

HYBRID GENETIC ALGORITHM APPLIED TO THE CLUSTERING PROBLEM

Danuza Prado de Faria Alckmin¹ and Flávio Miguel Varejão²

Federal University of Espirito Santo (UFES), Technological Center – Computer Science Department
Brasil

ABSTRACT

Clustering is a task, whose main objective is dividing a data set into partitions, so that patterns belonging to the same partition are similar to one another and dissimilar to patterns belonging to other partitions. It falls into the category of optimization tasks, since clustering ultimately aims at finding the best combination of partitions among all possible combinations. Metaheuristics, which are general heuristics capable of escaping local optima, can be applied to solve the clustering problem. This paper proposes a Hybrid Genetic Clustering Algorithm (HGCA) — whose initial population is generated partly by clustering algorithms — that combines a local search heuristic to the global search procedure. Such improvements are intended to provide solutions for search problems closer to the global optimum. Experiments are performed in real data sets in order to verify if the proposed approach presents any improvement in comparison with other algorithms evaluated in this work: agglomerative hierarchical; three versions of K-means, differing only in terms of initialization methods (Random, K-means ++ and PCA_Part); Tabu Search and Genetic Clustering Algorithm.

KEYWORDS. Metaheuristics. Clustering. Optimization.

MSC: 68T10

RESUMEN

El agrupamiento de datos es una tarea, cuyo principal objetivo es dividir un conjunto de datos en particiones, por lo que los patrones que pertenecen a la misma partición son similares entre sí y diferentes a los patrones que pertenecen a otras particiones. El agrupamiento de datos debe incluirse en la categoría de tareas de optimización, ya que en última instancia, agrupación aspira a encontrar la mejor combinación de particiones entre todas las combinaciones posibles.

Metaheurísticas, que son heurísticas generales capaz de escapar de óptimos locales, se puede aplicar para resolver el problema de agrupamiento. Este trabajo propone un Algoritmo de Agrupamiento Genético Híbrido (HGCA) — cuya primera población se genera en parte por algoritmos de agrupamiento — que combina una heurística de búsqueda local para el procedimiento de búsqueda global. Estas mejoras están destinadas a proporcionar soluciones a los problemas de búsqueda más cerca del óptimo global. Los experimentos se realizan en conjuntos de datos reales que el fin de comprobar si el enfoque propuesto presenta una mejora en comparación con otros algoritmos evaluados en este trabajo:

aglomeración jerárquica, tres versiones del K-means, difiriendo sólo en cuanto a los métodos de inicialización (al azar, K-means ++ y PCA_Part); Búsqueda Tabú y Algoritmo de Agrupamiento Genético.

1. INTRODUCTION

Given a data set $x_j \in \mathbb{R}^d$, $j = 1, \dots, N$, clustering algorithms aim to organize data into K partitions $\{C_1, \dots, C_K\}$ in order to optimize some cost function. This approach requires the definition of a function that associates a cost to each partition. The goal is to find the set of partitions that optimizes the sum of costs for all clusters [36].

Hybrid approaches exploring the combination of metaheuristics represent a promising technique to solve the clustering problem [32].

¹ danuza.faria@gmail.com

² fvarejao@inf.ufes.br

A hybrid approach using initialization methods combined to Simulated Annealing to tackle the clustering problem is presented in [28]. In this work, the results generated by Simulated Annealing initialized with PCA_Part (Principal Component Analysis) generated results superior to those obtained by K-means randomly initialized.

Babu [3] applied genetic algorithms to select the initial solution for K-means, an approach that overcomes the results of a direct application of genetic algorithms.

In this paper we propose a Hybrid Genetic Clustering Algorithm (HGCA) — whose initial population is generated by clustering techniques — that associates a local search heuristic to a global search process. In Section 2, we present the heuristics used in this work. We, then, describe the Hybrid Genetic Clustering Algorithm (HGCA) in Section 3. Later, in Section 4 we report our experimental results. Finally, in Section 5 we draw conclusions and suggest themes for future works.

2. APPROACHES TO SOLVE THE CLUSTERING PROBLEM

The clustering problem can be defined as: given a data set X with N patterns $X = \{x_1, \dots, x_N\}$, where each pattern $x_i = [x_{i1}, \dots, x_{id}]^T$ has d dimensions, the goal is to find K partitions $\{C_1, \dots, C_K\}$, so that patterns belonging to the same partition are more similar to each other than to those patterns belonging to other partitions.

The goal of clustering algorithms is to optimize an evaluation criterion (cost function) for K partitions. The most commonly used criterion is the Sum of the Squared Euclidean distance (SSE), which uses the Euclidean distance as dissimilarity measure [9]. SSE was the evaluation criterion selected for this study. The algorithms used in this work are presented as follows.

2.1. Agglomerative Hierarchical Algorithm

The agglomerative hierarchical algorithm implemented in this work uses the centroid linkage method based on [36] to determine the dissimilarity between clusters. According to it, the distance between clusters is defined as the distance between the representative point of each cluster, known as the centroid. In this work, the clustering problem goal is to find k partitions, where k is previously known. Therefore, the stopping criterion of the algorithm is to generate k partitions.

2.2. K-means

According to [11], the standard K-means algorithm generates initially a random set of K patterns from the data set, known as the centroids. They represent the problem's initial solution.

However, K-means is especially sensitive to the choice of initial centroids. The algorithm could converge to a local minimum if the initial solution is not chosen properly [19].

2.3. K-means with initialization by K-means++

Arthur [2] suggested a way of initializing K-means by choosing random starting centers with very specific probabilities. In this approach, a point p was chosen as a center with probability proportional to contribution to the minimization of SSE (sum-squared-error) criterion.

This method initially selects an arbitrary point in the data set to represent the first cluster center. Then, the remaining $K-1$ centroids are chosen iteratively, by selecting points with probability proportional to its contribution to the SSE criterion. Thus, the higher the contribution to minimize SSE, the higher the probability of point being considered a cluster center.

2.4. K-means with initialization by PCA-Part

Another attempt to overcome the initial sensitivity problem of K-means is to use deterministic methods that eliminate the dependence upon random factors. Su [35] developed an initialization method called PCA-Part, using a deterministic divisive hierarchical approach based on PCA (principal component analysis).

This method initially generates a single partition consisting of all data. After initialization, PCA-Part selects the partition C_j with the largest SSE at each iteration and splits it into two partitions C_{j1} and C_{j2} , whose centroids are respectively μ_{j1} and μ_{j2} . The partition C_j is divided by projecting each pattern $x_i \in C_j$ towards the first principal direction (the eigenvector corresponding to the largest eigenvalue of the covariance matrix), generating the vectors y_i . The same happens to its centroid μ_j , generating the vector α_j . The process is repeated until K partitions are generated. PCA-Part's goal is to minimize the value of SSE on each iteration.

Using this method combined with Simulated Annealing, [28] reported experimental results far more encouraging than those achieved using random initialization methods for K-means.

2.5. Tabu Search

Elaborated simultaneously by [13] and [16], Tabu Search is a local search technique that aims overcoming the problem of local solutions by using memory structures.

The method designed to find good approximations for any optimization problem has basically three fundamental principles: (i) the usage of a data structure (list) to store the search history; (ii) the usage of a control mechanism to balance the acceptance or rejection of a new configuration, which is based on restrictions and desired aspirations recorded on the tabu list; (iii) the incorporation of procedures that alternate strategies of diversification and intensification [14].

The Tabu Search method used in this paper is based on [14]. We used the classical concept of tabu list as a queue of fixed size. In other words, when a new solution is added, the oldest one leaves the list. In addition, the diversification strategy adopted uses the adaptive memory technique suggested by [4]. According to it, the best solution of a previous iteration is passed on to the next iteration as its initial solution. The neighborhood function adopted randomly selects a pattern to be moved from the currently partition to a different partition. The best solution found since the start of the execution and a tabu list are stored in memory. After reaching the stopping criterion, which is the maximum number of iterations, the solution given by the algorithm is the best solution found since the start of execution.

2.6. Genetic Clustering Algorithm

The Genetic Clustering Algorithm (GCA) is the Genetic Algorithm (GA) applied to the clustering problem in k partitions, when the value of k is previously known [5]. Tackling clustering tasks with GA requires adaptations in areas such as: the representation of the solution, fitness function, operators and the value of parameters. The changes we used in this paper are presented in the following paragraphs.

The representation of the solution used in this study is the Group-Number suggested by [5]. According to it, the solution is represented by a vector of size N , where each position i represents a pattern of the data set with value between $[1, K]$, where K is the number of partitions, indicating to which partition the pattern in position i belong.

Regarding the fitness function, this work uses the SSE criterion. Furthermore, the higher fitness scores were assigned to solutions with the smallest values of SSE.

The selection method used in this work is the roulette-wheel selection, as proposed by [15]. Selection is the process of choosing the fitter chromosomes to suffer the action of genetic operators. It was also used a strategy known as elitism, which consists in keeping in the currently generation the best chromosome of the previously generation, as suggested by [6] and [24].

Still based on [24], this work uses the single-point crossover operator. It was used a crossover rate of 80%, which is the arithmetic average between the lower limit (65%) and higher (95%) in the range of crossover rates suggested in literature.

The mutation operator used was the uniform mutation based on [24]. This genetic operator arbitrarily selects a pattern of the data set to be moved randomly to a different partition of the chain. The mutation rate used was 1%, which is the arithmetic average between the lower limit (0.01%) and higher (2%) in the range of mutation rates suggested in literature.

The stopping criterions used were a maximum number of generations or a fixed number of generations reached without improvement in the SSE value of the fitter chromosome. The details of the Genetic Clustering Algorithm used in this work are formally expressed as follows:

Algorithm 6: Genetic Clustering Algorithm procedure

Input: X , set of N patterns with d dimensions;
 K , number of centroids;
 T_{pop} , population size;
 P_{cros} , crossover rate;
 P_{mut} , mutation rate;
 N_{ger} , maximum number of generations;
Output: Best solution found s^*

```

1:  $t \leftarrow 0$ 
2: Set the initial population  $P(t)$  with size  $T_{pop}$ 
3: Evaluate  $P(t)$ 
4:  $s \leftarrow \text{select\_best\_individual}(P(t))$ 
5:  $s^* \leftarrow s$ 
6:  $S_{sse} \leftarrow SSE$  computed with the solution  $s^*$ 
7:  $without\_improvement \leftarrow 0$ 
8:  $P_{pai} \leftarrow \emptyset$ 
9: While ( $t < N_{ger}$  and  $without\_improvement < 0.05 * N_{ger}$ ) do
10:  $t \leftarrow t + 1$ 
11: Select  $P_{pai}(t)$  from  $P(t-1)$  with crossover rate  $P_{cros}$ 
12:  $P(t) \leftarrow \text{crossover } P_{pai}(t)$ 
13: Apply mutation  $P(t)$  with mutation rate  $P_{mut}$ 
14: Evaluate  $P(t)$ 
15: Apply elitism  $P(t) \leftarrow \text{select\_best\_individual } P(t-1)$ 
16:  $s' \leftarrow \text{select\_best\_individual } (P(t))$ 
17:  $S'_{sse} \leftarrow SSE$  computed with the solution  $s'$ 
18:  $\Delta_{sse} = S'_{sse} - S_{sse}$ 
19: If  $\Delta_{sse} \geq 0$  then
20:  $without\_improvement \leftarrow without\_improvement + 1$ 
21: Else
22:  $without\_improvement \leftarrow 0$ 
23:  $s^* \leftarrow s'$ 
24: End if
25: End While
26: Return  $s^*$ 

```

The algorithm starts from an initial population randomly generated.

3. HYBRID GENETIC ALGORITHM APPLIED TO THE PROBLEM OF GROUPING

Davis [6] suggests as much as possible the incorporation of the domain knowledge in Genetic Algorithms (GA). As well as the combination of GA with other optimization methods that could help solving the problems at hand. This combination turns a Genetic Algorithm into a Hybrid Genetic Algorithm.

It is believed that the appropriated combination of genetic algorithms and specific heuristics tends to be significantly superior to the canonical versions of genetic algorithms [25].

This paper presents a Hybrid Genetic Algorithm (HGA) approach to the clustering problem using the heuristic K-means. It includes a mechanism for improving the generation of initial population, as well as the exploration of promising regions of the search space.

The proposed algorithm has the basic features of a Genetic Clustering Algorithm (GCA), but incorporates additional mechanisms to obtain better results than those achieved using classic GCA — described in the

previous section — and also better than those achieved using the clustering heuristics described and evaluated in this study.

The Hybrid Genetic Clustering Algorithm (HGCA) proposed has the following characteristics in comparison to the classic GCA:

- The initial population is not only randomly-generated;
 - Part of the population is generated through three different versions of the heuristic K-means. The goal is generating — in a low computational time — distinct individuals with better fitness levels than those they would present if they were randomly generated;
 - At each generation, the heuristic K-means is applied to the currently population, performing an efficient local search in the solutions found in order to generate even better solutions.
- The details of the Hybrid Genetic Clustering Algorithm (HGCA) proposed are formally expressed as follows:

Algorithm 7: Hybrid Genetic Clustering Algorithm (HGCA) procedure

Input: X , set of N patterns with d dimensions;
 K , number of centroids;

T_{pop} , population size;

P_{cros} , crossover rate;

P_{mut} , mutation rate;

N_{ger} , maximum number of generations;

Output: Best solution found s^*

```

1:   $t \leftarrow 0$ 
2:  Set the initial population  $P(t)$  with size  $T_{pop}$ 
3:  Evaluate  $P(t)$ 
4:   $s \leftarrow \text{select\_best\_individual}(P(t))$ 
5:   $s^* \leftarrow s$ 

6:   $S_{sse} \leftarrow SSE$  computed with the solution  $s^*$ 
7:   $without\_improvement \leftarrow 0$ 
8:   $P_{pai} \leftarrow \emptyset$ 
9:  While ( $t < N_{ger}$  and  $without\_improvement < 0.05 * N_{ger}$ ) do
10:    $t \leftarrow t + 1$ 
11:   Select  $P_{pai}(t)$  from  $P(t-1)$  with crossover rate  $P_{cros}$ 
12:    $P(t) \leftarrow \text{crossover } P_{pai}(t)$ 
13:   Apply mutation  $P(t)$  with mutation rate  $P_{mut}$ 
14:   For all  $I_i \in P(t)$ 
15:     Apply LocalSearch( $I_i$ )
16:   End For
17:   Evaluate  $P(t)$ 
18:   Apply elitism  $P(t) \leftarrow \text{select\_best\_individual } P(t-1)$ 
19:    $s' \leftarrow \text{select\_best\_individual } (P(t))$ 
20:    $S'_{sse} \leftarrow SSE$  computed with the solution  $s'$ 
21:    $\Delta_{sse} = S'_{sse} - S_{sse}$ 
22:   If  $\Delta_{sse} \geq 0$  then
23:      $without\_improvement \leftarrow without\_improvement + 1$ 
24:   Else
25:      $without\_improvement \leftarrow 0$ 
26:      $s^* \leftarrow s'$ 
27:   End If
28: End While
29: Return  $s^*$ 

```

The HGCA starts generating the initial population using four algorithms: three versions of the K-means heuristic, whose initialization methods are random; K-means ++; PCA_part; and an algorithm that generates random solutions. Then, the initial population is evaluated and to each chromosome is attributed a fitness value.

From this point on, the algorithm enters its main loop. The HGCA selects the parents using the roulette-wheel technique. Then, the single-point crossover is applied to them generating the children chromosomes. After the crossover, it is applied a uniform-mutation operator. At this point the K-means (described in section 2.1) makes the refinement of all chromosomes in current population. K-means was chosen because it is a quick method that generates good results. Next, the current population is evaluated and each chromosome has its fitness value updated. Then, the elitism is applied. Throughout the evolution process a new population replaces the previous one. The stopping criterions used are the maximum number of generations or a fixed number of generations reached without improvement in the fitness value of the best chromosome.

4. EXPERIMENTS AND RESULTS

Using the algorithms described in the previous section, the experiments were performed with eight real data sets available in public repositories. The data was separated into training and testing sets with, respectively, 10% and 90% of the total samples. The training sets were used to adjust the parameters of the Tabu Search, the GCA and the HGCA. Table 1 shows the characteristics of the real data sets and the partition of data into training and testing sets.

Table 1 - Characteristics of the data sets used in the experiments.

Name	n° patterns	n° attributes	n° classes	n° patterns Test data set	n° patterns Training data set
Iris	150	4	3	135	15
Wine	178	13	3	161	17
Vehicle	846	18	4	762	84
Cloud	1024	10	10	922	102
Segmentation	2310	19	7	2079	231
Spam	4601	57	2	4141	460
Pendigits	10992	16	10	9893	4099
Letter	20000	16	26	18000	2000

Each algorithm, nondeterministic, ran 10 times for each testing set. In addition, SSE's averages and runtime's averages were calculated.

The algorithms were implemented in C and tests were performed on a machine with the following configuration: AMD Athlon 64 X2 Dual Core 3800 +, 512KB Cache; Motherboard ASUS M2NPV-VM; 2 GB DDR2 RAM 533; HD SATA 100 GB; Ubuntu OS 6.4.0.

Table 2 presents the results achieved with the application of the algorithms on the data sets listed in Table 1. For each data set, experiments were performed with three different values of K, being one of the values equal to the number of existing classes in the data.

A less rigorous analysis of the results presented in Table 2 shows that the HGCA obtained the best results in all 24 tests with only three draws, indicating a possible superiority of metaheuristics in comparison with the other algorithms evaluated, although this is not statistical evidence.

More importantly, HGCA's performance was consistently superior to other algorithms evaluated even in the largest base (Letter), indicating that metaheuristics present a satisfactory performance in large databases.

In order to complement the assessment it was conducted a runtime analysis to quantify the difference between the CHGA and the other algorithms. Table 3 presents the execution time (in seconds) and Table 4 the standard deviation in relation to the SSE criterion.

The results presented in Table 3 show that K-means initialized with the PCA-part had the lowest runtime for all databases. This is due to a good choice of the initial centroids that required few iterations of the algorithm to reach the final solution. GCA and HGCA presented similar runtimes when applied to smaller data sets (Iris and Wine). However, a significant difference starts to appear on larger data sets. This is due to the local search performed in all individuals at each new generation in HGCA.

Table 2 – SSE’s averages obtained with the algorithms evaluated.

Name	k	Agglo. Hier.	kmeans	Tabu Search	kmeans (kmeans++)	kmeans (PCA Part)	GCA	HGCA
Iris	2	142,505	140,015	156,687	140,015	140,015	140,606	140,015
	3	73,48	78,63	73,63	72,87	72,90	75,76	72,86
	4	62,68	54,38	60,00	52,98	51,45	52,98	51,45
Wine	2	4,094E+06	4,079E+06	4,674E+06	4,078E+06	4,079E+06	4,088E+06	4,077E+06
	3	2,549E+06	2,134E+06	2,496E+06	2,237E+06	2,331E+06	2,120E+06	2,096E+06
	4	2,262E+06	1,202E+06	1,722E+06	1,210E+06	1,201E+06	1,224E+06	1,195E+06
Veh.	3	5,174E+06	4,559E+06	5,322E+06	4,612E+06	4,506E+06	4,645E+06	4,505E+06
	4	4,974E+06	3,489E+06	4,301E+06	3,276E+06	3,286E+06	3,459E+06	3,226E+06
	5	3,317E+06	2,434E+06	3,723E+06	2,257E+06	2,138E+06	2,756E+06	2,138E+06
Cloud	9	8,693E+06	7,668E+06	1,602E+07	6,576E+06	6,539E+06	9,198E+06	6,321E+06
	10	8,512E+06	6,762E+06	1,425E+07	5,719E+06	5,350E+06	8,466E+06	5,228E+06
	11	8,485E+06	6,545E+06	1,254E+07	5,169E+06	4,673E+06	7,936E+06	4,472E+06
Segm.	6	3,623E+07	1,519E+07	1,666E+07	1,431E+07	1,351E+07	1,849E+07	1,339E+07
	7	3,607E+07	1,400E+07	1,497E+07	1,297E+07	1,203E+07	1,673E+07	1,188E+07
	8	3,593E+07	1,190E+07	1,375E+07	1,152E+07	1,066E+07	1,640E+07	1,052E+07
Spam	2	1,397E+09	8,98888E+08	9,478E+08	1,074E+09	8,98888E+08	9,166E+08	8,98883E+08
	3	1,331E+09	5,999E+08	7,576E+08	5,314E+08	5,999E+08	7,102E+08	5,020E+08
	4	1,053E+09	5,065E+08	6,252E+08	3,689E+08	5,065E+08	6,025E+11	3,109E+08
Pend.	9	1,158E+08	4,897E+07	4,807E+07	4,788E+07	4,799E+07	5,514E+07	4,722E+07
	10	1,080E+08	4,574E+07	4,427E+07	4,537E+07	4,485E+07	5,163E+07	4,422E+07
	11	1,075E+08	4,403E+07	4,301E+07	4,285E+07	4,335E+07	4,981E+07	4,185E+07
Letter	25	1307900,0	563782,0	561161,0	564735,0	561189,0	652184,0	557366,0
	26	1302360,0	555925,0	551063,0	554031,0	555722,0	647331,0	549394,0
	27	1304280,0	549175,0	544910,0	549874,0	548137,0	632960,0	541938,0

Table 3 – Evaluated algorithms runtime’s averages in ten tests.

Name	k	Agglo. Hier.	kmeans	Tabu Search	kmeans (kmeans++)	kmeans (PCA Part)	GCA	HGCA
Iris	2	0,07	< 0,01	55,87	< 0,01	< 0,01	0,99	1,01
	3	0,06	< 0,01	57,40	< 0,01	< 0,01	1,52	1,38
	4	0,06	< 0,01	62,70	< 0,01	< 0,01	2,12	1,60
Wine	2	0,20	< 0,01	78,17	< 0,01	< 0,01	1,78	2,61
	3	0,20	0,01	81,53	< 0,01	< 0,01	2,31	3,23
	4	0,15	0,02	99,90	0,02	< 0,01	4,15	4,21
Veh.	3	15,77	0,15	145,16	0,15	0,01	10,47	25,27
	4	15,96	0,16	175,09	0,11	0,01	14,04	31,89
	5	15,41	0,13	203,21	0,14	0,02	19,18	43,54
Cloud	9	18,63	0,46	241,41	0,17	0,02	20,72	100,22
	10	18,65	0,57	260,61	0,20	0,09	19,31	106,83
	11	18,63	0,51	283,01	0,16	0,04	18,56	125,68
Segm.	6	322,16	0,57	631,73	0,43	0,09	30,00	242,24
	7	321,83	0,69	712,40	0,53	0,09	40,59	200,59
	8	322,12	0,61	798,78	0,77	0,20	38,62	214,59
Spam	2	6267,00	0,87	1541,92	0,54	0,24	92,90	357,29
	3	6120,00	1,69	2012,22	0,63	0,43	95,66	670,64
	4	6185,00	2,93	2496,00	0,87	0,56	150,03	1026,22
Pend.	9	27599,00	4,08	1333,97	4,21	0,59	220,67	1042,93
	10	30946,00	4,24	1451,93	4,72	0,37	279,85	1082,67
	11	31765,00	4,39	1580,43	4,49	0,67	239,63	1186,29
Letter	25	192900,00	42,42	3066,80	48,30	11,22	595,06	17093,20
	26	192840,00	53,24	3183,10	42,12	6,98	1000,15	19701,70
	27	192540,00	50,27	3306,40	47,52	7,78	684,17	19440,11

4.1. Experiments using statistical methods

In order to perform a more precise and rigorous evaluation of the experiments, we chose to run a

statistical analysis of results. To this end, we used the Friedman Test, which ordinales the algorithms by assigning to each of them a rank; and the Nemenyi Test, that identifies pairs of statistically different algorithms.

The experiments using statistical methods in multiple were performed in two steps. The first step consisted in verify the null hypothesis that all algorithms were statistically equivalent based on the results presented in Table 2. Friedman's test result — performed with a significance level of 5% — indicated that the null hypothesis could be rejected. Thus, the opposite hypothesis was confirmed, in other words, there was at least one pair of statistically different algorithms.

The second step consisted in identify which pairs have a significant difference. According to the Nemenyi Test, with a significance level of 5%, two algorithms are considered different if the difference between their job averages is at least equal to a critical difference CD. The test showed that HGCA was significantly better than the following algorithms: Agglomerative Hierarchical, K-means randomly initialized, Tabu Search and GCA. In the remaining pairs of algorithms the differences were smaller than the critical difference CD. Therefore, it was not possible to establish whether they were equal or different to one another.

5. CONCLUSIONS AND FUTURE RESEARCH

This paper proposes a Hybrid Genetic Clustering Algorithm (HGCA), whose initial population was generated partly by clustering methods, which perform a global search procedure associated with local search. Such improvements are intended to provide solutions for search problems closer to the global optimum.

To analyze if the HGCA performed better than other algorithms while solving the same problem, this paper conducted experiments with eight real data sets available in public repositories.

Another six algorithms were implemented: Agglomerative Hierarchical; three versions of K-means, differing only in terms of initialization methods (Random, K-means ++ and PCA_Part); Tabu Search and Genetic Clustering Algorithm.

Throughout the experiments, the seven algorithms were compared in multiple areas in order to find out if there was a general improvement trend regarding the HGCA. In this case, statistical analysis showed that, in general, the Hybrid Genetic Clustering Algorithm provided better results than other algorithms tested — agglomerative hierarchical, K-means randomly initialized, the tabu search and the classic version of Genetic Clustering Algorithm. However, comparing the six previously mentioned algorithms, the analysis does not suggest whether there is difference between them.

An important contribution of this work was the development of a Hybrid Genetic Clustering Algorithm (HGCA). Its initial population is partly generated by three versions of K-means, securing a better initial population than if it was only randomly-generated. Moreover, HGCA's global search procedure is associated with local search method, providing an efficient mechanism to explore potential solutions. Another important contribution of this study was the extensive comparative experiments performed, showing that the proposed method delivers superior results.

Since the algorithm proposed in this work uses three versions of the K-means for generating the initial population, a possible continuation of this study would examine the application of other heuristics with the same purpose. Another possible continuation of this study is to examine the inclusion of other local search heuristics in HGCA's global search procedure.

Increasing the number of problems ensures a more efficient comparison with the statistical analysis conducted in this works. Therefore, it is recommended as future work to carry out more experiments using other real problems, as well as a greater number of data sets, in order to achieve a more accurate assessment of the algorithms studied.

In experiments with the largest data set (Letter), HGCA obtained results in all tests superior to the other algorithms, indicating that the metaheuristics performance is satisfactory at large basis. A possible continuation for this work is to conduct experiments to analyze the performance of HGCA in even larger

data sets.

Another possible continuation of this study is to implement the algorithms with smaller runtimes in the same interval of time HGCA required in the previous experiments. In other words, each algorithm would be run several times until the maximum time spent by HGCA. Thus, all runtimes would be the same and the algorithms would be evaluated only by the quality of their results.

Finally, a possible continuation of this work would be applying an approach suggested by [38] to HGCA. According to it, only a certain percentage of the population — considered elite-individuals — would receive heuristic improvements. Therefore, the local search would be limited to certain group of individuals, which would reduce the algorithm's runtime. However, it would require further analysis to determine if this approach to reduce the execution time could pose a threat to the quality of the solutions found.

RECEIVED JULY, 2010

REVISED JUNE 2011

REFERENCES

- [1] ALSULTAN, K. (1995): A Tabu Search Approach to the Clustering Problem, **Pattern Recognition**, 28, 1443-1451.
- [2] ARTHUR, D. E VASSILVITSKII, S. (2007): K-means++: The Advantages of Careful Seeding. **Symposium on Discrete Algorithms (SODA)**.
- [3] BABU, G. P. E MURTY, M. N. (1993): A Near-Optimal Initial Seed Value Selection in K-means Algorithm using a Genetic Algorithm, **Pattern Recognition Letters**, 14, 763-769.
- [4] BERGER, D.; GENDROM B. E POTVIN J. Y. (1999): "Tabu Search for a Network Loading Problem", **Institute for Systems Research**, Technical Report 99-23.
- [5] COLE, R. M. (1998): **Clustering with genetic algorithms**. Thesis - (Master of Science), Department of Computer Science, University of Western Australia.
- [6] DAVIS, L. D. (1991): **Handbook of Genetic Algorithms**. New York: Van Nostrand Reinhold.
- [7] DEMSAR, J. (2006): Statistical Comparisons of Classifiers over Multiple Data Sets. **Journal of Machine Learning Research**, 7, 1–30.
- [8] DUDA, R., P. HART, & D. STORK. (2001): **Pattern Classification**. John Wiley & Son, Chichester.
- [9] FAYYAD, U. M., PIATETSKY-SHAPIO G., SMYTH P. E UTHURUSAMY R. (1996): **Advances in Knowledge Discovery and Data Mining**, MIT Press, Massachussets.
- [10] EVERITT, B. S.; LANDAU, S.; LEESE, M. (2001): **Cluster Analysis**. [S.l.]: Hodder Arnold Publication, N. York.
- [11] FORGY, E. W. (1965): Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications, **Biometrics**, 21, 768-780.
- [12] FUKUNAGA, K. (1990): **Introduction to Statistical Pattern Recognition**, Academic Press, New York.
- [13] GLOVER, F. (1986): Future paths for Integer Programming and links to Artificial Intelligence. **Computers and Operations Research**, 5:553_549.
- [14] GLOVER, F. AND LAGUNA, M. (1997): **Tabu Search**. Kluwer Academic Publishers, N.York.

- [15] GOLDBERG, D. E. (1989): **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley Longman Publishing Co., In, Bostonc.
- [16] HANSEN, P. (1986): The steepest ascent mildest descent heuristic for combinatorial programming. **In Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy.**
- [17] HARTIGAN, J. A. (1975): **Clustering Algorithms**, Wiley, New York.
- [18] IMAN, L.; DAVENPORT, J. M. (1980): Approximations of the Critical Region of the Friedman Statistic. **Communications in Statistics**, 9, 571–595.
- [19] JAIN, A. K., MURTY, M. N. E FLYNN, P. J. (1999): Data Clustering: A Review, **ACM Computing Surveys**, 31, 264-323.
- [20] JOLLIFFE, I. T. (1986): **Principal Component Analysis**, Springer-Verlag, New York.
- [21] KAUFMAN, L. E ROUSSEEUV P. J. (1990): **Finding Groups in Data – An Introduction to Cluster Analysis**, John Wiley, New York.
- [22] LAGUNA, M. (1994): **A guide to implementing Tabu Search**. Boulder, University of Colorado,
- [23] MEZ, P. (2000): **Memetic Algorithm for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies.**
- [24] MICHALEWICZ, Z. (1996): **Genetic Algorithms + Data Structures = Evolution Programs**, 3rd edition, Springer-Verlag, Heidelberg.
- [25] MOSCATO, P. (1999): **Memetic algorithms: a short introduction**. In: Corne, D.; Dorigo, M.; Glover, F. ed. *New Ideas in Optimization*. London: McGraw-Hill, 219–234.
- [26] MURTY, C. A. E CHOWDHURY, N. (1996): In Search of Optimal Clusters using Genetic Algorithms, **Pattern Recognition Letter**, 17, 825-832.
- [27] NEMENYI, P. B. (1963): **Distribution-free multiple comparisons**. Tese (Doutorado) - Princeton University.
- [28] PERIM, T. G. (2008): **Uso de métodos de inicialização combinados ao Simulated Annealing para resolver o problema de agrupamento de dados**. 76f. Dissertação - (Mestrado em Informática) - UFES, Universidade Federal do Espírito Santo.
- [29] PERIM, T. G. E VAREJÃO, F. M. (2008): Aplicação de método baseado em *pca* para inicialização do *Simulated Annealing* no problema de particionamento de dados, **XL Simpósio Brasileiro de Pesquisa Operacional**.
- [30] PERIM, T. G.; WANDEKOKEM D. E. AND VAREJÃO, M. F. (2008): K-Means Initialization Methods for Improving Clustering by Simulated Annealing, **Lecture Notes in Computer Science, Advances in Artificial Intelligence – IBERAMIA**.
- [31] RARDIN, R. L. E UZSOY, R. (2001): Experimental evaluation of heuristic optimization algorithms: a tutorial. **Journal of Heuristics**, 7, 261-304.
- [32] RAYWARD-SMITH, V. J. (2005): Metaheuristics for Clustering in KDD, **IEEE Congress on Evolutionary Computation**, 3, 2380-2387.
- [33] SCHAEFER, A. (1996): Tabu search techniques for large high-school timetabling problems. **In Proceedings of the 30th National Conference on Artificial Intelligence**, 363_368.
- [34] SU, T. e DY, J. (2004): A Deterministic Method for Initializing K-means Clustering, **IEEE International Conference on Tools with Artificial Intelligence**, 784-786.

- [35] SU, T. and DY, J. (2007): In Search of Deterministic Methods for Initializing K-means and Gaussian Mixture Clustering, **Intelligent Data Analysis**, 11, 319-338.
- [36] XU, L. E WUNSCH, D. (2005): Survey of Clustering Algorithms. **IEEE Transactions on Neural Networks**, 16, n° 3.
- [37] WERRA, D. (1989): Tabu Search Techniques: A Tutorial and an Application to Neural Networks. **OR Spektrum**, 11:131_141.
- [38] YEN, J.; LEE, B. (1997): A simplex genetic algorithm hybrid. **IEEE International Conference on Evolutionary Computation (ICEC'97)**, 4, Indianapolis (USA). IEEE Press, 175–180.
- [39] ZHANG, T., RAMAKRISHNAN, R. E LIVNY, M. (1997): BIRCH: A New Data Clustering Algorithm and Its Application, **Data Mining and Knowledge Discovery**, 1, 141-182.