We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

7

Tune Up of a Genetic Algorithm to Group Documentary Collections

José Luis Castillo Sequera
University of Alcala, Department of Computer Science, Madrid
Spain

1. Introduction

Both in industry and science there are some real problems regarding the optimization of difficult solution characterized by computational complexity, because the available exact algorithms are inefficient or simply impossible to implement. The metaheuristics (MHs) are a family of approximate methods of general purpose consisting in iterative procedures that guide heuristics, intelligently combining different concepts to explore and exploit properly the search space [12]. Therefore, there are two important factors when designing MHs : intensification and diversification. The diversification generally refers to the ability to visit many different regions of search space, while intensification refers to the ability to obtain high quality solutions in these regions. A search algorithm must achieve a balance between these two factors so as to successfully solve the problem addressed.

On the other hand, Information Retrieval (IR) can be defined as the problem of information selection through a storage mechanism in response to user queries [3]. The Information Retrieval Systems (IRS) are a class of information systems that deal with databases composed of documents, and process user's queries by allowing access to relevant information in an appropriate time interval. Theoreticly, a document is a set of textual data, but technological development has led to the proliferation of multimedia documents [4].

Genetic Algorithms (GAs) are inspired by MHs in the genetic processes of natural organisms and in the principles of natural evolution of populations [2]. The basic idea is to maintain a population of chromosomes, which represent candidate solutions to a specific problem , that evolve over time through a process of competition and controlled variation. One of the most important components of GAs is the crossover operator [7]. Considering all GA must have a balance between intensification and diversification that is capable of augmenting the search for the optimal, the crossover operator is often regarded as a key piece to improve the intensification of a local optimum. Besides, through the evolutionary process, every so often there are species that have undergone a change (mutation) of chromosome, due to certain evolution factors, as the mutation operator is a key factor in ensuring that diversification, and finding all the optimum feasible regions.

Efficiently assigning GA parameters optimizes both the quality of the solutions and the resources required by the algorithm [13]. This way, we can obtain a powerful search

algorithm and domain independent, which may be applied to a wide range of learning tasks. One of the many possible applications to the field of IR might be solving a basic problem faced by an IRS: the need to find the groups that best describe the documents, and allow each other to place all documents by affinity. The problem that arises is in the difficulty of finding the group that best describes a document,since they do not address a single issue, and even if they did, the manner the topic is approached can also make it suitable for another group. Therefore, this task is complex and even subjective as two people could easily assign the same document to different groups using valid criteria.

Clustering is an important tool in data mining and knowdledge discovery because the ability to automatically group similar items together enables one to discover hidden similarity and key concepts [10]. This enables the users to comprehend a large amount of data. One example is searching the World Wide Web, because it is a large repository of many kinds of information, many search engines allow users to query the Web, usually via keyword search. However, a typical keyword search returns a large number of Web pages, making it hard for the user to comprehend the results and find the information that really needs. A challenge in document clustering is that many documents contain multiple subjects.

This paper presents a GA applied to the field of documentation, the algorithm improved itself by refining its parameters, offering a balance between intensification and diversity that ensures an acceptable optimal fitness along an unsupervised document cluster.

2. Documentary base

In this study we make use of two collections, the "Reuters 21578" collection and a Spanish documentary base that includes editorials of "El Mundo" from 2006 and 2007 in an open access format.

Reuters Documentary Base consists of real news wires that appeared in Reuters in 1987, this collection is becoming a standard within the domain of the automatic categorization of documents and is used by many authors in this area. The collection consists of 21578 documents distributed in 22 files. We developed a documentary process named NZIPF [6] [11] to generate documentary vectors that feed the system.

The documentary process consists of several stages of document processing, each of which represents a process that was developed on the base document to obtain documentary vectors more efficiently.

The first step is the called process of *Filter* whose main objective is to define the documents of the documental base with the purpose of having documents that belong to a single category, that which will allow to have a smaller complexity in the treatment of the documents. Then, the purpose of the process of *Zonning* on the documents is the one of obtaining the free text of each document. Next, we use a process of *Stop List*, we extract the terms of the text of the document where each one of the extracted words will be compared with a list of empty words that will eliminate the words that don't have interest or they lack own meaning. Then, the words will be able to suffer a process of cutting of their roots "*Stemming*", in our case, we have implemented and used an algorithm of Porter in English and another in Spanish. In this step, the *frequency* of the obtained terms is calculated, for all

the documents of our documental base, with the purpose of being able to know that terms are more used in each one of the documents; and then with this information to be able to carry out a process of selection of those terms that are more representative. The following step will consist on selecting those terms with discriminatory bigger power to proceed to its normalization We apply the law of *Zipf*, we calculate the Point of Goffman [3] and the transition area that it allows us to obtain the terms of the documental base. Finally, we assign weight using a function *IDF* (Invert Document Frecuency) developed for Salton [4] that uses the frequency of a word in the document. After all these processes, we obtain the characteristic vectors of documents in the collection document.

The process is outlined in Figure 1.

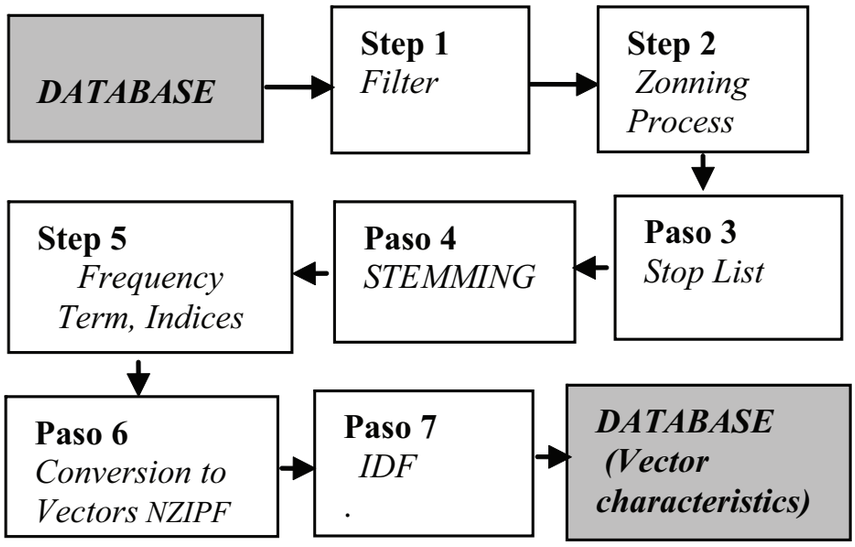


Fig. 1. Documentary process conducted

On the other hand, within the testing environment there should be a user to provide documents that are meant to be grouped. The role of the user who provides documents will be represented by the samples of "very few (20), few (50), many (80) and enough (150)" documents, with the requirement that belonged to only two categories of Reuters or distribution of Editorials in Spanish represented by their feature vectors stemmer. Figure 2 shows the documentary environment [10] that we used for the experiments, it is important to note that, unlike the algorithms of the type monitored, where the number obtained groups needs to be known, our algorithm will evolve to find the most appropriate structure, forming the groups by itself.

Due to the nature of simulation of GA, its evolution is pseudo-random, this translates into the need for multiple runs with different seeds to reach the optimal solution. The generation of the seed is carried out according to the time of the system. For this reason, the experiments with GA were made by carrying out five executions to each of the samples taken from experimental collections [1]. The result of the experiment will be the best fitness obtained and their convergence. To measure the quality of the algorithm, the best solution obtained and the average of five runs of the GA must be analized.

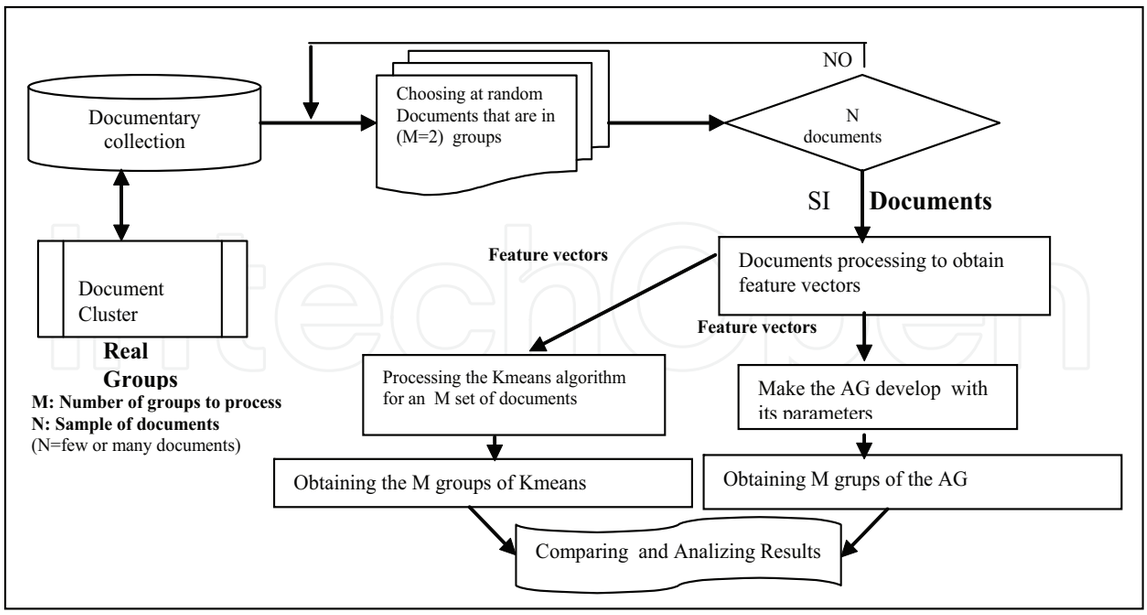


Fig. 2. Experimental environment used in the tests with the GA.

3. Genetic algorithm for document clustering

3.1 Individuals

The population consists of a set of individuals, where each of it is made of a linear chromosome that is represented through a tree structure (hierarchical structure). An individual shall formed on a binary tree structure *cluster all documents* prepared at the top, where each document consists of a *feature vector.* The vector will consist of the weighted values of the frequencies of the stemmer terms that have been selected to implement the document processing scheme [4]. This representation will be attempted to evolve so that the chromosome will undergo genetic changes and find the groups "Clusters" more appropriate for all documents of the IRS. Within the root node we will have our fitness function *(fitness)* that measure the quality of the resulting clustering. Depending on the number of documents that need to be processed and the depth (height) of the tree you want to create, chromosome may be of variable length.

The Figure 3 shows the initial generation (0), a scheme of tree-based representation is adopted in order to allow the encoding of sufficiently complex logical structures within a chromosome. The search area for the GA is the space of all possible trees that can be generated, resulting from the whole relevant functions and terminals. This way we can evolve individuals of various shapes and sizes [8], allowing evolution to decide what are the best settings.

Although the initial population is random, there is a defined set of parameters governing the establishment of such individuals. For example, *there should not be created in the initial set two equal individuals,* for this production rules are created to ensure the compliance with this condition. The above mentioned rules require that the building grammar of each individual nodes takes place in Preorder.

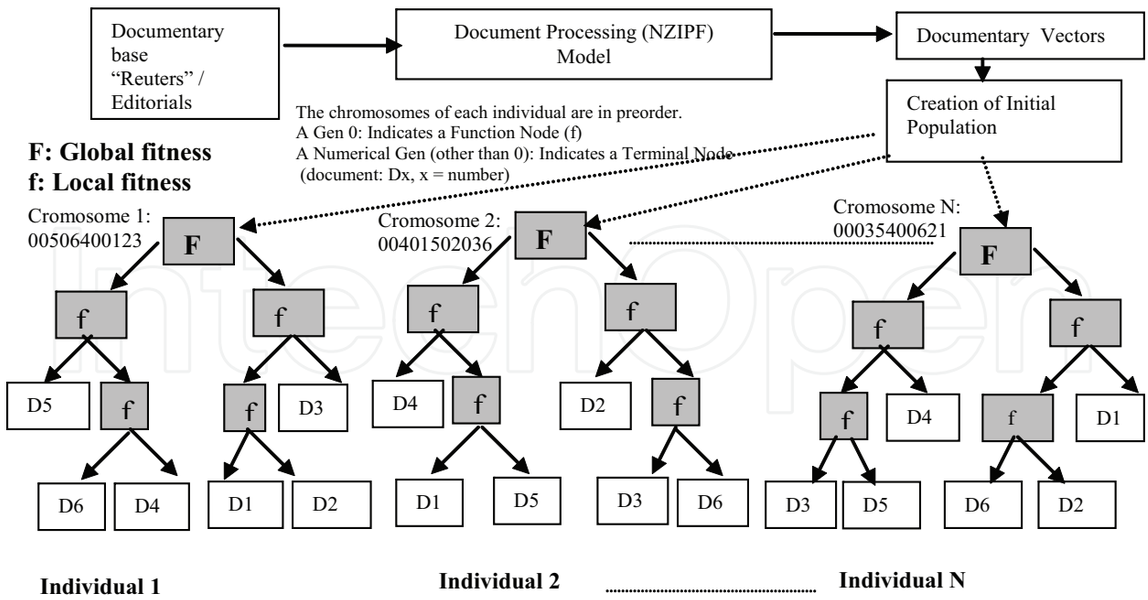


Fig. 3. Initial Population of Individuals GA (generation "0")

3.2 Production operators

The production operators are applied to each new generation. One or two individuals can be taken to produce new individuals for next generation by applying the transformations imposed by the operator. Both mutation operators and crossover will be implemented indistinctably. Both operators depend on a mutation probability and / or cross that is assigned to GA [7].

A *mutation operator* is applied on nodes (documents), selecting an individual from the population using the tournament method, and then randomly select a pair of terminal nodes of that individual to mutate its terminal nodes, generating a new individual transposing the nodes that have been chosen (see figure 4).

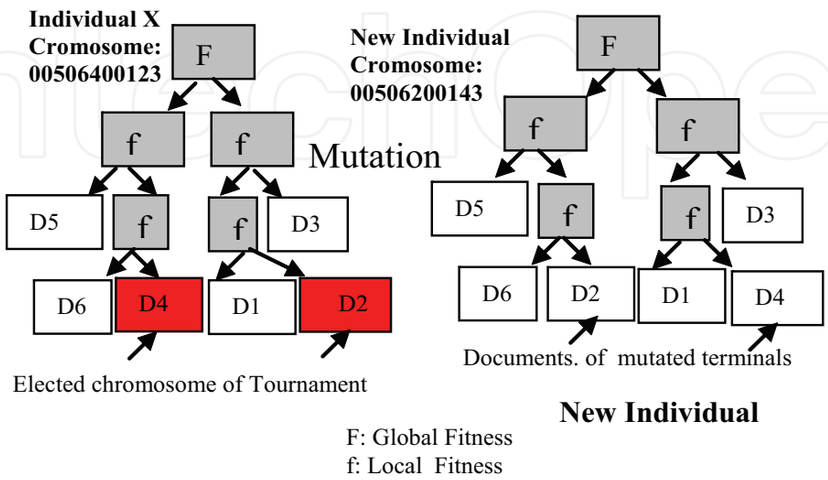


Fig. 4. Basic mutation operator applied to terminal

For the crossover operator, an operator based on *mask crossover* [9] is applied, which selects through tournament method two parent individuals, randomly chooses the chromosome of one parent to be used as "*crossover mask of the selected individual¨*. The crossing is done by analyzing the chromosome of both parents. If both chromosomes have at least one function node (node 0), the chosed father mask is placed, but if we find documents in the chromosomes of both parents, then, the father "*not elected*" document will be selected and we'll use it as pivot on the father "*elected*" (mask) to make the crossing that corresponds to the mentioned father, while interchanging the chromosomes of the mentioned father. This *creates a new individual,* and ensure that in the given chromosome set there are the same structural characteristics of the parents but we only incorporate it in the population if the child has a better fitness than their parents. (see figure 5).

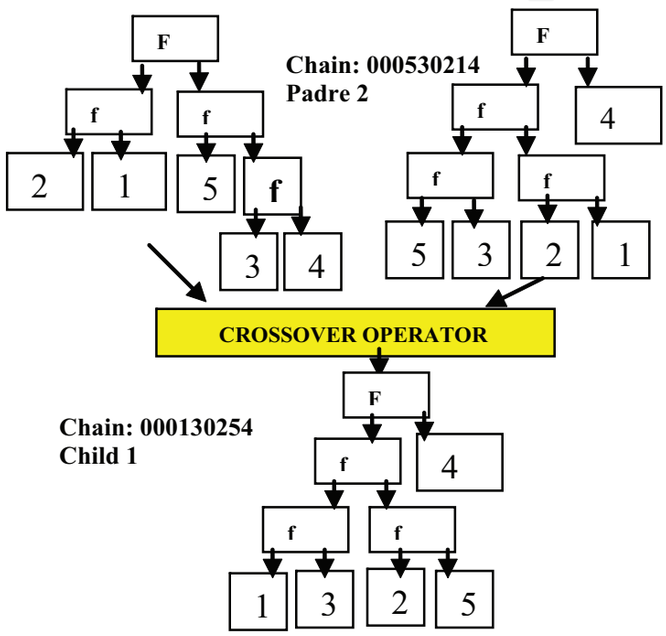


Fig. 5. Crossover operator (crossover mask)

3.3 Selection

After we evaluate population's fitness, the next step is chromosome selection. Selection embodies the principle of 'survival of the fittest' [5]. Satisfied fitness chromosomes are selected for reproduction, for it, we apply the method of selection of the tournament, using a tournament of 2, and we apply Elitism in each generation [2].

4. Parameter control

For its size, and the influence that small changes have on the behavior of the GA during the experiments [1], the choice of parameter values that are going to be used appears as a critical factor. For their election we paid attention to the variation of the GA performance indicators when it changed the value of any of these, specifically the evolution of the successes and the evolution of "*fitness*". Therefore, these parameters are very important parts as they directly influence the performance of the GA [13]. These parameters can be treated independently,

but the overall performance of the algorithm does not depend exclusively on a single parameter but on a combination of all parameters. Many researchers pay more attention to some parameters than others, but most agree that the parameters that should be under controlare: selection schemes, population size, genetic variation operators and rates of their chances.

Because GA have several **parameters** that must be carefully chosen to obtain a good performance and avoid premature convergence, in our case and *after much testing,* we opted for the control of parameters, and some strategies such as:

To control the *population size* we use the strategy called GAVAPS (Genetic variation in population size) proposed by Michalewicz [9] using the concept of age and lifetime. When creating the first generation all individuals are assigned a zero age, referring to the birth of the individual, and every time a new generation is born the age of each individual increases by one. At the same time an individual is born it is assigned a lifetime, which represents how long it will live within. Therefore, the individual will die when it will reach the given age. The lifetime of each individual depends on the value of its fitness compared to the average of the entire population. Thus, if an individual has better fitness will have more time to live, giving it greater ability to generate new individuals with their features. In our case, we allow each generation to generate new individuals with similar characteristics with this strategy.

Therefore, we adopt this approach essentially the best individuals from each generation, and apply it to maintain *elitism* in the following generations, thus ensuring optimum intensification of available space, while keeping them during their lifetime [9]. However, to ensure diversity we *randomly* generate *the remaining individuals* in each generation. This way, we explore many different regions of the search space and allow for balance between intensification and diversity of feasible regions.

In all cases, the population size has been set at 50 individuals for the experiments conducted with samples following the suggestion of [1], which advises working with a population size between *l* and *2l* in most practical applications (the length of chromosome l) In our case, "l" the length of our chromosome is always equal to:

2 * number of documents to cluster -1.

On the hand, we use two measures of function fitness to calculate the distance and similarity between documents and to be able to form better cluster (see table 1).

Distance Euclidean	$d_{ij} = \sqrt{\sum_{k=1}^{t} (x_{ik} - x_{jk})^2}$
Coefficient of correlation of Pearson (Similarity)	$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}_i}{\sigma x_i} \right) \left(\frac{x_j - \bar{x}_j}{\sigma x_j} \right)$
Fitness Global	Min (α Distance(Documents $_i$) + (1- α) (1/ Similarity(Documents $_i$)))

Table 1. Measures of the Function

with x_i and x_j the characteristic vectors of the documents that we are grouping, ***"n"*** the number of examples and σx_i , σx_j are the standard deviation of x_i and x_j and where: ***a:*** it will be the parameter that adjustment the distance and similarity. The fitness function is used to minimize the distance between the documents and maximize the similarity between them.

Therefore, for the experiments with our experimental environment, we used samples of documents "very few (20), few (50), many (80) and enough (150)" documents with the requirement that they belonged only to two categories of Reuters collections or Editorials. Each of the samples processed with five different seeds, and each of the results are compared with the method "*Kmeans.*" Then, each experiment was repeated by varying the rate of probability of genetic algorithm operators, using all the parameters shown in table 2 up to find that value of α that best fit the two metrics hat combine in our function fitness.

Parameters	**Values**
Population size (tree number)	50
Número de evaluaciones (Generaciones)	5000 maximum
Tournament size	2
Mutation Probability (Pm)	0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0, 5, 0.7
Crossover Probability (Pc)	0.70,0.75,0.80,0.85,0.90,0.95
Document cuantity	Very Few, Few, Many, enough
α coefficients	0.85 (best value found)
Depth Threshold	7 /10

Table 2. Parameters taken into consideration for the Genetic algorithm with composite function

4.1 Studies to determine the value of α in the GA

We use the distribution Reuters 21 of be that greater dispersion across your documents and apply the GA varying the value of α in each of the tests with the usual parameters, always trying to test the effectiveness of the GA. We analyzed the relationship between fitness and the value of α using the values in table 2. (the results are shown in table 3 and figure 6).

In figure 6, we can see that there is an increased dispersion of fitness values over 0.85, due to the increased contribution of Euclidean distance which makes it insensitive to fitness to find the clusters. The results, suggest that a value of α close to 0.85, provides better results because it gives us more effective in terms of number of hits, and a better fitness of the algorithm. This was corroborates with other distribution.

Documents	α	Generatión	Best *Fitness*	Average Middle *Fitness*	Hits	Effectiviness (%)
20	0,75	1436	0,25291551	0,46489675	15	75,0
20	0,80	1592	0,20298477	0,47026890	16	80,0
20	0,85	2050	0,15255487	0,24504483	17	85,0
20	0,90	3694	0,15266796	0,25909582	17	85,0
20	0,95	1520	0,15319261	0,24596829	17	85,0
50	0,75	3476	0,25290429	0,28744261	35	70,0
50	0,80	3492	0,20285265	0,27862528	36	72,0
50	0,85	3355	0,15312467	0,29128428	36	72,0
50	0,90	2256	0,15318358	0,28347470	36	72,0
50	0,95	2222	0,15345986	0,27863789	36	72,0
80	0,75	3049	0,25704660	0,36871676	61	76,2
80	0,80	1371	0,20782096	0,33303315	61	76,2
80	0,85	2131	0,15784449	0,34447947	62	77,5
80	0,90	1649	0,15815252	0,32398087	62	77,5
80	0,95	2986	0,17796620	0,36009861	61	76,2
150	0,75	2279	0,26194273	0,29866150	91	60,6
150	0,80	1273	0,20636391	0,22933754	93	62,0
150	0,85	3257	0,15468909	0,27518240	94	62,6
150	0,90	1136	0,25482251	0,28218144	94	62,6
150	0,95	2452	0,25456480	0,26788158	91	60,6
250	0,75	3617	0,25754282	0,31144435	120	48,0
250	0,80	3274	0,20844638	0,25112189	121	48,4
250	0,85	3066	0,15805103	0,19299910	121	48,4
250	0,90	2343	0,20634355	0,20432140	121	48,4
250	0,95	2047	0,25541276	0,27844937	120	48,0

Table 3. Results of tests with GA, takong different samples of documents with the distribution 21 of the Reuters collection, to determine the best value for α

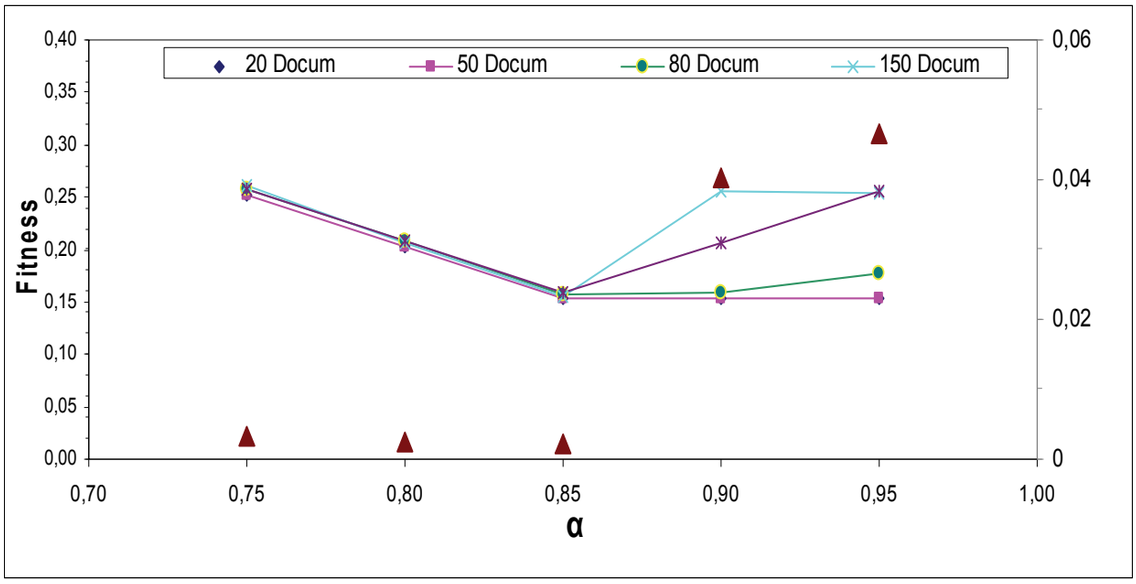


Fig. 6. Best Fitness versus α values for different samples of documents of the Reuters Collection: Distribution 21

4.2 Tests to determine the value of the rate of mutation operator and crossover operator rate

We began conducting an analysis of system behavior by varying the rate of mutation operator in a wide range of values to cover all possible situations. During experiments using different samples distribution Reuters. Thus, for the rate of mutation operator discussed a wide range of values in the range of: 0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0, 5, 0.7; that allowed us to apply the mutation operator of GA in different circumstances and study their behavior. For the study to determine the optimal value of the rate of crossover operator, is traced the interval from 0.70 to 0.95; value high, but oriented to frequently apply the operator we designed because that an optimum value for the *mutation probability* is much more important than the crossover probability, and choose to make a more detailed study of the odds ratio in our experiments. As a quality index value of the operator was given to the number of hits of the GA.

As for the *size of the tournament,* the value 2 has been chosen, because the binary tournament has shown a very good performance in a large number of applications of EAs. Although determining a optimal *fitness* function is not one of the fundamental objectives of this experiment, we have tried to add in a single value the measuring results as powerful and distinct as are the Euclidean distance and the Pearson correlation coefficient (based on cosine similarity).

Therefore, to find and the adjustment coefficient α that governs the weight that is to be given to both the distance as the inverse of similarity of the cluster documents, we've made many parameter controlled tests in order to obtain a value that allows an adequate contribution of both metrics with respect to fitness., finally finding a value for of 0.85.

The **number of maximum generations** the system has been set to is 5000, but this parameter may vary depending on the convergence of the algorithm. As for the **number of stemmer terms** to be used for representing the feature vectors of each of the documents we have used the terms, which have been selected through the NZIPF processing method [6][11].

Finally, we have established a limit called the **threshold of depth** for individuals (trees). Such a threshold, in the case of "*very few and few documents*" take the value of 7, and for the"*many and enough documents*" is set 10. To analyze the results, and to verify their effectiveness, we compared the results of the GA with the existing real groups of the document collection [6], and also compared the results with another supervised type of clustering algorithm in optimal conditions (Kmeans). We analized the following:

a. **Cluster efectiveness:** It is the most important indicator of the comparison of results considering the quality of the cluster. An analyzing process was carried out to see the successes achieved with the best fitness of GA, and also the average scores in all executions of the GA.
b. **Fitness evolution.** Analysis was carried out to see the evolving fitness in each of the performances, assessing their behaviour and successes of the GA when varying the probability rate.
c. **Convergence of the algorithm**: In which process the GA obtains the best fitness (best cluster).

Since, the GA parameters directly affect the fitness behavior, before the experiments, we performed a comprehensive analysis of all GA performances, in order to determine its

robustness and adjusting each of its parameters. Finally, we experimentally used the parameters discussed in Table 1 and analyzed the behavior of the algorithm. We show in Figure 7 the average number of hits returned by the GA for samples of 20, 80 and 150 documents, changing the mutation rate, and show the hit factor of the GA against the mutation rate. We appreciate that we got the best performance with a rate of 0.03, this result shows that the best medium fitness could also be obtained by using this rate. We corroborated that conduct with another collection.

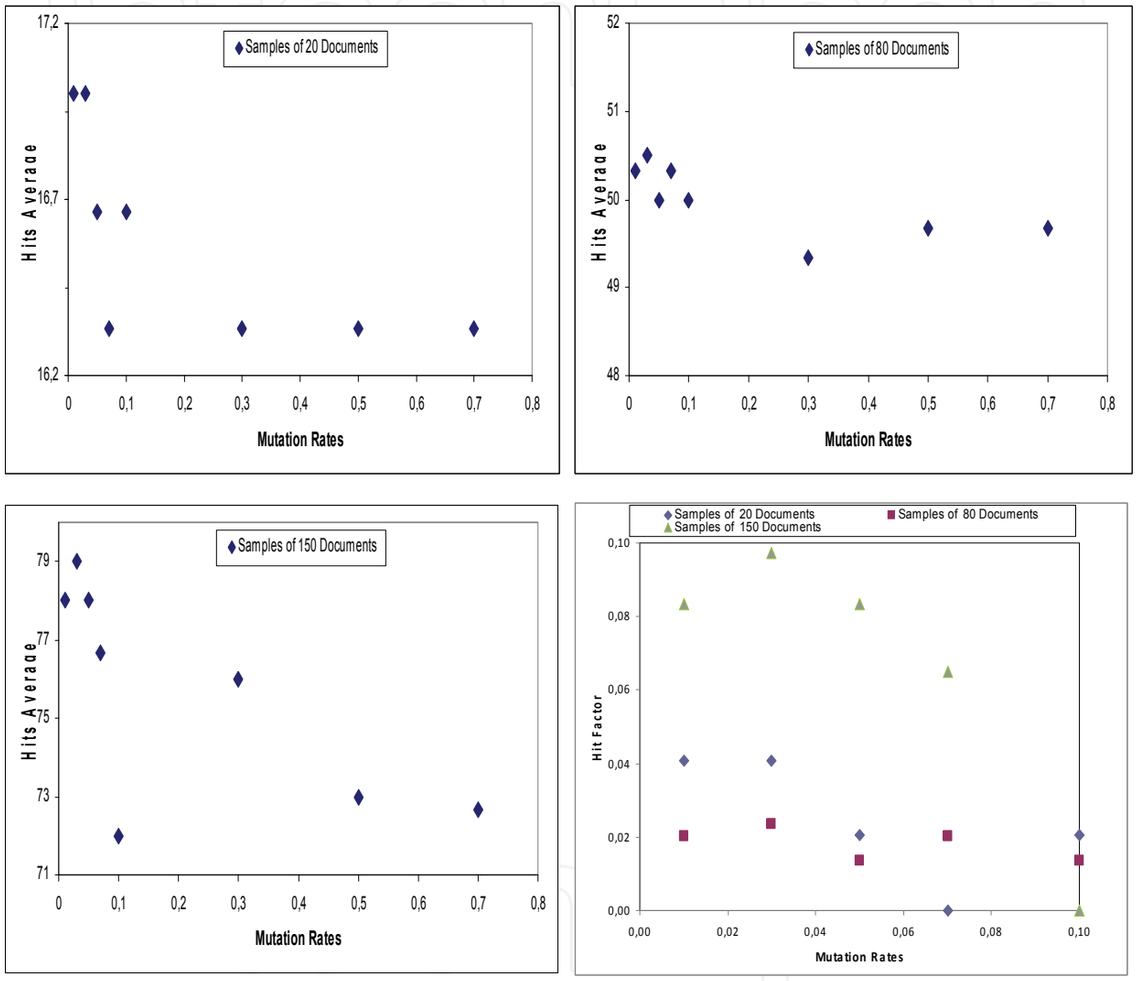


Fig. 7. Hits average of GA with samples 20, 80 and 150 documents varying mutation rate and hit the GA.

In addition, we analyzed the incidence of crossover operator on the final results. The figures 8 show the behavior of the crossover rate versus hits average with very few samples (20), many (80) and many documents (150) respectively. Besides a comparative analysis is the success factor of GA varying the crossover rate. It makes clear, the GA performed better when using a rate of 0.80 for the crossover operator, regardless of the sample. Therefore, this value appears to be ideal if we maximize the efficiency of the algorithm, which is why we conclude that is the rate that gives us better results.

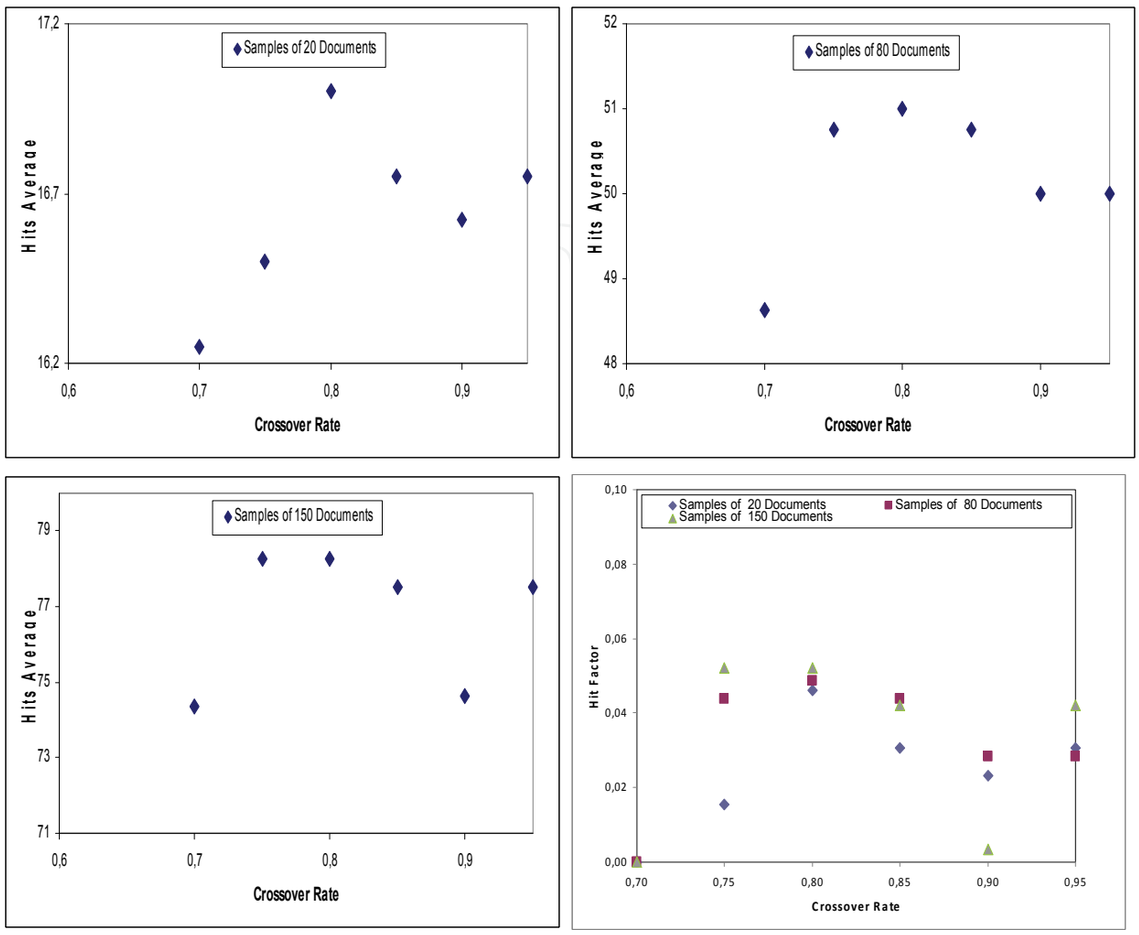


Fig. 8. Hits average of GA with samples 20, 80 and 150 documents varying crossover rate and hit the GA.

To corroborate the results of the GA, we compare their results with the *Kmeans* algorithm, which was processed with *the same samples*, passing as input the number of groups that needed to be obtained. This algorithm used exclusively as a function of the Euclidean distance measure and being a supervised algorithm, the only adjustment of parameters was the *number of groups to process*, and is therefore executed on *Kmeans* in optimal conditions. We proved that *the medium effectiveness* of the GA is very acceptable, being in most cases better than *Kmeans* supervised algorithm [10] when using these parameters of mutation and crossover, but with the added advantage that we processed the documents in an unsupervised way, allowing evolution perform clustering with our adjustment. So, details of such behavior, we show graphically in figure 7 and 8, even showing a comparison of the same for each type of operator used in our experiments the evolutionary algorithm processed proposed for this work.

Then, in the table 4, 5, 6 and 7 show comparative results obtained with our algorithm using the optimal parameters of mutation and crossover with major documentary collection distribution Reuters 21578.

Distribution 2 Reuters Collection 1	*Documents Categories: Acq y Earn Best Result*			*Best Average*			
Samples of documents	*Fitness*	*Effectiveness*	*Convergence*	*Average Fitness*	*Deviation Fitness*	*Average Convergence*	*Kmeans*
Very Few documents (20 documents)	0,155447570	*85% (18 hits)*	886	0,15545476	0,00000828	*1086*	*16,6*
Few Documents (50 documents)	0,156223280	*94% (47 hits)*	*3051*	0,15624280	0,00002329	*2641*	*45,8*
Many Documents (80 documents)	0,159009400	*89% (71 hits)*	*2500*	0,15921181	0,00020587	*2246*	*67,8*
Enough Documents (150 documents)	0,165013920	*77% (115 hits)*	*2342*	0,16508519	0,00007452	*2480*	*121,6*
More Documents (246 documents)	0,174112100	*69% (170 hits)*	*2203*	0,17430502	0,00033602	*2059*	*202,8*

Table 4. Comparative results Evolutionary System with various samples of documents showing the best results and the average results of evaluations with the "Distribution 2" of the Reuters 21578 collection.

Distribution 8 Reuters Collection 2	*Documents Categories: Acq y Earn Best Result*			*Best Average*			
Samples of documents	*Fitness*	*Effectiveness*	*Convergence*	*Average Fitness*	*Deviation Fitness*	*Average Convergence*	*Kmeans*
Very Few documents (20 documents)	0,151163560	*85% (17 hits)*	555	0,15116356	0,00000000	*679*	*15,8*
Few Documents (50 documents)	0,154856500	*96% (48 hits)*	*1615*	0,15485650	0,00000000	*1334*	*43,8*
Many Documents (80 documents)	0,157073880	*85% (68 hits)*	*746*	0,15708362	0,00000898	*1360*	*66,2*
Enough Documents (150 documents)	0,162035070	*69,3% (104 hits)*	*1989*	0,16242664	0,00033091	*2283*	*117,6*
More Documents (188 documents)	0,163014600	*68,63% (129 hits)*	*2293*	0,16334198	0,00027325	*1773*	*140,6*

Table 5. Comparative results Evolutionary System with various samples of documents showing the best results and the average results of evaluations with the "Distribution 8" of the Reuters 21578 collection.

Distribution 20 Reuters Collection 3	*Documents Categories: Acq y Earn Best Result*			*Best Average*			
Samples of documents	*Fitness*	*Effectiveness*	*Convergence*	*Average Fitness*	*Deviation Fitness*	*Average Convergence*	*Kmeans*
Very Few documents (20 documents)	0,153027060	*85% (17 hits)*	1092	0,15321980	0,00018398	*1108*	*16,8*
Few Documents (50 documents)	0,156198620	*92% (46 hits)*	*2173*	0,15666137	0,00030077	*2635*	*44,8*
Many Documents (80 documents)	0,158069980	*81,25% (65 hits)*	*2196*	0,15810383	0,00001884	*1739*	*66,8*
Enough Documents (108 documents)	0,159031080	*69.4% (75 hits)*	*1437*	0,15927630	0,00026701	*2636*	*82,2*

Table 6. Comparative results Evolutionary System with various samples of documents showing the best results and the average results of evaluations with the "Distribution 20" of the Reuters 21578 collection.

Distribution 21 Reuters Collection 4	*Documents Categories: Acq y Earn Best Result*			*Best Average*			
Samples of documents	***Fitness***	***Effectiveness***	***Convergence.***	***Average Fitness***	***Deviation Fitness***	***Average Convergence***	***Kmeans***
Very Few documents (20 documents)	0,152048900	*90% (18 hits)*	1163	0,15206069	0,00001601	*1165*	*17,8*
Few Documents (50 documents)	0,153006650	*92% (46 hits)*	*2079*	0,15304887	0,00004569	*2736*	*45,6*
Many Documents (80 documents)	0,156029510	*81% (65 hits)*	*2787*	0,15637693	0,00025014	*2810*	*66,4*
Enough Documents (132 documents)	0,157012180	*70,4% (93 hits)*	*3359*	0,15720766	0,00024132	*1980*	*98,6*

Table 7. Comparative results Evolutionary System with various samples of documents showing the best results and the average results of evaluations with the "Distribution 21" of the Reuters 21578 collection

To then display the results graphically in figure 9.

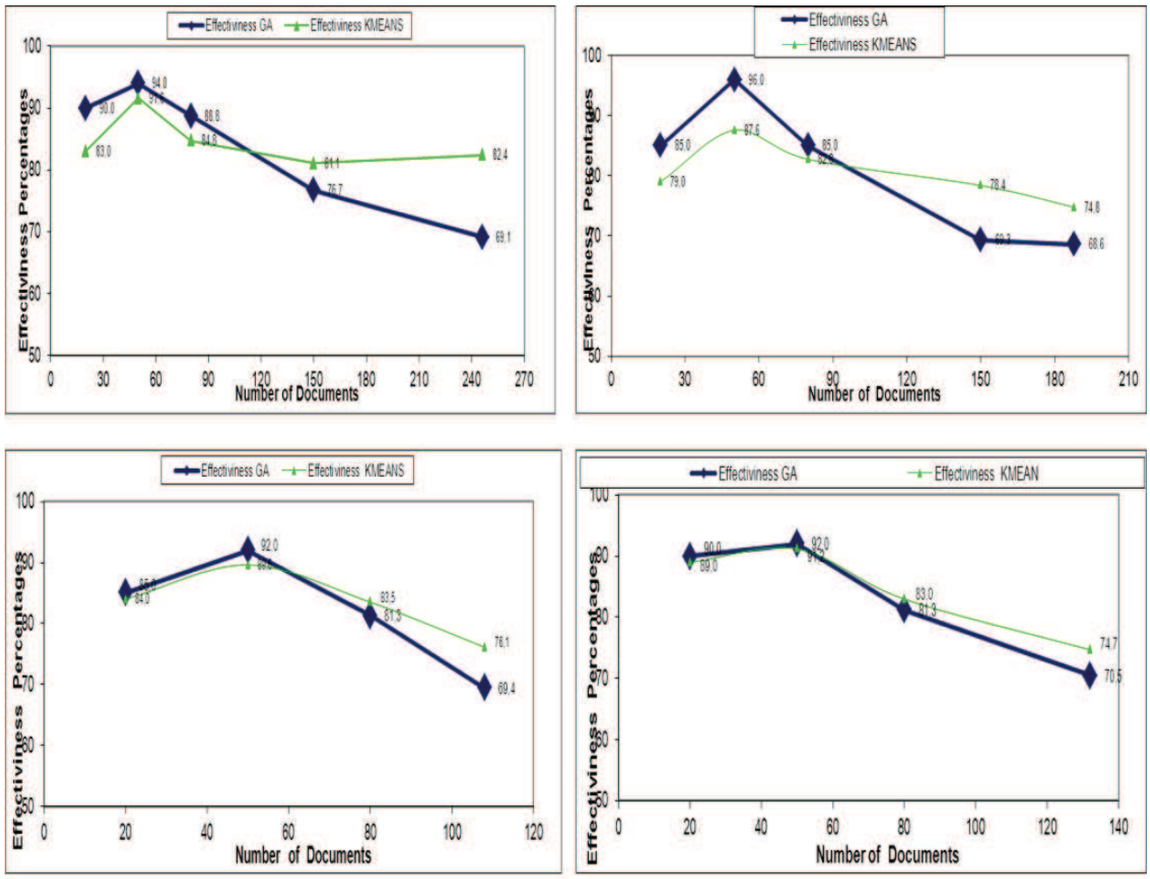


Fig. 9. Graphs compare the results obtained with the composite function against Kmeans (four collection Reuters)

Finally, to corroborate the results,we compare their results with the other collection in Spanish, which was processed in the same way, using all values of table 2. (see figure 10).

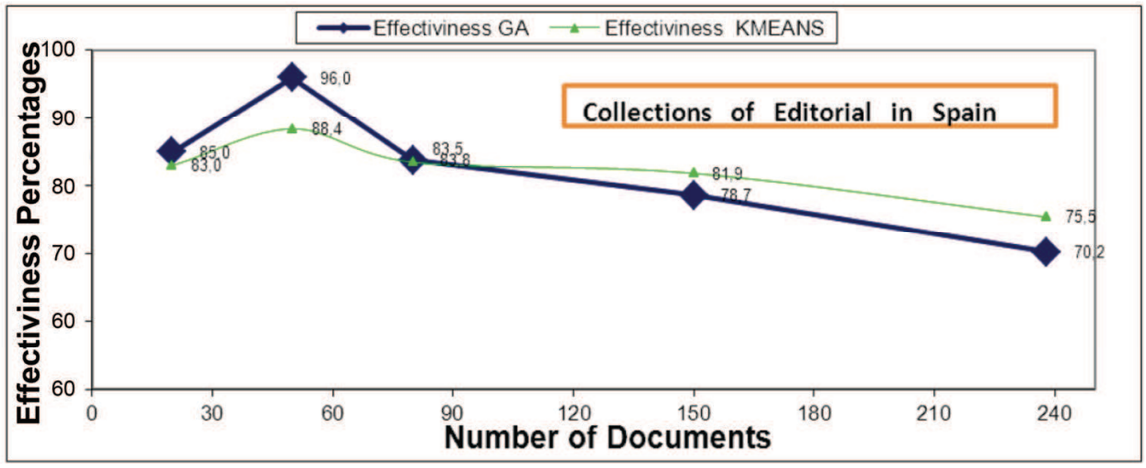


Fig. 10. Graphs compare the results obtained with the composite function against Kmeans (Spain collection)

5. Conclusion

In this study, we have proposed a new taxonomy of parameters of GA numerical and structural, and examine the effects of numerical parameters of the performance of the algorithm in GA based simulation optimization application by the use of a test clustering problem. We start with the characteristics of the problem domain.

The main characteristic features of our problem domain are:

- There is a dominance of a set of decision variables with respect to the objective function value of the optimization problem: The objective function value is directly related with the combination of this dominant set of variables equal a value of α close to 0.85.
- The good solutions are highly dominant over other solutions with respect to the objective function value, but not significantly diverse among each other.

These properties of the problem domain generate a rapid convergent behavior of GA. According to our computational results lower mutation rates give better performance. GA mechanism creates a lock-n effect in the search space, hence lower mutation rates decreases the risk of premature convergence and provides diversification in the search space in this particular problem domain. Due to the dominance crossover operator does not have significant impact on the performance of GA. Moreover, starting with a seeded population generates more efficient results.

We can conclude that the GA had a favourable evolution, offering optimal document cluster in an acceptable and robust manner, based on a proper adjust of the parameters. We proved that *the medium effectiveness* of the GA is very acceptable, being in most cases better than Kmeans supervised algorithm, but with the added advantage that we processed the documents in an unsupervised way, allowing evolution perform clustering with our adjustment. As a result of our experiments, we appreciate that we got the best performance with a rate of 0.03 for the mutation operator and using a rate of 0.80 for the crossover operator, this values appears to be ideal if we maximize the efficiency of the genetic algorithm.

As a future research direction, the same analyses can be carried out for different problem domains, and with different structural parameter settings, and even the interaction between the numerical and structural parameters could be investigated.

6. References

[1] [Alander, 1992] Alander. J"*On optimal populations size of genetic algorithms*" Proc CompEuro 1992.

[2] [Bäck, 1996] Bäck T, "*Evolutionary Algorithms in theory and Practice*", Oxford University Press, 1996.

[3] [Berry Michael, 2004] M. Berry , Survey of Text Mining – Clustering and Retrieval, Springer 2004.

[4] [Berry Michael, et al,2008] M. Berry, Malu Castellano Editors:"*Survey of Text Mining II*", Springer,2008.

[5] [Castillo,Fernandéz,León,2008] "*Information Retrieval with Cluter Genetic*" IADIS Data Mining, 2008.

[6] [Castillo,Fernandéz,León,2009] "*Feature Reduction for Clustering with NZIPF*" IADIS e-Society 2009.

[7] [Goldberd D,1989] *Genetic algorithms in search, optimization and machine learning*. Addison Wesley M.A. 1989.

[8] [Holland J.H, 1975] *Adaptation in Natural and Artificial Systems* University of Michigan Press, Ann Arbor 1975.

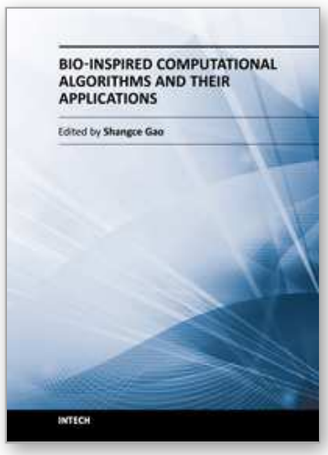
[9] [Michalewicz, 1999] Michalewicz Z."*Genetic Algorithms + Data Structures = Evolution*". Springer-1999.

[10] [Olson David, 2008] Olson D. "*Advanced Data Mining Techniques*", Springer 2008 ISBN:978-3-540-76916-3

[11] [Pao M.L, 1976] Pao M."*Automatic indexing based on Goffman transition of word occurrences*". In American society for Information Science. Meeting (40th: 1977:ChicagoII). Information Management in the 1980's: proceedings of the ASIS annual meeting 1977, volume 14:40th annual meeting, Chicago.

[12] [Reeves CR, 1993] *Modern Heuristic Techniques for Combitational* Problems, Wiley, New York, 1993.

[13] [Schaffer et al,1989] Shaffer, et al, "*A study of control parameters performance on GA for function optimization", 1989.*



Bio-Inspired Computational Algorithms and Their Applications

Edited by Dr. Shangce Gao

ISBN 978-953-51-0214-4
Hard cover, 420 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

Bio-inspired computational algorithms are always hot research topics in artificial intelligence communities. Biology is a bewildering source of inspiration for the design of intelligent artifacts that are capable of efficient and autonomous operation in unknown and changing environments. It is difficult to resist the fascination of creating artifacts that display elements of lifelike intelligence, thus needing techniques for control, optimization, prediction, security, design, and so on. Bio-Inspired Computational Algorithms and Their Applications is a compendium that addresses this need. It integrates contrasting techniques of genetic algorithms, artificial immune systems, particle swarm optimization, and hybrid models to solve many real-world problems. The works presented in this book give insights into the creation of innovative improvements over algorithm performance, potential applications on various practical tasks, and combination of different techniques. The book provides a reference to researchers, practitioners, and students in both artificial intelligence and engineering communities, forming a foundation for the development of the field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen