IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-10 E-ISSN: 2347-2693

A Study on Genetic Algorithm and its Applications

L. Haldurai[1*], T. Madhubala[2] and R. Rajalakshmi[3]

[123*]Department of Computer Science (PG), Kongunadu Arts and Science College, Coimbatore, India

Available online at: www.ijcseonline.org

Received:19/Sep/2016 Revised: 28/Sep/2016 Accepted:21/Oct/2016 Published: 31/Oct/2016

Abstract— **In order to obtain best solutions, we need a measure for differentiating best solutions from worst solutions. The measure could be an objective one that is a statistical model or a simulation, or it can be a subjective one where we choose better solutions over worst ones. Apart from this the fitness function determines a best solution for a given problem, which is subsequently used by the GA to guide the evolution of best solutions. This paper shows how GA is combined with various other methods and technique to derive optimal solution, increase the computation time of retrieval system the applications of genetic algorithms in various fields.**

Keywords- Genetic Algorithm; Optimal Solution; Fitness function

I. INTRODUCTION

Genetic algorithms [1] are search and optimization algorithms based on the principles of natural evolution, which were first introduced by john Holland in 1970. Genetic algorithms also implement the optimization strategies by simulating evolution of species through natural selections. Genetic algorithm is generally composed of two processes. First process is selection of individual for the production of next generation and second process is manipulation of the selected individual to form the next generation by crossover and mutation techniques [2]. The selection mechanism determines which individual are chosen for reproduction and how many offspring each selected individual produce. The main principle of selection strategy is the better is an individual; the higher is its chance of being parent.

II. GENETIC ALGORITHM

Genetic algorithms (GA) are search algorithms based on the principles of natural selection and genetics, introduced by J Holland in the 1970's and inspired by the biological evolution of living beings. Genetic algorithms abstract the problem space as a population of individuals, and try to explore the fittest individual by producing generations iteratively. GA evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment. The procedure starts from an initial population of randomly generated individuals.

During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, i.e. selection, crossover and mutation [3].

*Corresponding Author: L. Haldurai
e-mail: haldurai@gmail.com

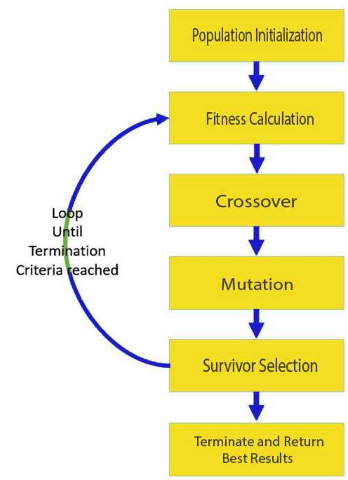

Figure 1. Flowchart of GA System

The GAs is computer program that simulate the heredity and evolution of living organisms [3]. An optimum solution is possible even for multi modal objective functions utilizing GAs because they are multi-point search methods. Also, GAs is applicable to discrete search space problems. Thus, GA is not only very easy to use but also a very powerful optimization tool [4]. In GA, the search space consists of strings, each of which representing a candidate solution to the problem and are termed as chromosomes. The objective function value of each chromosome is called its fitness value. Population is a set of chromosomes along with their associated fitness. Generations are populations generated in an iteration of the GA [5]. Genetic algorithm to search a space of candidate solutions to identify the best one is as follows:

Procedure:
{
1. [***Start***] Generate random population of n chromosomes (suitable solutions for the problem).

2. [***Fitness***] Evaluate the fitness f(x) of each chromosome x in the population.
3. [***New population***] Create a new population by repeating following steps until the new population is complete
a. [***Selection***] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).
b. [***Crossover***] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
c. [***Mutation***] With a mutation probability mutate new offspring at each locus (position in chromosome).
d. [***Accepting***] Place new offspring in a new population.
4. [***Replace***] Use new generated population for a further run of algorithm.
5. [***Test***] If the end condition is satisfied, stop, and return the best solution in current population.
6. [***Loop***] Go to step 2.
}

III. GENETIC OPERATORS

GA searches for better solutions by genetic operations, including *selection* operation, *crossover* operation and *mutation* operation.

A. Selection Operation

Selection operation is to select elitist individuals as parents in current population, which can generate offspring. Fitness values are used as criteria to judge whether individuals are elitist. There are many methods how to select the best chromosomes, for example *roulette wheel* selection, *Boltzman* selection, *tournament* selection, *rank* selection, *steady-state* selection, elitism selection and some others. Some of them will be described shortly.

1) Roulette Wheel Selection

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where are placed all chromosomes in the population, every has its place big accordingly to its fitness function like on the Figure 2. Chromosome with bigger fitness will be selected more times.

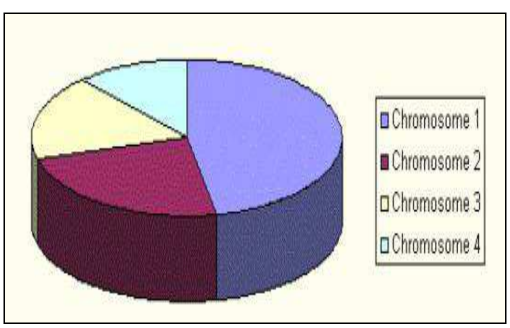

Figure 2. Roulette Wheel Selection

2) Rank Selection

The previous selection method will have problems when the fitness's differ very much. For example, if the best chromosome fitness is 90% of the entire roulette wheel, then the other chromosomes will have very few chances to be selected. Rank selection first sorts the population by fitness and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N (number of chromosomes in population). After this, all the chromosomes have a chance to be selected. The probability that a chromosome will be selected is then proportional to its rank in this sorted list, rather than its fitness. But this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

3) Elitism Selection

When creating new population by crossover and mutation; we have a big chance, that we will lose the best chromosome. Elitism is name of method, which first copies the best chromosome (or a few best chromosomes) to new population. The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution.

B. Crossover Operations

The generation of successors in a GA is determined by a set of operators that recombine and mutate selected members of the current population. The two most common operators are crossover and mutation. The crossover operator produces two new offspring from two parent strings, by copying selected bits from each parent. The bit at position i in each offspring is copied from the bit at position i in one of the two parents. The choice of which parent contributes the bit for position i is determined by an additional string called the crossover mask. Figure 3.10 below illustrates crossover operator briefly. There are three types of crossover operators, namely as single-point, two-point and uniform crossover.

1) Single-Point Crossover

In the single-point crossover, the crossover mask is always constructed so that it begins with a string containing n contiguous 1s, followed by the necessary number of 0s to complete the string. This results in offspring in which the first n bits are contributed by one parent and the remaining bits by the second parent. Each time the single-point crossover operator is applied, the crossover point n is chosen at random, and the crossover mask is then created and applied. To illustrate, consider the single-point crossover operator at the top of the figure and consider the topmost of the two offspring in this case. This offspring takes its first five bits from the first parent and its remaining six bits from the second parent, because the crossover mask 11111000000 specifies these choices for each of the bit positions. The second offspring uses the same crossover mask, but switches the roles of the two parents. Therefore, it contains the bits that were not used by the first offspring.

2) Two-Point Crossover

In two-point crossover, offspring are created by substituting intermediate segments of one parent into the middle of the second parent string. Put another way, the crossover mask is a string beginning with n0 zeros, followed by a contiguous string of nl ones, followed by the necessary number of zeros to complete the string. Each time the two-point crossover operator is applied, a mask is generated by randomly choosing the integers n0 and nl. For instance, in the example shown in Figure 3.10 the offspring are created using a mask for which n0= 2 and nl = 5. Again, the two offspring are created by switching the roles played by the two parents.

3) Uniform Crossover

Uniform crossover combines bits sampled uniformly from the two parents, as illustrated in Figure 3. In this case the crossover mask is generated as a random bit string with each bit chosen at random and independent of the others.

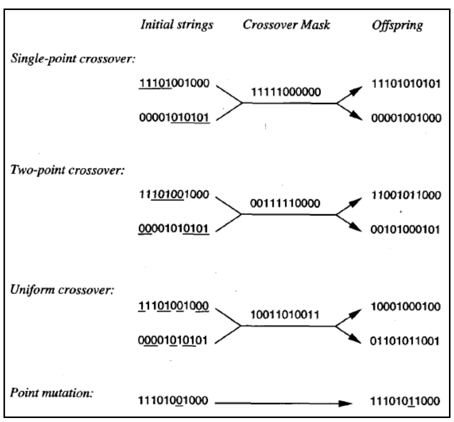

Figure 3. Crossover and Mutation Operations

C. Mutation Operations

In addition to recombination operators that produce offspring by combining parts of two parents, a second type of operator produces offspring from a single parent. In particular, the *mutation* operator produces small random changes to the bit string by choosing a single bit at random, then changing its value. Mutation is often performed after crossover as in Figure 3.

IV. Fitness Function

The fitness function defines the criterion for ranking potential hypotheses and for probabilistically selecting them for inclusion in the next generation population. If the task is to learn classification rules, then the fitness function typically has a component that scores the classification accuracy of the rule over a set of provided training examples. Often other criteria may be included as well, such as the complexity or generality of the rule. More generally, when the bit-string hypothesis is interpreted as a complex procedure (e.g., when the bit string represents a collection of if-then rules that will be chained together to control a robotic device), the fitness function may measure the overall performance of the resulting procedure rather than performance of individual rules.

V. Review of Literature

A novel hybrid genetic k-means algorithm (GKA) [6] to find a globally optimal partition of a given data into a specified number of clusters. The proposed GA circumvent expensive crossover operator used to generate valid child chromosomes from parent chromosomes. It hybridized the GA by using a classical gradient descent algorithm used in clustering viz., K-means algorithm. In genetic K-means algorithm (GKA), K-means operator was defined and used as a search operator instead of crossover. It defined a biased mutation operator specific to clustering called distance-based-mutation. The authors used finite Markov chain theory to prove that the proposed GKA converges to the global optimum. It was also observed that GKA searches faster than some of the other evolutionary algorithms used for clustering.

An improved version of GKA known as Fast Genetic K-means Algorithm (FGKA) was proposed in [7]. The proposed GA featured several improvements over GKA. It was evident from experiments in [6][7] that K-means algorithm might converge to a local optimum, both FGKA and GKA always converge to the global optimum. FGKA initializes the population to P0 and obtains the next population by applying selection, crossover and mutation operators and it keeps on evolving until some termination condition is met. Illegal strings are permitted in FGKA during initialization phase, but were considered as the most undesirable solutions by defining their total within cluster variation (TWCVs) as infinity (+∞). By allowing illegal strings the overhead of illegal string in the evolution process was avoided and thus improved the time performance of the algorithm as compared to GKA.

Incremental Genetic K-means Algorithm (IGKA) proposed in [8] was an extension to previously proposed clustering algorithm, the Fast Genetic K- means Algorithm (FGKA). The performance of IGKA was found to be better when the mutation probability was small. IGKA was based calculating the Total Within-Cluster Variation (TWCV) and to cluster centroids incrementally whenever the mutation probability was small for the clustering task. Like FGKA, IGKA also always converges to the global optimum.

A GA-based unsupervised clustering technique was proposed in [9], which selects cluster centers directly from the data set, thus speeding up the fitness evaluation process by constructing a look-up table in advance and saving the distances between all pairs of data points. Binary representation rather than string representation is used to encode a variable number of cluster centers and more effective operators for selection, crossover, and mutation were introduced.

A novel clustering algorithm for mixed data was proposed in [10].Most of the existing clustering algorithms were only efficient for the numeric data rather than the mixed data set but the proposed GA worked efficiently for datasets with mixed values by modifying the common cost function.

A hybrid genetic based clustering algorithm, called HGA-clustering was proposed in [11] to explore the proper clustering of data sets. This algorithm, with the cooperation of tabulist and aspiration criteria, has achieved harmony between population diversity and convergence speed.

A genetic algorithm was proposed in [12] which designed a dissimilarity measure, termed as Genetic Distance Measure (GDM) to improve the performance of the K-modes algorithm which is an extension of k- means.

A novel approach of [13] parallel indexing the color and feature extraction of images and genetic algorithm has been implemented. Its main functionality is image-to-image matching and its intended use is for still-image retrieval. The evaluation criteria are provided by the GA and have been successfully employed as a measure to evaluate the efficacy of content-based image retrieval process.

In [14], P.R. Srivastava and Tai have presented a method for optimizing software testing efficiency by identifying the most critical path clusters in a program. The SUT is converted into a CFG. Weights are assigned to the edges of the CFG by applying 80-20 rule. 80 percentage of weight of incoming credit is given to loops and branches and the remaining 20 percentage of incoming credit is given to the edges in sequential path. The summation of weights along the edges comprising a path determines criticality of path. Higher the summation more critical is path and therefore must be tested before other paths. In this way by identifying most critical paths that must be tested first, testing efficiency is increased.

In [15], Zhao used the neural network and GA for the functional testing. Neural network is used to create a model that can be taken as a function substitute for the SUT. The emphasis is given on the outputs which exhibit the important features of SUT than inputs. In that case, test cases should be generated from the output domain rather than input domain. The feed forward neural network and back propagation training algorithm is used for creating a model. Neural network is trained by simulating the SUT. The outputs generated from the created model are fed to the GA which is used to find the corresponding inputs so that automation of test cases generation from output domain is completed.

In [16] introduces a new algorithm based on the traditional genetic algorithm, for the traditional GA algorithm the new algorithm has done some improvements: By introducing genetic selection strategy, decreased the possibility of being trapped into a local optimum. Compared the traditional genetic algorithm, the new algorithm enlarges the searching space and the complexity is not high. By analyzing the testing results of benchmarks functions optimization, it is concluded that in the optimization precision, the new algorithm is efficiency than the traditional genetic algorithm. We also use this new algorithm for data classification and the experiment results shown that our proposed algorithm outperforms the KNN with greater accuracy.

Managing groundwater supplies has found AI and GAs useful. [17] Proposed used GAs to fit parameters of a model to optimize pumping locations and schedules for groundwater treatment. They then combined the GA with a neural network (NN) to model the complex response functions within the GA [18]. Then [19] combined Simulated Annealing (SA) and GAs to maximize efficiency and well use the easily applied parallel nature of the GA. Most recently, [20] together with Peralta used a Pareto GA to sort optimal solutions for managing surface and groundwater supplies, together with a fuzzy-penalty function while using an Artificial Neural Network (ANN) to model the complex aquifer systems in the groundwater system responses.

Evolutionary methods have also found their way into oceanographic experimental design. In [21] showed that a genetic algorithm is faster than simulated annealing and more accurate than a problem specific method for optimizing the design of an oceanographic experiment. [22] found that an evolutionary programming strategy was more robust than traditional methods for locating an array of sensors in the ocean after they have drifted from their initial deployment location.

VI. CONCLUSION

Genetic Algorithms proved to be better in finding areas of complex and real world problems. Genetic Algorithms are adaptive to their environments, as this type of method is a platform appearing in the changing environment. In Present these algorithms are more applicable. Several improvements must be made in order that GAs could be more generally applicable.

REFERENCES

[1] J. Holland, "Adaptation in natural and artificial systems", University of Michigan press, Ann Arbor, 1975.

[2] Noraini Mohd Razali, John Geraghty "A genetic algorithm performance with different selection strategies", Proceedings of the World Congress on Engineering Vol II, 2011.

[3] D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[4] Gaidhani, Chaitali R., Vedashree M. Deshpande, and Vrushali N. Bora. "Image Steganography for Message Hiding Using Genetic Algorithm." (2014).

[5] Radwan A., Latef B., Ali A., and Sadek O., "Using Genetic Algorithm to Improve Information Retrieval Systems", World Academy of Science and Engineering Technology, 17, Issue-2, Page No (6-13), 2006.

[6] K. Krishna and M. N. Murty, "Genetic K-Means Algorithm", IEEE Transaction on Systems, Man, and Cybernetics—Part B: CYBERNETICS, Vol. 29, No. 3, June 1999.

[7] Shrivastava, Animesh, and Singh Rajawat. "An implementation of hybrid genetic algorithm for clustering based data for web recommendation system." Int J Comput Sci Eng 2.4 (2014): 6-11.

[8] Yi Lu1, Shiyong Lu1, Farshad Fotouhi1, Youping Deng, d. Susan, J. Brown," an Incremental genetic K- means algorithm and its application in gene expression data analysis", BMC Bioinformatics 2004.

[9] Sangari, Fardin Esmaeeli, and Mehrdad Nabahat. "Efficacy of Different Strategies in Graph Coloring with Parallel Genetic Algorithms." (2014): 138-141.

[10] LI Jie, G. Xinbo, "A GA-Based Clustering Algorithm forLarge Data Sets With Mixed Numeric and Categorical Values",IEEE, Proceedings of the Fifth International Conference onComputational Intelligence and Multimedia Applications(ICCIMA'03) 0-7695-1957-1/03, 2003.

[11] Y. Liu, Kefe and X. Liz," A Hybrid Genetic Based Clustering Algorithm", Proceedings of the Third InternationalConference on Machine Leaming and Cybernetics, Shanghai,26-29 August 2004.

[12] S. Chiang, S. C. Chu, Y. C. Hsin and M. H. Wang, "Genetic Distance Measure for K-Modes Algorithm",International Journal of Innovative Computing, Informationand Control ICIC,ISSN 1349- 4198, Volume 2, Number 1, pp.33-40 February 2006.

[13] L. Haldurai and V. Vinodhini, "Parallel Indexing on Color and Texture Feature Extraction using R-Tree for Content Based Image Retrieval", International Journal of Computer Sciences and Engineering, Volume-03, Issue-11, Page No (11-15), Nov -2015, E-ISSN: 2347-2693

[14] Praveen Ranjan Srivastava and Tai-hoon Kim, "Application of genetic algorithm in software testing" International Journal of software Engineering and its Applications, 3(4), pp.87- 96, 2009.

[15] Ruilian zhao, shanshan lv, "Neural network based test cases generation using genetic algorithm" 13th IEEE international symposium on Pacific Rim dependable computing. IEEE, pp.97 – 100, 2007.

[16] Sreedharamurthy S K and H.R.Sudarshana Reddy, "Feature Subset Selection Using Genetic Algorithms for Handwritten Kannada Alphabets Recognition", International Journal of Computer Sciences and Engineering, Volume-03, Issue-06, Page No (94-99), Jun -2015.

[17] Aly, A.H. and R.C. Peralta,: Comparison of a genetic algorithm and mathematical programming to the design of groundwater cleanup systems, Water Resources Research, 35(8), pp. 2415-2425, 1999.

[18] Amin Dastanpour, Suhaimi Ibrahim and RezaMashinchi, "Effect of Genetic Algorithm on Artificial Neural Network for Intrusion Detection System", International Journal of Computer Sciences and Engineering, Volume-04, Issue-10, Page No (10-18), Oct -2016

[19] Shieh, H-J. and R.C. Peralta,: Optimal system design of in-situ bioremediation using genetic annealing algorithm. In Ground Water: An Endangered Resource, Proceedings of Theme C, Water for a changing global community, 27th Annual Congress of the International Association of Hydrologic Research, pp 95-100, 1997.

[20] Fayad, H, Application of neural networks and genetic algorithms for solving conjunctive water use problems, Ph.D. Dissertation, Utah State University, 152 pp, 2001.

[21] Barth, H, Oceanographic Experiment Design II: Genetic Algorithms, Journal of Oceanic and Atmospheric Technology, 9, 1992, pp. 434-443, 1992.

[22] Porto, V.W., D.B. Fogel, and L.J. Fogel, Alternative neural network training methods, IEEE Expert Syst, June 1995.

Authors Profile

Mr. L. Haldurai done his Bachelor of Science, Master of Computer Applications and Master of Philosophy from Bharathiar University. He currently working as Assistant Professor in Department of Computer Science (PG), Kongunadu Arts and Science College.His main research work focuses on Data Mining and Data Communications. He has 8 years of Teaching experience and 2 years of Research experience.

Ms. T. Madhubala done Bachelor of Science and Master of Science from Bharathiar University. She currently working as Assistant Professor in Department of Computer Science, Kongunadu Arts and Science College. Her main research work focuses on Data Mining. She has 3 years of Teaching experience.

Ms. R. Rajalakshmi done Master of Computer Applications and Master of Philosophy from Bharathiar University. She currently works as Assistant Professor in Department of Computer Science (PG), Kongunadu Arts and Science College. Her main research work focuses on Data Mining. She has 8 years of Teaching experience and 2 years of Research experience.

