



Genetic Algorithm for Document Clustering with Simultaneous and Ranked Mutation

K. Premalatha (Corresponding Author)

Kongu Engineering College

Perundurai, Erode, TN, India

E-mail: kpl_barath@yahoo.co.in

A.M. Natarajan

Bannari Amman Institute of Technology

Coimbatore, TN, India

Abstract

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. The clustering algorithm attempts to find natural groups of components, based on some similarity. Traditional clustering algorithms will search only a small sub-set of all possible clustering and consequently, there is no guarantee that the solution found will be optimal. This paper presents the document clustering based on Genetic algorithm with Simultaneous mutation operator and Ranked mutation rate. The mutation operation is significant to the success of genetic algorithms since it expands the search directions and avoids convergence to local optima. In each stage of the genetic process in a problem, may involve aptly different mutation operators for best results. In simultaneous mutation the genetic algorithm concurrently uses several mutation operators in producing the next generation. The mutation ratio of each operator changes according to assessment from the respective offspring it produces. In ranked scheme, it adapts the mutation rate on the chromosome based on the fitness rank of the earlier population. Experiments results are examined with document corpus. It demonstrates that the proposed algorithm statistically outperforms the Simple GA and K-Means.

Keywords: Genetic algorithm, Clustering, Tf-idf, Convergence, K-means, Simultaneous and Ranked mutation

1. Introduction

Document clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. Unlike document classification (Wang, K., Zhou, S., & He Y. 2001), no labeled documents are provided in clustering; hence, clustering is also known as unsupervised learning. Document clustering is widely applicable in areas such as search engines, web mining, Information retrieval and topological analysis. Many clustering techniques have been developed and they can be applied to clustering documents. (Jain A.K., M.N. Murthy, P.J. Flynn. 1999) contain examples of using such techniques. A Genetic Algorithm (GA) is a computational abstraction of biological evolution that can be used to solve some optimization problems (John Holland. 1975.) (Goldberg). In this paper, the documents are clustered based on Genetic algorithm with dynamic mutation operator and adaptive mutation rate for faster convergence. The proposed system is an iterative process applying a series of genetic operators such as selection, crossover and mutation to a population of elements. These elements, called chromosomes or individuals represent possible solutions to the problem; the initial chromosomes are selected randomly from the solution space. Genetic operators combine the genetic information of the elements to form new generations of the population; this process is known as reproduction and mutation. Each chromosome has an associated fitness value which quantifies its value as a solution to the problem- a chromosome representing a better solution will have a higher fitness value. The chromosomes compete to reproduce based on their fitness values, thus the chromosomes representing better solutions have a higher chance of survival

2. Review of Related works

While a lot of work has focused on clustering of numeric data only a limited number of studies have focused on categorical clustering; these include STIRR (Gibson D, J. Kleinberg, and P. Raghavan. 1998.), ROCK (Guha S, R.

Rastogi, K. Shim. 1999.) CACTUS (Ganti V, J. Gehrke, and R. Ramakrishnan. 1999.) COOLCAT (Barbara D, Y. Li, and J. Couto. 2002.), kmodes (Huang Z. 1997.) and others more (Cristofor D, D. Simovici. 2002.)(Zhang. Y, A. Fu, C. Cai, P. Heng. 2000.) Other works have focused more narrowly on binary or transactional data (Ordonez C. 2003.)(Wang, K., Zhou, S., & He Y. 2001.) , on a framework to compress high dimensional categorical datasets (KoyutÄurk M , A. Grama. 2003.) and on using hypergraph partitioning to cluster itemsets (Han E, G. Karypis, V. Kumar, B. Mobasher. 1997.). For document clustering the recent studies have shown that partitioning clustering algorithms are more suitable for clustering large datasets due to their relatively low computational requirements (Makagonov, P., Alexandrov, M., Gelbukh, A. 2002.)(Zhao Y, Karypis G. 2004.). In the field of clustering, K-means algorithm is the most popularly used algorithm to find a partition that minimizes mean square error (MSE) measure. Although K-means is an extensively useful clustering algorithm, it suffers from several drawbacks. The objective function of the K-means is not convex and hence it may contain local minima. Consequently, while minimizing the objective function, there is possibility of getting stuck at local minima (also at local maxima and saddle point) (Selim SZ, Ismail MA. 1984.). The performance of the K-means algorithm depends on the initial choice of the cluster centers. Besides, the Euclidean norm is sensitive to noise or outliers. Hence K-means algorithm should be affected by noise and outliers (Wu KL, Yang MS. 2002.)(Jones G, Robertson A, Santimetvirul C, Willett P. 1995.).

There are earlier works that apply GA and evolutionary programming to clustering. Some of them deal with clustering a set of objects by assuming that the appropriate value of k is known (Goldberg.)(Chu S.C., Roddick J.F., Pan J.S. 2002.) (Murthy C.A., Chowdhury N. 1996.)(Merz P., Zell A. 2002.)(Lucasius C.B., Dane A.D., Kateman G. 1993.). However, in [23] an evolutionary programming-based clustering algorithm is proposed that groups a set of data into an optimum number of clusters. It is based on the well known K-means algorithm. They use two objective functions that are minimized simultaneously: one gives the optimum number of clusters, whereas the other leads to proper identification of each cluster's centroids. (Casillas, M. T. Gonzalez de Lena, and R. Martinez) used only one objective function at the same time both aspects of the solution is calculated: an approximation to the optimum. k value, and the best grouping of the objects into these k clusters. In (Makagonov, P., Alexandrov, M., Gelbukh, A. 2002.) discusses other heuristics to split the dendrite in an optimal way without fixing the number of clusters.

In this work, a method is proposed for the clustering a set of documents using Genetic algorithm with simultaneous mutation operator with ranked mutation rate. The main aim is to provide the best grouping of the objects into k clusters. The remainder of the paper is organized as follow: Section 3 presents the Simple Genetic Algorithm. Section 4 describes the proposed genetic algorithm with simultaneous and ranked mutation for document clustering. Section 5 describes the experiments and their results.

3. Simple Genetic Algorithm

Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination and mutation operators to these structures so as to preserve critical information. An implementation of a genetic algorithm begins with a population of (usually random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions. The goodness of a solution is typically defined with respect to the current population. Usually there are only two main components of genetic algorithms that are problem dependent: the problem encoding and the fitness function (objective function / evaluation function). A problem can be viewed as a black box with different parameters: The only output of the black box is a value returned by an evaluation function indicating how well a particular combination of parameter settings solves the optimization problem. The goal is to set the various parameters so as to optimize some output. In more traditional terms that to maximize (or minimize) some function $F(X_1, X_2, \dots, X_m)$

The first assumption that is typically made is that the variables representing parameters can be represented by strings and the evaluation function is usually given as part of the problem description. The genetic algorithm can be viewed a two stage process. It starts with the current population. Selection is applied to the current population to create an intermediate population. Then recombination and mutation are applied to the intermediate population to create the next population. The process of going from the current population to the next population constitutes one generation in the execution of a genetic algorithm. (Goldberg) refers to this basic implementation as a Simple Genetic Algorithm (SGA)

In the first generation the current population is also the initial population. There are a number of ways to do selection. After selection has been carried out the construction of the intermediate population is complete and recombination can occur. This can be viewed as creating the next population from the intermediate population. Crossover is applied to randomly paired strings with a probability denoted P_c . (The population should already be sufficiently shuffled by the random selection process) Pick a pair of strings with probability P_c recombine these strings to form two new strings that are inserted into the next population. After recombination, mutation operator is applied. For each bit in the population, mutate with some low probability P_m . Typically the mutation rate is applied with less than 1% probability.

In some cases mutation is interpreted as randomly generating a new bit in which case, only 50% of the time will the mutation actually change the bit value. After the process of selection, recombination and mutation, the next population can be evaluated. The process of evaluation, selection, recombination and mutation forms one generation in the execution of a genetic algorithm.

Algorithm

- 1) Generate an initial, random population of chromosomes.
- 2) Test the fitness of each chromosome in the population.
- 3) Select parents as the most fit members of the population.
- 4) Reproduce from selected parents to produce a new population.
- 5) Mutate according to some probability.
- 6) Test the fitness of each chromosome in the new population.
- 7) Evaluation
- 8) Iterate steps 3 to 7 until termination criterion is met.

To successfully apply a GA to solve a problem one needs to determine the following:

- 1) How to represent possible solutions, or the chromosomal encoding;
- 2) What to use as the fitness function which accurately represents the value of the solution;
- 3) Which genetic operators to employ; and
- 4) The parameter values (population size, probability of applying operators, etc.) which are suitable.

4. GA With Simultaneous and Ranked Mutation

The algorithm begins with the initial solutions population of the problem. This population is generated randomly. Each one of these solutions must be evaluated by means of a fitness function; the result of this evaluation is a measure of individual adaptation. The individuals with the best adaptation measure have more chances of reproducing and generating new individuals.

4.1 Problem Formulation

The objective of optimization is to seek value for set of parameters that maximize or minimize objective functions subject to certain constraints. A choice of values for the set of parameters that satisfy all constraints is called a feasible solution. Feasible solutions with objective function value as good as the values of any other feasible solutions are called optimal solutions.

The objective function of the document clustering problem is given as follows:

$$f = \frac{\sum_{i=1}^{N_c} \frac{\sum_{j=1}^{P_i} m_{ij} \cdot O_j}{\|m_{ij}\| \cdot \|O_j\|}}{N_c} \quad (1)$$

The function f should be maximized.

- where
- m_{ij} : j th document vector belongs to cluster i
 - O_i : Centroid vector of the i^{th} cluster
 - P_i : stands for the number of documents, which belongs to cluster C_i ;
 - N_c : number of clusters.

4.2 Document Vectorization

It is necessary to convert the document collection into the form of document vectors. Firstly, to determine the terms that is used to describe the documents. Extraction of all the words from each document.

- 1) Elimination of the stopwords from a stopword list generated with the frequency dictionary of (Kucera, H., & Francis, N. 1967.)
- 2) Stemming the remaining words using the Porter Stemmer which is the most commonly used stemmer in English
- 3) Formalizing the document as a dot in the multidimensional space and represented by a vector d , such as $d = \{w_1, w_2, \dots, w_n\}$, where w_i ($i = 1, 2, \dots, n$) is the term weight of the term t_i in one document. The term weight value represents the significance of this term in a document. To calculate the term weight, the occurrence frequency of the term within a

document and in the entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF). The weight of term i in document j is given in equation (2)

$$W_{ji} = tf_{ji} \times idf_{ji} = tf_{ji} \times \log_2 (n/df_{ji}) \quad (2)$$

where tf_{ji} is the number of occurrences of term i in the document j ; df_{ji} indicates the term frequency in the collections of documents; and n is the total number of documents in the collection.

4.3 Chromosome Representation

The algorithm uses chromosomes which codify the whole partition P of the data set in a vector of length n , where n is the size of the dataset. Thus, each gene of the chromosome is the label where the single item of the dataset belongs to; in particular if the number of cluster is k each gene of the chromosome is an integer value in the range $\{1, \dots, K\}$. An example of chromosome is reported in Figure 2.

4.4 Initial Generation

At the initial stage, each individual randomly chooses k different document vectors from the document collection as the initial cluster centroid vectors. For, each individual, a gene assigns a document vector from the document collection to the closest centroid cluster. The allele of gene represents the cluster where the document is present. The objective function for each individual can be calculated based on the equation (1).

4.5 Selection

In selection the offspring producing individuals are chosen. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own objective value and the objective value of all other individuals in the selection pool. This fitness is used for the actual selection in the step afterwards. The simplest selection scheme is roulette-wheel selection, also called stochastic sampling with replacement. The proposed system employs roulette-wheel selection method.

4.6 Crossover

The interesting behavior happens from genetic algorithms because of the ability of the solutions to learn from each other. Solutions can combine to form offspring for the next generation. Occasionally they will pass on their worst information, but doing crossover in combination with a powerful selection technique perceives better solutions result. Crossover occurs with a user specified probability called, the crossover probability P_c . Many crossover techniques exist for individual. In single point crossover, a position is randomly selected at which the parents are divided into two parts. The parts of the two parents are then swapped to generate two new offspring.

4.7 Simultaneous and Ranked Mutation

The purpose of mutation is to diversify the search direction and prevent convergence to the local optimum. Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search as help to prevent the population from stagnating at any local optima. It prevents local searches of the search space and increases the probability of finding global optima. Mutation occurs during evolution according to a user-definable mutation probability P_m

4.7.1 Simultaneous Mutation Operator

The proposed algorithm at once uses several mutation operators in producing the next generation. The mutation ratio of each operator changes according to evaluation results from the respective offspring it produces (Tzung-Pei Hong, Hong-Shung Wang, Wei-Chou Chen. 2000.). Thus, the appropriate mutation operators can be expected to have increasingly greater effects on the genetic process. It automatically chooses a proper mutation operator, or to handle situations in which different operators are suitable for different genetic stages. Initially, the mutation ratios are the same for all operators. After a new generation or several generations, the mutation ratio of each operator is automatically and dynamically adjusted. The mutation ratios of suitable operators (evaluated according to fitness values) are increased and the mutation ratios of unsuitable operators are decreased. The highly effective mutation operators will have increasingly greater effects on the genetic process.

Assume N mutation operators are applied in the proposed algorithm. Initially, the mutation ratios of all available mutation operators are set equal ($1/N$ of the total mutation ratio for the problem). Each mutation operator is then applied according to its assigned probability, and offspring fitness is evaluated. The mutation operators that result in higher average fitness values then have their control ratios increased. The mutation operators that result in lower average fitness values then have their control ratios decreased. Finally, the most suitable mutation operator stands out and

controls almost all the mutation behavior in the population. The dynamic mutation genetic algorithm is then theoretically better than most genetic algorithms that use only single mutation operators.

Algorithm

Calculate the average growth value $P(M_i)$ $I = 1$ to m (the number of applied mutation operators), given by each mutation operator. Assume a parent p is chosen by a mutation operator to produce a child a . the progress value of the mutation operator for this operation is calculated is given in equation (3)

$$P = \max(f(p), f(a)) - f(p) \quad (3)$$

$f(p)$ – Fitness function selected for Parent

$f(a)$ – Fitness function for a

Assume r parents are chosen by this mutation operator in this generation. The progress value $P(M_i)$ of this mutation operator M_i is then the average of the r values. The Mutation ratio is calculated as follows (4) :

$$M_r = \frac{P(M_i)}{\sum P} \times N \quad (4)$$

M_r = Mutation Ratio

N – Number of Parents selected for Mutation

4.7.2 Ranked Mutation Rate

In this study, different mutation rates are assigned to each chromosome based on the fitness rank of the earlier population.

$$P_m = \left(\text{Max} - \frac{\text{Max} - \text{Min}}{\text{NOI}} \times T \right) \times \frac{\text{rank}}{R}$$

Max = Maximum Mutation rate

Min = Minimum Mutation rate

NOI = Number of Iterations

T = Time epoch (or) Iteration number

R = Total number of Ranks

4.8 Evaluation

After producing offspring they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original population. Another case is, when not all offspring are to be used at each generation or if more offspring are generated than needed. By a reinsertion scheme is determined which individuals should be inserted into the new population and which individuals of the population will be replaced by offspring. The used selection algorithm determines the reinsertion scheme. The elitist combined with fitness-based reinsertion prevents this losing of information and is the recommended method. At each generation, a given number of the least fit parent is replaced by the same number of the most fit offspring. Fig 3 shows the proposed system.

5. Experiment Results

The proposed system experimented on some common datasets CISI, Cranfield and ADI available at the Glasocow [http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/]. The number of individuals selected for each generation is 20 for 3 clusters. Random and Boundary mutation operators are preferred for simultaneous mutation. In K-Means algorithm the clusters reformed such that the sum of square errors to be minimal. The main goal of the system is to maximize the fitness value and fastening the convergence to near global optimal solution. Figure 4 shows the fitness value obtained from GA with random mutation operator, GA with boundary mutation operator, GA with simultaneous random and boundary mutation operators with ranked mutation rate and K-Means algorithm.

CONCLUSION

The objective of this paper is to present the potential power of the mutation operator in Genetic algorithm to solve the combinatorial optimization problems, which has a variety of applications in the real world. The proposed technique markedly increased the success of the document clustering problem. Two models were used in comparison, namely the Simple GA and K-Means. The proposed model was outperformed by the Simple GA and K-Means regarding a comparison of the best optima found. Yet, the proposed system had a marginally faster convergence than Simple GA and K-Means due to simultaneous and ranked nature of mutation operator.

References

- Barbara D, Y. Li, and J. Couto. (2002). Coolcat: an entropy-based algorithm for categorical clustering. In ACM Press, pages 582-589.
- Casillas, M. T. Gonzalez de Lena, and R. Martinez, Document Clustering into an unknown number of clusters using a Genetic Algorithm.
- Chu S.C., Roddick J.F., Pan J.S.: "An Incremental Multi-Centroid, Multi-Run Sampling Scheme for k-medoids-based Algorithms-Extended Report". Proceedings of the Third International Conference on Data Mining Methods and Databases, Data Mining III, (2002), 553–562.
- Cristofor D, D. Simovici.(2002). An information-theoretical approach to clustering categorical databases using genetic algorithms. In *2nd SIAM ICDM, Workshop on clustering high dimensional data*.
- Estivill-Castro V., Murray A.T.: "Spatial Clustering for Data Mining with Genetic Algorithms". *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems*, EIS-98, (1998).
- Ganti V, J. Gehrke, and R. Ramakrishnan. CACTUS: Clustering categorical data using summaries. In ACM SIGKDD Int'l Conference on Knowledge discovery in Databases, 1999.
- Gibson D, J. Kleinberg, and P. Raghavan. (1998). Clustering categorical data: An approach based on dynamical systems. In 24th Int'l Conference on Very Large Databases.
- Goldberg. D Genetic Algorithms in Search Optimization and Machine Learning, Reading, MA: Addison Wesley.
- Guha S, R. Rastogi, K. Shim.(1999). Rock: A robust clustering algorithm for categorical attributes. In International Conference on Data Engineering.
- Han E, G. Karypis, V. Kumar, B. Mobasher.(1997). Clustering based on association rule hypergraphs. In *Research Issues on Data Mining and Knowledge Discovery*.
- Holland. J, Adaptation In Natural and Artificial Systems, University of Michigan Press, 1975.
- Huang Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Tucson, AZ, 1997.
- Jain A.K., M.N. Murthy, P.J. Flynn. (1999). Data Clustering : A Review ACM Computing Surveys, 31(3): 264-323.
- John Holland.Adaption in Natural and Artificial Systems. University of Michigan Press, 1975.
- Jones G, Robertson A, Santimetrovirul C, Willett P. (1995). Non-hierarchic document clustering using a genetic algorithm. *Information Research*, 1(1).
- KoyutÄurk M , A. Grama. (2003). Proximus: A framework for analyzing very high-dimensional discrete attributed datasets. In *Proceedings of the Ninth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD 2003)*, pages 147 - 156.
- Kucera, H., & Francis, N. (1967). Computational analysis of present-day American English. Providence, RD: Brown University Press.
- Lucasius C.B., Dane A.D., Kateman G.: "On k-medoid clustering of large data sets with the aid of Genetic Algorithm: background, feasibility and comparison". *Analytica Chimica Acta*, Elsevier Science Publishers B.V. 283(3), (1993) 647–669.
- Makagonov, P., Alexandrov, M., Gelbukh, A.: "Selection of typical documents in a document flow". *Advances in Communications and Software Technologies*, WSEAS Press (2002). 197–202.
- Merz P., Zell A.: "Clustering Gene Expression Profiles with Memetic Algorithms". *Lecture Notes in Computer Science* 2439, Springer-Verlag Berlin (2002). 811–820.
- Murthy C.A., Chowdhury N.: "In search of Optimal Clusters Using Genetic Algorithms". *Pattern Recognition Letters*, 17(8), (1996), 825–832.
- Ordonez C. (2003). Clustering binary data streams with k-means. In *Proceedings of DMKD*, pages 12 -19.
- Sarkar, M., Yegnanarayana, B., Khemani, D.: "A clustering algorithm using an evolutionary programming-based approach". *Pattern Recognition Letters*, 18, (1997). 975–986.
- Selim SZ, Ismail MA. (1984). K-means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6, 81-87.
- Tzung-Pei Hong, Hong-Shung Wang, Wei-Chou Chen, Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms, *Journal of Heuristics*, 6: 439–455, (2000).

Wang K , C. Xu, B. Liu. (1999). Clustering transactions using large items. In *CIKM*, pages 483 -490.

Wang, K., Zhou, S., & He Y. (2001, Apr.). Hierarchical classification of real life documents. *SIAM International Conference on Data Mining, SDM'01*, Chicago, United States.

Wu KL, Yang MS. (2002). Alternative C-means Clustering Algorithms. *PatternRecognition*, 35, 2267-2278.

Zhang. Y, A. Fu, C. Cai, P. Heng. (2000). Clustering categorical data. In *Proceedings of the ICDE*, page 305.

Zhao Y, Karypis G (2004). Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, *Machine Learning*, 55(3), 311-331.

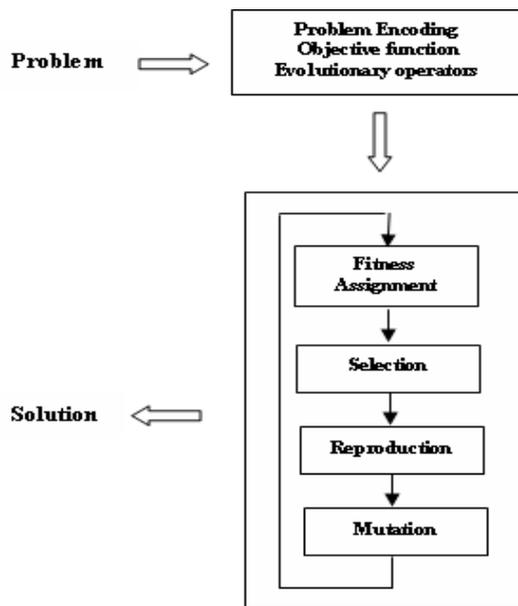


Figure 1. Simple Genetic algorithm

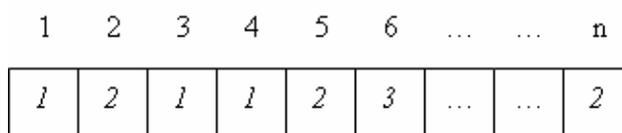


Figure 2. Chromosome representation

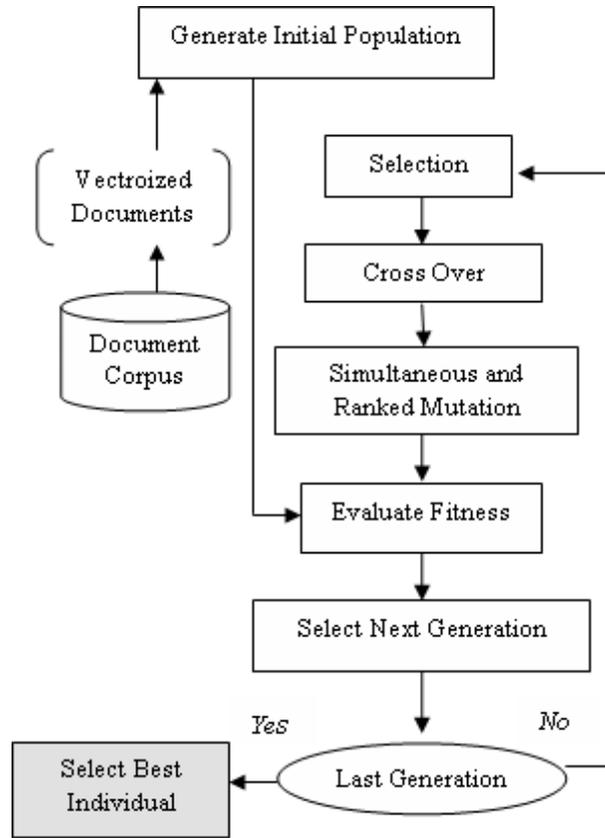


Figure 3. GA Simultaneous and Ranked Mutation for Document clustering

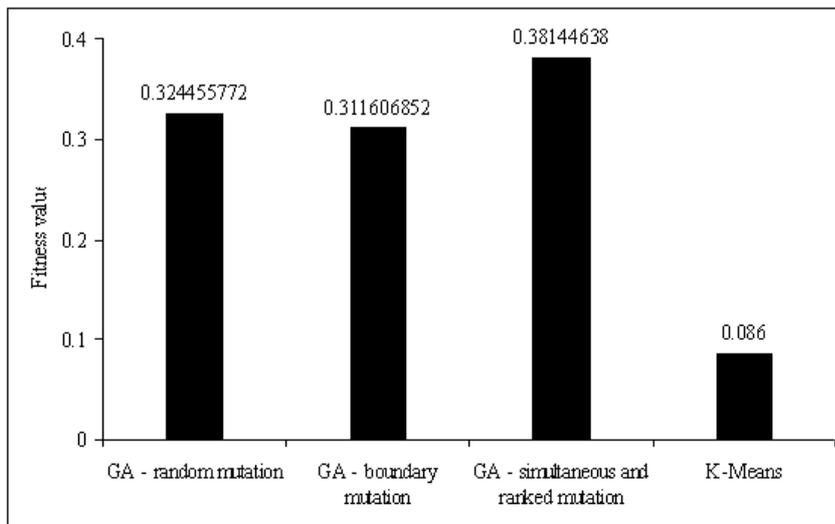


Figure 4. Fitness value obtained from various systems