Multivariate Microaggregation Based Genetic Algorithms

Agusti Solanas, Antoni Martínez-Ballesté, Josep M. Mateo-Sanz and Josep Domingo-Ferrer

Abstract— **Microaggregation is a clustering problem with cardinality constraints that originated in the area of statistical disclosure control for microdata. This article presents a method for multivariate microaggregation based on genetic algorithms (GA). The adaptations required to characterize the multivariate microaggregation problem are explained and justified. Extensive experimentation has been carried out with the aim of finding the best values for the most relevant parameters of the modified GA: the population size and the crossover and mutation rates. The experimental results demonstrate that our method finds the optimal solution to the problem in almost all experiments when working with small data sets. Thus, for small data sets the proposed method performs better than known polynomial heuristics and can be combined with these for larger data sets. Moreover, a sensitivity analysis of parameter values is reported which shows the influence of the parameters and their best values.**

Keywords— **Microaggregation, Genetic algorithms, Privacy, Evolutionary computation, Statistical disclosure control.**

I. Introduction

Microaggregation is a clustering problem with minimum cardinality constraints that originated in statistical disclosure control (SDC). The aim of SDC techniques is to protect data collected by statistical agencies in such a way that data release does not result in privacy violations for respondents.

Microaggregation can be operationally defined in terms of two steps:

- During the *partition* step, the original records in the data set are partitioned into several groups in such a way that records in the same group are very similar to each other. The number of records in each group must be at least k. This parameter k can be considered a security parameter: the larger k, the more secure the microaggregation. The resulting clustering of the data set is known as k-partition.
- In the *aggregation* step, an aggregation operator, such as the mean, is used to compute a centroid for each group. Then each record is replaced by its group centroid. This replacement implies a loss of information which grows with k. Values $k = 3$, 4 or 5 are typically chosen to strike a balance between information loss and security.

Microaggregation is relevant not only in SDC, but also in artificial intelligence [1]. In the latter field, the application is to increase the *knowledge* of a system for decision making and domain representation. Microaggregation techniques may also be used in data mining in order to scale down the data set while minimizing the information loss.

A. Optimal microaggregation

In [2], optimal microaggregation is defined as the one that yields a k-partition maximizing the within-groups homogeneity: since microaggregation replaces values in a group by the group centroid, the higher the within-groups homogeneity, the lower the information loss.

The *sum of squares criterion* is commonly used for measuring the homogeneity in clustering [3]. In terms of sums of squares, maximizing the within-groups homogeneity is equivalent to finding a k-partition minimizing the within-groups sum of squares (SSE).

In [4] a polynomial algorithm was presented for optimal microaggregation in the univariate case (only one attribute being microaggregated). However, optimal multivariate microaggregation has been shown to be NP-hard [5].

B. Heuristics for multivariate microaggregation

Although a method for optimal univariate microaggregation exists, realistic data sets are multivariate (several attributes) and must be microaggregated heuristically. Microaggregation heuristics fall into two categories: *fixed-size microaggregation* [2], [6] (yielding k-partitions where all groups have size k, except perhaps one group) and *data-oriented microaggregation* (yielding k-partitions where all groups have sizes between k and $2k - 1$ [2]).

Fixed-size microaggregation heuristics are simple and computationally efficient, with computational cost around $O(n^2)$, where n is the number of records in the data set.

On the other hand, data-oriented heuristics can often achieve lower information loss because they are able to adapt the choice of group sizes to the structure of the data set. The price paid in increased computational complexity, resulting in $O(n^3)$ cost; this can be a problem for large data sets.

C. Contribution and plan of this paper

In this paper we present a new method based on a modified genetic algorithm (GA) for microaggregating a multivariate numerical data set and we study the best values for its main parameters. In Section II the proposed GA is described. Section III reports experimental results. Based on the empirical results obtained, in Section IV we follow some ideas of [7] and develop a sensitivity analysis of our GA to identify the significant parameters (*i.e.* crossover rate, mutation rate and population size) and their best values . Finally, Section V is a conclusion.

II. The proposed scheme

The proposed method is based on a classical GA scheme and can be described in terms of a population of chromosomes, a coding sequence, a selection algorithm, a fitness function and some genetic operators.

The innovation of our method lies in the design of some of the above components. The modifications introduced in our scheme with respect to a classical GA are intended to boost performance when microaggregating multivariate data. We next deal with those modifications in detail.

A. The coding

The way in which candidate solutions are encoded is a key factor in the success of a GA [8]. Although the array of possible coding alternatives can be both invigorating and bewildering,

Agusti Solanas, Antoni Martínez-Ballesté and Josep Domingo-Ferrer are with the Department of Computer Science and Mathematics, Rovira i Virgili University of Tarragona, Av. Països Catalans, 26, 43007 Tarragona, Catalonia, Spain. (e-mail: agusti.solanas@urv.net; antoni.martinez@urv.net; josep.domingo@urv.net.)

Josep M. Mateo-Sanz is with the Group of Statistics and Operations Research, Rovira i Virgili University of Tarragona, Av. Països Catalans, 26, 43007 Tarragona, Catalonia, Spain. (e-mail: josepmaria.mateo@urv.net.)

The authors are partly supported by the Spanish Ministry of Science and Education through project SEG2004-04352-C04-01 "PROPRIETAS" and by the Catalan government under grant 2005 SGR 00446.

codings can be mainly classified according to their alphabet in three main categories:

- Binary codings: They are based on a two-symbol alphabet (*i.e.* 0 and 1) and have been widely used because they were the codings initially introduced by Holland in his pioneering work [9]. Some recent examples are [10] [11][12]. Moreover, lots of heuristics based on binary codings have been developed for determining the value of parameters such as the crossover rate and the mutation rate [13] [14][7].
- N-ary codings: They differ from binary codings in that each gene can take N different alleles. It is clear that all problems that can be encoded using N-ary codings can be also encoded by using a binary alphabet which would encode the N-ary alphabet. However, some problems are more easily and naturally encoded by using an N-ary alphabet [15] [16]. As it will be shown next, this is the case of multivariate microaggregation.
- Real-valued codings: These codings assign a real value to each gene. This kind of coding is basically used when the representation of a real number in a binary alphabet is awkward, this is, when the number's precision or range are not previously and clearly known [17][18] [19].

In [2] a binary coding was used to solve the problem of univariate microaggregation. The idea of this coding was to sort univariate records in ascending order and to represent the i-th record by the i-th symbol in a binary string (chromosome). A chromosome can represent a k-partition as follows. If a cluster starts at the i-th record, its bit in the chromosome is initialized with a "1" symbol; otherwise it is set to "0". This coding makes sense for univariate data sets because they can be sorted. Unfortunately, no generalization to the multivariate case is possible because there is no way of sorting multivariate records without giving a higher priority to one of the attributes. To overcome this limitation, we propose a new coding that can be applied to data sets containing any number of dimensions (*i.e.* attributes).

Microaggregation requires that each group in a k-partition contain at least k records. Thus, the maximum number G of groups in a k-partition is

$$G = \left\lfloor \frac{n}{k} \right\rfloor \tag{1}$$

where n is the total number of records to be microaggregated. The maximum number of groups G can be viewed as the number of different alleles which must be defined in order to be able to encode a valid solution of the problem within each chromosome. Thus, the cardinality of our alphabet is G and each symbol of the alphabet represents one group of the k-partition.

When encoding the chromosome, we consider a fixed number of genes equaling the number of records in the data set. This is, the value of the i-th gene in a chromosome defines the group of the k-partition which the i-th record in the data set belongs to.

Using alphabets with more than G symbols is conceivable, but the *principle of minimum alphabets* states that the smallest alphabet that permits a natural expression of the problem should be selected [20].

B. Initializing the population

The standard approach to initializing the population of chromosomes is through a uniform pseudo-random generator mapped on the desired alphabet. Pseudo-random chromosome initialization with n records and G different alphabet symbols results in

$$V'_{(G,n)} = G^n \tag{2}$$

possible chromosomes. Unfortunately, only a small fraction of such chromosomes meet the cardinality constraints to represent valid k-partitions, let alone candidate optimal k-partitions. We show this next.

Domingo-Ferrer and Mateo-Sanz [2] proved that

Result 1: **In an optimal k-partition, each group has between k and $2k-1$ records.**

From Result 1 the minimum number of groups in a k-partitions is

$$g = \left\lceil \frac{n}{2k-1} \right\rceil \tag{3}$$

The probability of randomly obtaining a k-partition candidate to optimality (that is, meeting the cardinality constraints of Result 1) is the total number of candidate optimal k-partitions divided by the the number of possible chromosomes given in Equation (2). Computing the exact number of k-partitions candidate to optimality is not straightforward and is beyond the scope of this paper. However, we provide an upper bound that can be used to determine the maximum probability of obtaining a k-partition candidate to optimal by random initialization.

First, we have to count the number $a_{n,k}$ of *representatives* [1] of optimal k-partitions given n and k. We use the following recursive definition for $a_{n,k}$

$$a_{n,p} = 0 \text{ if } n < p \ , k \le p \le 2k-1;$$

$$a_{n,p} = 1 \text{ if } p \le n \le 2k-1 \ , k \le p \le 2k-1;$$

otherwise, if $n \ge 2k$

$$a_{n,p} = a_{n-p,p} + a_{n-(p+1),p+1} + a_{n-(p+2),p+2} + \ldots$$
$$\ldots + a_{n-(2k-1),2k-1} \ , k \le p \le 2k-1; \tag{4}$$

where $a_{n,p}$ denotes the number of representatives with n elements with the first subset having a cardinality $\ge p$.

Table I lists the values of $a_{n,k}$ for several choices of k and n.

TABLE I
VALUES OF $a_{n,k}$ FOR SEVERAL CHOICES OF k AND n

n	$k=3$	$k=4$	$k=5$
10	2	2	1
20	6	6	5
30	11	14	14
40	18	27	29
50	26	45	56
100	94	272	520
500	2134	26477	197549
1000	8434	205037	2953389

We now take the representative with the maximal number G of groups and obtain the number μ of different candidate optimal k-partitions it represents by using the following multinomial formula

$$\mu = \frac{n!}{k_1!k_2!\ldots k_G!}\frac{1}{G!} \tag{5}$$

where k_m is the cardinality of the m-th group in the k-partition. Clearly, μ is an upper bound for the number of candidate optimal k-partition represented by any representative. Finally,

[1] A representative is a k-partition with its groups enumerated in ascending order of cardinality. Thus, the k-partition with three groups of cardinalities 3, 4 and 5, respectively, represents all k-partitions formed by three groups with the same cardinalities (*i.e.* $\{3,4,5\}$, $\{3,5,4\}$, $\{4,3,5\}$, $\{4,5,3\}$, $\{5,3,4\}$ and $\{5,4,3\}$).

ALGORITHM I

Initialization of a chromosome considering the constraints imposed by microaggregation

```
01) function Init_Chromosome (inout chromosome, in k) is
02)   integer records_in_group[number_of_genes]
03)   integer gene_number
04)   integer group_number
05)   begin
06)     gene_number = 0
07)     while gene_number < chromosome_size do
08)       group_number = rand() mod G
09)       if records_in_group[group_number]< 2k − 1 then
10)         chromosome[gene_number]←group_number
11)         records_in_group[group_number] += 1
12)         gene_number += 1
13)       else
14)         Do nothing
15)       end if
16)     end while
17) end function
```

multiplying μ by the number of representatives $a_{n,k}$ and dividing by the number G^n of random chromosomes, we get an upper bound on the probability P(random candidate optimal) of hitting a candidate optimal k-partition by random initialization:

$$P(\text{random candidate optimal}) \leq \frac{a_{n,k}\mu}{G^n} \tag{6}$$

Example 1: Let us consider a data set with $n = 30$ records and $k = 3$. By applying Expression (4) we obtain $a_{30,3} = 11$. Then by applying Expression (5) we have $\mu = \frac{30!}{3!^{10}}\frac{1}{10!} = 1208883745669600000$. Finally Inequality (6) yields

$$P(\text{random candidate optimal}) \leq \frac{\mu \times 11}{G^n} =$$

$$= \frac{13297721202365600000}{10^{30}} = 13,2977212023656 \times 10^{-12} \approx 0$$

□

From the previous example, it becomes clear that random initialization is not suitable to obtain canditate optimal k-partitions. The cardinality constraints stated in Result 1 must be embedded in the initialization procedure. Thus, we propose Algorithm I to initialize the chromosomes with solutions that are candidate to optimality. After applying Algorithm I we have a partition built up by groups of cardinality at most $2k-1$. However, there can be groups with less than k records. Thus, after applying Algorithm I we randomly move records from groups with cardinality more than k to groups under cardinality k. Finally we obtain a k-partition which is candidate to be optimal.

C. The fitness function

The evaluation of a chromosome in the population is the costliest part in the whole evolution process of our GA. We want to obtain a measure of the homogeneity of groups in the k-partition. To do so, there are a lot of known measures based on different distance definitions (*e.g.* Euclidean distance, Minkowski distance, Chebyshev distance, etc.).

The most frequently used homegeneity measure for a clustering is the within-group sum of squares SSE [2][21]. The SSE is the sum of square distances from the centroid of each group to every element (*i.e.* record) in the group. For a k-partition, SSE can be computed as:

$$SSE = \sum_{i=1}^{s}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)'(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i) \tag{7}$$

where s is the number of groups in the k-partition and n_i is the number of records in the i-th group. Expression (7) is applied to the data after standardizing them, this is, after subtracting from the values of each attribute the attribute mean and dividing the result by the attribute standard deviation [2].

During the evolution process, some chromosomes can appear which are not candidate optimal k-partitions due to the effect of genetic operators. This is, some groups can appear whose cardinality is no longer in the range $[k, 2k-1]$. In these cases, we do not remove the chromosome from the population; instead, we penalize it in order to nearly preclude its reproduction. By penalizing these undesirable chromosomes, we reduce the search space and avoid subsequent computation of their SSE. Therefore, the time needed for evolving a whole population decreases.

Our goal is to minimize SSE. Thus, the fitness value that we finally assign to a chromosome is

$$Fitness = \frac{1}{SSE + 1} \tag{8}$$

D. Selection Scheme and Genetic operators

Our GA uses the well-known roulette-wheel selection algorithm. This selection algorithm is based on weighting the chromosomes of the population according to their fitness. Thus, the probability of a chromosome being selected for passing to the next generation is proportional to its fitness. This method tends to converge quickly because it does not preserve the diversity of the population. Even though this entails some risk of missing good solutions, experimental results below show that this risk is small and can be afforded for the sake of speed.

We have used the most common genetic operators, this is, crossover and mutation. In many occasions these operators are modified in order to improve the genetic search for a good solution. However, in this work we want to study the influence of these operators in their most genuine form to determine the best values for their rates. Thus, we apply simple one-point crossover and mutation to each individual gene.

III. Experimental results

In this section we present the experimental results of the genetic approach to microaggregation. We report on a battery of tests with several random data sets and we compare the results with those obtained using MDAV.

MDAV falls into the category of fixed-size microaggregation heuristics. At the time of this writing and to the best of our knowledge, it is the most suitable heuristic for large data sets [22], [23]. On the other hand, ES always gives the optimal solution, but exhaustive search is infeasible on data sets as small as $n = 20$ records.

A test consisted of several runs of the GA with the following parameters:

- Mutation rate (M_r) from 0 to 1, with step 0.1.
- Crossover rate (C_r) from 0 to 1, with step 0.1.
- Population size, *i.e.* the number of chromosomes, from 10 to 100, with step 10.

The above implies 1210 different parameter settings. For each setting, the GA was run 10 times, so that altogether a test comprised 12100 runs of the GA. Tests were performed on Pentium 4

[2] Standardization does not affect attribute correlations.

running at 3 GHz with Debian Linux operating system. Programs were coded in C++.

Random data sets $n \times d$ were used were constructed for several numbers n of records and several numbers of attributes d. Attribute values were drawn from the $[-10000, 10000]$ interval by simple random sampling. For data sets with $n = 11$, $n = 20$ and $n = 35$ the number of epochs (number of iterations) was 10^4. For the data set with $n = 50$, the number of epochs was 10^5. Note that for microdata sets with $n = 11$ records, the optimal k-partition could be found by ES in a reasonable time. For $n > 11$ the SSE obtained with GA was only compared to the SSE obtained using MDAV.

A. Results for the random data sets

Using exhaustive search ES to find the optimal k-partition results in $SSE = 6.83$ for the 11x2 data set and $SSE = 15.20$ for the 11x3 data set.

As mentioned above, 10 runs were performed for each setting of M_r, C_r and the population size. This means 100 runs for each pair (M_r, C_r). All averages mentioned below are over those 100 runs. Figure 1 shows the average SSE for each pair (M_r, C_r). It can be observed that, for $M_r > 0$ and $C_r > 0$, the k-partition with the optimal SSE is always obtained.

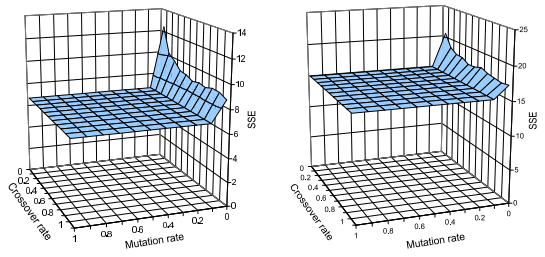

Fig. 1. Average SSE obtained for each pair (M_r, C_r), for data sets 11x2 and 11x3

Although an optimal SSE is generally obtained with the GA, the convergence epoch varies according to M_r and C_r. Figures 2 and 3 show the average value of the convergence epoch for the 100 tests per (M_r, C_r) pair. It can be observed that this average grows with M_r.

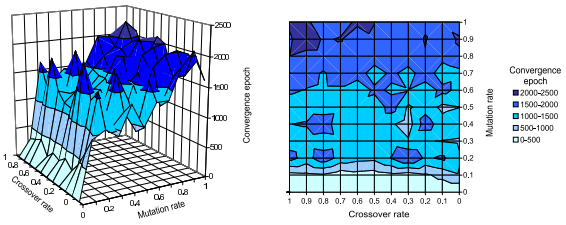

Fig. 2. Average convergence epoch for each (M_r, C_r) pair, for data set 11x2

Now we consider data sets with $d = 2$ and $n = 20$, $n = 35$, $n = 50$ and $n = 100$. Figure 4 shows the average SSE obtained for each (M_r, C_r) pair. It can be observed that, for data set 20x2 (top left), the best SSE values are always obtained for $M_r \in [0.1, 0.4]$. In data sets 35x2 (top right) and 50x2 (bottom left) the behaviour is quite similar: only values $M_r \approx 0.1$ output the lowest average SSE values, *i.e.* the best k-partitions. For $M_r \in [0.2, 1]$ larger average SSE values are obtained. data set 100x2 (bottom right) also features low average SSE values for $M_r = 0.1$.

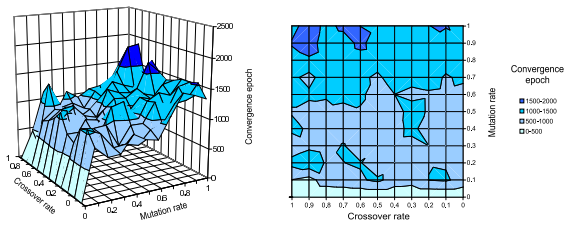

Fig. 3. Average convergence epoch for each (M_r, C_r) pair, for data set 11x3

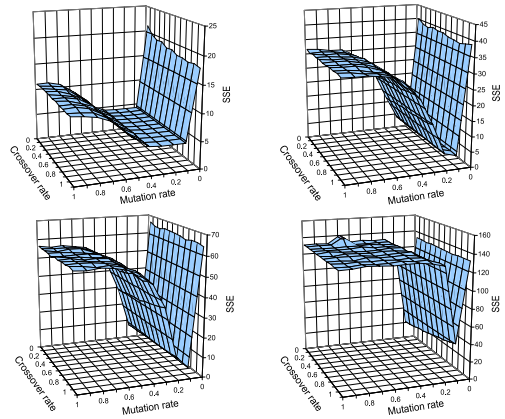

Fig. 4. Average SSE for each (M_r, C_r) pair, for data sets 20x2 (top left), 35x2 (top right), 50x2 (bottom left) and 100x2 (bottom right).

Table II shows: i) the minimum SSE obtained together with the settings of C_r, M_r and population size that yielded it; ii) the average SSE per (M_r, C_r) pair.

It is apparent that both minimum and average SSE with GA are always lower than those obtained with MDAV, except for data set 100x2. The convergence epoch plays a key role in the results for larger data sets: the best k-partitions are obtained using more iterations and low values of M_r. Table III shows the average convergence epoch of the tests yielding the best k-partition.

TABLE II

MINIMUM AND AVERAGE SSE VALUES FOR TESTS WITH LARGER DATA SETS

Data set	MDAV	GA_m	M_r	C_r	pop.size
20x2	5.39	4.29	0.1	0.4	60
35x2	5.01	3.13	0.1	0.1	90
50x2	5.92	4.02	0.1	0.1	50
100x2	5.48	15.37	0.1	0.2	50

Data set	MDAV	GA_A	M_r	C_r
20x2	5.39	4.61	0.1	0.7
35x2	5.01	3.59	0.1	0.5
50x2	5.92	4.67	0.1	0.5
100x2	5.48	23.78	0.1	0.0

TABLE III

AVERAGE CONVERGENCE EPOCH OF TESTS WITH BEST RESULTS

n	Av.Conv.E.
11	1384
20	3251
35	5269
50	53405

B. Results vs data set dimensionality

In both the 11x2 and 11x3 data sets the optimal SSE was obtained. In order to check whether the optimal SSE is obtained for larger values of d, tests of GA have been performed with random data sets with $d = 5$, $d = 7$ and $d = 10$. In all cases, the k-partition generated for $M_r \neq 0$ and $C_r \neq 0$ equals the optimal k-partition output by ES. Table IV shows the results of these tests. In all three data sets, around a 90% of tests output the optimal SSE value.

TABLE IV

SSE FOR DATA SETS WITH $n = 11$ AND DIFFERENT d VALUES

Data set	SE	MDAV	GA	Av.Conv.E.
11x5	29.36	39.11	29.36	931
11x7	43.28	52.47	43.28	839
11x10	68.66	71.14	68.66	705

IV. Sensitivity analysis

Based on the data obtained from the empirical work described in the previous section, we conducted a statistically sound sensitivity analysis. In particular, this analysis confirmed the inferences on the best parameter values that seem apparent from the graphics presented in the previous section. We next describe the procedure followed.

The Analysis Of Variance (ANOVA), the Tukey and the Scheffé tests [24] [25] [26] were used. The sensitivity analysis can be divided in the four steps below:

1. ANOVA test: The ANOVA test was applied on each data set taking SSE as the response variable and the following as independent factors:

(a) Mutation rate: For each data set, 11 values for the mutation rate ranging from 0 to 1 with step 0.1 were tested.

(b) Crossover rate: Similarly to the mutation rate analysis, 11 values for the crossover rate ranging from 0 to 1 with step 0.1 were analyzed.

(c) Population size: This parameter is different from the previous ones in that it cannot be initialized to 0 because a population with 0 chromosomes makes no sense. Thus, for each data set, 10 different values of the population size ranging from 10 to 100 with step 10 were studied.

ANOVA made it possible to decide which factors have real influence on SSE. In this case, we considered that a significance level under 0.05 meant that a factor was significant.

2. Analyze the significant parameters: Once the significant parameters were detected the Tukey test and the Scheffé test were applied in order to determine the most promising values for each of the significant parameters. The Tukey and Scheffé tests group the values when they are not significantly different. Thus, the result of their application was a set of groups of values. The best group was the one considered in the next step of analysis.

3. Analyze the best group: We applied the ANOVA test on the best group of parameters obtained in the previous steps but taking the convergence epoch (number of iterations) as the response variable. As a result, the parameters with an influence on the convergence epoch were identified.

4. Determine the best values: In this last step, we wished to find out which among the best values according to the SSE minimization criterion yielded the lowest convergence epoch. In order to find these values, the Tukey and Scheffé tests were applied on the significant parameters determined in the previous step.

Although this method may look a bit complicated, it is very easy to implement because the above statistical tests are available in most statistical packages.

We summarize in Table V the best parameter values identified by the above sensitivity analysis for the random data sets used in the previous sections.

TABLE V

BEST VALUES FOR THE MUTATION RATE, CROSSOVER RATE AND POPULATION SIZE OBTAINED THROUGH THE SENSITIVITY ANALYSIS

Data set	Mut. rate	Cross. rate	Pop. size
11x2	0.1 - 1	0.3 - 1	80 - 100
20x2	0.1 - 0.3	0.1 - 1	80 - 100
35x2	0.1	0 - 0.3	90 - 100
50x2	0.1	0 - 1	60 - 100
100x2	0.1	0 - 0.8	70 - 100
11x3	0.1 - 1	0.5 - 1	60 - 100

Finally, we draw the following conclusions from the sensitivity analysis.

Result 2: **Mutation is very important to obtain a low SSE in a reasonable number of epochs . A value close to 0.1 seems a good choice.**

Result 3: **Crossover is important to obtain a low SSE but it has little influence on the length of the convergence epoch. A value of 0.3 should work well.**

Result 4: **The population size has an influence on SSE and the convergence epoch. A large population, say 100 chromosomes, seems reasonable.**

V. Conclusions and future work

This paper has presented a new method for multivariate microaggregation which is based on genetic algorithms. New coding and initialization schemes have been presented and the influence of the main parameters of the GA (*i.e.* mutation rate, crossover rate and population size) has been experimentally and statistically analyzed. The sensitivity analysis is a contribution in its own right and demonstrates that our method yields optimal k-partitions when working with data sets of up to 50 records.

Future research directions include:

- *Hybrid version for larger data sets.* The proposed GA works better than MDAV and is practically optimal for data sets of up to $n = 50$ records. Its performance degrades for larger data sets. These could be dealt with by using MDAV with $k = 50$ to split the data set into groups of 50 records; then each group could be microaggregated using GA with $n = 50$ and a smaller k.
- *Modification of the genetic operators.* In this article, we have used the canonical mutation and crossover operators. These operators work well when the number of records in the data set is not too big. Unfortunately, their efficiency decreases dramatically for larger data sets. Devising efficient adaptations of these operators for bigger search spaces is a future research challenge.

Acknowledgments

The authors would like to acknowledge the suggestions of Dr. Blas Herrera.

References

[1] J. Domingo-Ferrer and V. Torra, "On the connections between statistical disclosure control for microdata and some artificial intelligence tools," *Information Sciences*, vol. 151, pp. 153–170, May 2003.

[2] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 189–201, 2002.

[3] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.

[4] S. L. Hansen and S. Mukherjee, "A polynomial algorithm for optimal univariate microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 1043–1044, July-August 2003.

[5] A. Oganian and J. Domingo-Ferrer, "On the complexity of optimal microaggregation for statistical disclosure control," *Statistical Journal of the United Nations Economic Comission for Europe*, vol. 18, no. 4, pp. 345–354, 2001.

[6] A. Hundepool, A. Van de Wetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P.-P. DeWolf, J. Domingo-Ferrer, V. Torra, R. Brand, and S. Giessing, *μ-ARGUS version 4.0 Software and User's Manual*, Statistics Netherlands, Voorburg NL, may 2005, http://neon.vb.cbs.nl/casc.

[7] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, and R. Gupta, "Statistical exploratory analysis of genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 405–421, 2004.

[8] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1999.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975, Second edition: MIT Press, 1992.

[10] S. J. Ovaska, T. Bose, and O. Vainio, "Genetic algorithm-assisted design of adaptive predictive filters for 50/60 hz power systems instrumentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 2041–2048, 2005.

[11] W. C. Liu, "Design of a cpw-fed notched planar monopole antenna for multiband operations using a genetic algorithm," *IEE Proceedings on Microwaves, Antennas and Propagation*, vol. 152, no. 4, pp. 273 – 277, 2005.

[12] A. Solanas, E. Romero, S. Gómez, J. M. Sopena, R. Alquezar, and J. Domingo-Ferrer, "Feature selection and oulier detection with genetic algorithms and neural networks," in *Artificial Intelligence Research and Developement*, P. Radeva B. López, J. Meléndez and J. Vitrià, Eds., Amsterdam, NL, 2005, pp. 41–48, IOS Press.

[13] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[14] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of the Third International Conference on Genetic algorithms*, San Francisco, CA, USA, 1989, pp. 51–60, Morgan Kaufmann Publishers Inc.

[15] A. S. Wu and I. I. Garibay, "The proportional genetic algorithm: Gene expression in a genetic algorithm.," *Genetic Programming and Evolvable Machines*, vol. 3, no. 2, pp. 157–192, 2002.

[16] K. Knodler, J. Poland, A. Mitterer, and A. Zell, "Optimizing data measurements at test beds using multi-step genetic algorithms," in *Advances In Fuzzy Systems and Evolutionary Computation (Proceedings of WSES EC 2001)*, N. Mastorakis, Ed., 2001, pp. 277 – 282.

[17] G. M. Megson and I. M. Bland, "Synthesis of a systolic array genetic algorithm.," in *IPPS/SPDP*, 1998, pp. 316–320.

[18] Y. K. Yu, K. H. Wong, and M. M. Y. Chang, "Pose estimation for augmented reality applications using genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, no. 6, pp. 1295 – 1301, 2005.

[19] W. Chien and C.-C. Chiu, "Using nu-ssga to reduce the searching time in inverse problem of a buried metallic object," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 10, pp. 3128 – 3134, 2005.

[20] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing, 1989.

[21] Y. Jung, H. Park, D.-Z. Du, and B. L. Drake, "A decision criterion for the optimal number of clusters in hierarchical clustering," *Journal of Global Optimization*, vol. 25, pp. 91–111, Jan 2005.

[22] J. Domingo-Ferrer, A. Martínez-Ballesté, and J. M. Mateo-Sanz, "Efficient multivariate data-oriented microaggregation," *Manuscript*, 2005.

[23] M. Laszlo and S. Mukherjee, "Minimum spanning tree partitioning algorithm for microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 902–911, 2005.

[24] Douglas C. Montgomery, *Design and Analysis of Experiments (5th Ed.)*, Wiley, 2000.

[25] C. Pérez, *Técnicas estadísticas con SPSS*, Pearson Educación, 2001.

[26] Willian Mendenhall and Terry Sincich, *Statistics for Engineering and the Sciences (4th Ed.)*, Prentice Hall, 1995.

[27] J. Domingo-Ferrer, F. Sebé, and A. Solanas, "A polynomial-time approximation to optimal multivariate microaggregation," *manuscript*, 2005.

[28] A. W. F. Edwards and L. L. Cavalli-Sforza, "A method for cluster analysis," *Biometrics*, vol. 21, pp. 362–375, 1965.

[29] A. D. Gordon and J. T. Henderson, "An algorithm for euclidean sum of squares classification," *Biometrics*, vol. 33, pp. 355–362, 1977.

[30] P. Hansen, B. Jaumard, and N. Mladenovic, "Minimum sum of squares clustering in a low dimensional space," *Journal of Classification*, vol. 15, pp. 37–55, 1998.

[31] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.

[32] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous k-anonymity through microaggregation," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 195–212, 2005.