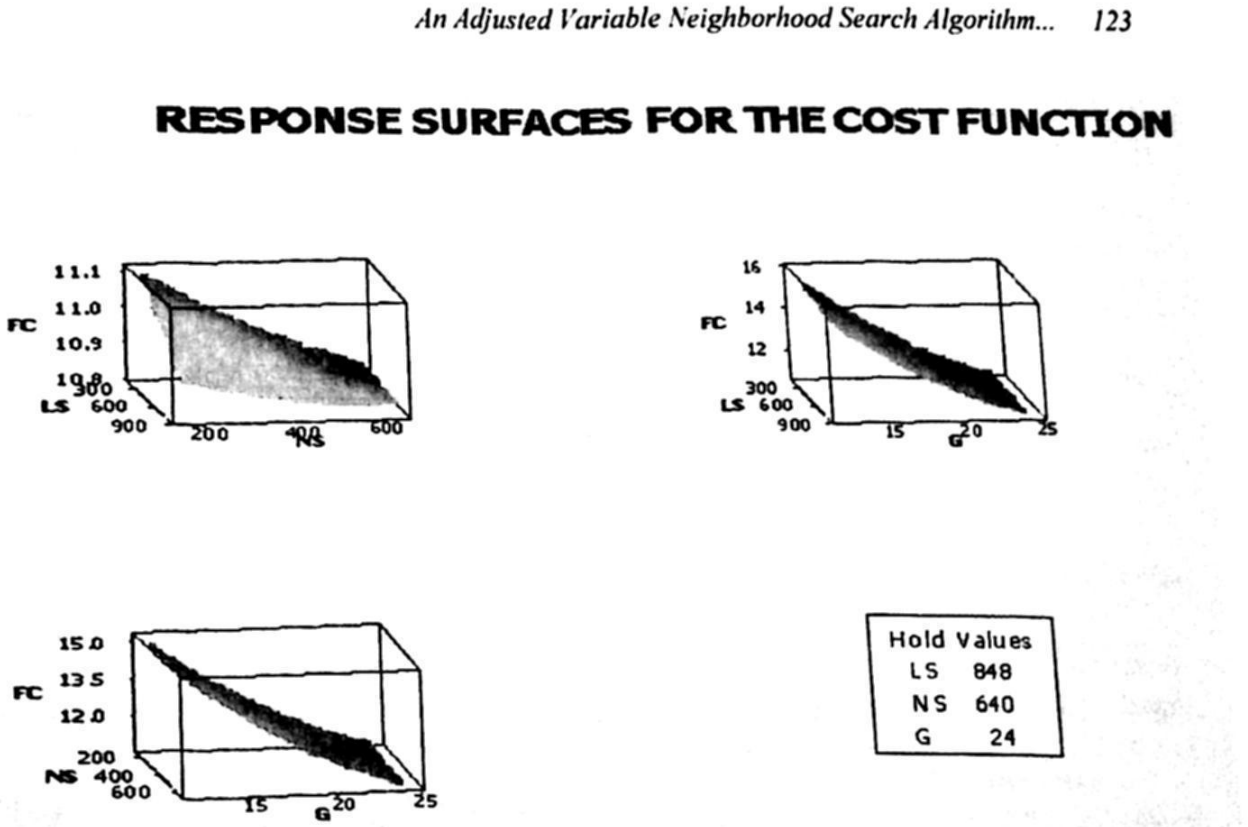

Fig. 3. Surface Plots for the Cost Function

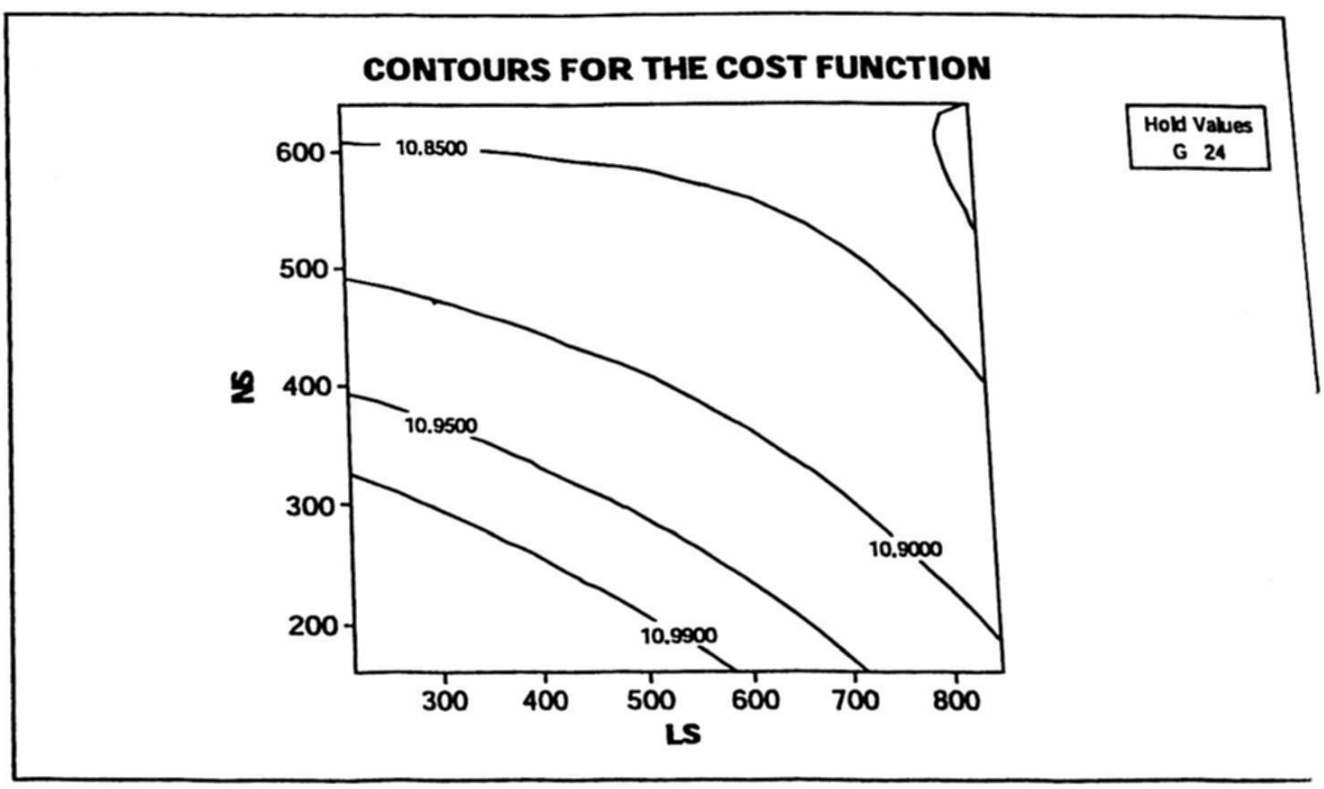

Fig. 4. Contour for 24 groups

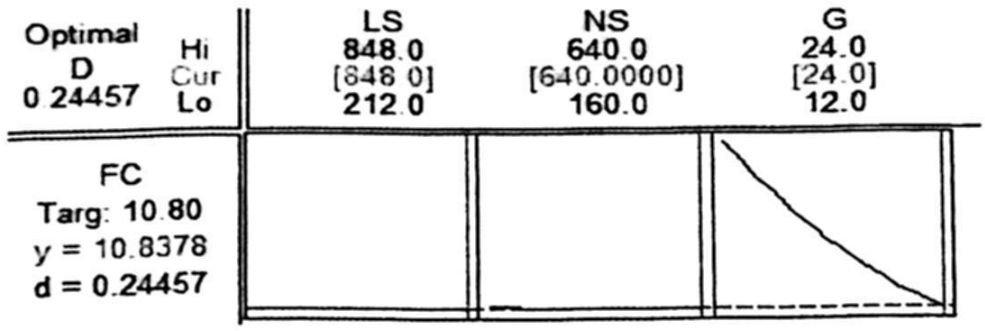

Fig. 5. Regression with the Second Order Model.

4 Conclusions

From the results obtained through all this work, we have found the VNS parameter values, used for solving the GCP that accompanied the best objective solutions.

a) In general, the number of groups is directly proportional to the quality of the solutions.
b) A value of NS close to 640 units, independently of the group size, will yield better values in the objective function.
c) The best objective values were found for LS values between 848 and 530.

The experiment used the results obtained with the empirical combinations, where 24 proved to be a good number of groups. For this reason, Tables 1 of section 4 used 24 groups. With these data development all the corresponding work tried to find a stationary point that could not be found. The future work pretends to extend the experiment, increasing the parameters value, leading to generate more instances that permit the experiment to be more extensive.

The experience obtained in the implementation of VNS for the GCP has been satisfactory in the sense that in few minutes the algorithm creates good quality solutions after 616529 iterations.

We have repeated the experiment using simulated annealing and got good solutions in a shorter time [4]. But the VNS solutions shown in this paper got better results. These results will allow us to perform a comparative work of both heuristics as a future research.

References

1. Barr, R. S., Golden, J.P., Resende, M. G., W.R. Stewart W.R.: Designing and Reporting on Computational Experiments with Heuristics Methods. Journal of Heuristics, Kluwer Academic Publisher (1995) 9–32.
2. Bernábe, L. B., López S.: Statistical Classificatory Analysis Applied to Population Zones. 8th. World Multiconference on Systemics, Cybernetics and Informatics (2004).

3. 3. Bernábe, L. B., A. Osorio, M. A., Duque J. C.: Clasificación Sobre Zonas Geográficas: Un Enfoque de Optimización Combinatoria para el Problema de Regionalización. XIII CLAIO Congreso Latino-Iberoamericano de Investigación Operativa (2006).
4. Bernábe, L. B., Ramírez R. J., Espinosa, R. J.: Evaluación de un algoritmo de recocido simulado con superficies de respuestas. Revista de Matemáticas Teoría y Aplicaciones, ISSN: 1409-2433 volume 16 number 1, (2009) 159-177
5. Duque, J. C.: Design of homogeneous territorial units. A Methodological Proposal and Applications. PhD dissertation, University of Barcelona (2004)
6. Duque, J. C., Ramos, R., Suriñach, J.: Supervised Regionalization Methods: A Survey. International Regional Science Review (2007) 195-220.
7. Fernando Bação, Victor Lobo, Marco Painho.: Applying genetic algorithms to zone design. Springer Verlag (2004)
8. Gordon, A. D.: A survey of constrained classification. Computational Statistics & Data Analysis (1996) 17–29.
9. Hansen P., Mladenovic, N.: Variable neighborhood search, Les Cahiers du GERAD (1996) 96-49.
10. Hansen P., Mladenovic, N.: Variable neighbourhood search. In Fred Glover and Gary A. Kochenberger, editors, Handbook of Metaheuristics, Kluwer (2003).
11. Kaufman, L., Rousseeuw, P.J.: 1987. Clustering by means of medoids. Statistical Data Analysis based on the L1 Norm, North-Holland, Amsterdam (1987) 405-416.
12. Montgomery D.: Design and Analisis of Experiments. Ed. Wiley 2ª Edition (1991)
13. Murtagh F.: A survey of algorithms for contiguity-constrained clustering and related problems. Computer Journal (1991) 82-88
14. Openshaw, S., Wymer C.: Classifying and regionalizing census data.In Census users handbook, ed. S. Openshaw, Cambridge, UK, GeoInformation International. (1995) 239-70
15. Pizza, E., Murilo, A., Trejos, J.: Nuevas técnicas de particionamiento en clasificación automática. Revista de Matemáticas Teoría y Aplicaciones, issn: 1409-2433 (1999) 51-66.
16. Romero D., Burguete J., Martinez E., Velasco J.: Parcelación del territorio nacional: Un enfoque de optimización combinatoria para la construcción de marcos de muestreo en hogares. INEGI (2004)
17. Rousseeuw, P.J., Hubert M., Struyf A.: Clustering in an object-oriented environment. Journal of Statistical Software (1997) 02-10
18. Zamora, A. E.: Implementación de un algoritmo compacto y homogéneo para la clasificación de zonas geográficas AGEBs bajo una interfaz gráfica. Tesis de Ingeniería en Ciencias de la Computación BUAP FCC, Asesor B. Bernábe (2006)
19. Zoltners, A., Sinha, P.: 1983, Towards a unified territory alignment: A review and model. Management Science, (1983) 1237-1256
20. http://www.inegi.gob.mx Instituto Nacional de Estadística, Geografía e Infomática (INEGI).