

GENERIC ALGORITHMS: EVOLUTION, ENCODING AND THEIR APPLICATIONS

Vikas Nandal¹ and Deepak Nandal²

¹Assistant Professor, UIET, MDU Rohtak (Haryana)

²Assistant Professor, PDMCE, Bahadurgarh (Haryana)

Abstract: This paper is to study the evolution of genetic algorithms, their encoding schemes and their current applications. The discussion includes the theoretical concepts of genetic algorithms, evolution history, encoding and GAs current applications. Theoretical topics under review include basic concepts of GA, genetic operators, encoding techniques and applications. Examples of genetic algorithms application that are included in this review are optimization, economic models, pattern recognition, robotics, artificial life, expert system, electronic circuit design and biological applications.

Keywords: genetic algorithms, genetic operators, genetic algorithms applications.

Introduction

Genetic Algorithms have been intensively studied during the last few decades. Numerous applications have been flourished from the utilization of Genetic Algorithms. Theoretical approaches on genetic algorithms have also helped researchers to understand the techniques that are employed in genetic algorithms. This paper is reviewed as an attempt to provide a brief description of evolutionary concepts, encoding and current application areas of genetic algorithm. This discussion can be carried out roughly into three main parts:

Theoretical aspects of genetic algorithms, encoding schemes, and genetic algorithm applications.

Theoretical aspects of genetic algorithms

Evolution history:

Genetic Algorithm (GA), first introduced by John Holland in the early seventies, is a powerful stochastic heuristic algorithm that is based on the principles of natural selection and natural genetics, which has been successfully, and extensively applied in the area of machine learning and optimization problems [3]. To solve any kind of problem, a GA maintains a population of individuals (also called strings or chromosomes) and probabilistically modifies the population by making use of genetic operators such as selection, crossover and mutation with the goal of providing a near optimal solution for the problem. Genetic Algorithm (GA) is the most powerful heuristic methods to solve optimization problems that are based on inherent selection, a process that drive out the biological evolution. The GA repeatedly modifies a population of individuals at random from the existing population to become parents using which produce the children for the next generation. The population generated, moves towards an optimal solution through successive generations. The population contains a number of sub-populations. Each species defines a variable or a part of the problem

IJCSMS

www.ijcsms.com

which is needed to be optimized. The combination of all species will leads to a

complete solution that provide quality index for fitness function.

Vocabulary for Genetic algorithm:

Explanation of Genetic Algorithm terms is represented in the table given below:

Genetic Algorithms	Meaning of Terms
Chromosome(string, individual)	coding
Genes	Defines a part of coding
Locus	Defines position of the defined genes
Alleles	Values of genes
Phenotype	Decoded solution for gene
Genotype	Encoded solution for gene

Figure 1: Genetic Algorithms Vocabulary

The basic operations that are performed in genetic algorithms are:

- Reproduction
- Crossover
- Mutation

Reproduction is usually the first operator that acts upon population. From the population, two chromosome are selected as parents and crossover to produce offspring's. Crossover is a genetic operator that mates two parents to produce offspring of next generation. After crossover mutation is performed. Mutation is a genetic operator used to maintain genetic diversity from one generation of population to next generation.

Some special codes

In the applications of GAs to various problems, some special genetic codes and encoding methods have been vastly employed [7]. They have outperformed some other classical genetic techniques and played an important role in solving some

typical problems. The typical codes are introduced as below:

Bias code

Bias encoding technique is used to provide the encoding solution space in different member density. In some subspace, in which we find a large possibility of solution, more members required to be encoded.

Uniform code

Uniform code is used to distribute the population members in a uniform manner.

Tri-sector Code

Tri-sector code is a special kind of Bias Code. This code is developed under the umbrella of sectoring scheme. Following the definition, the population will be encoded in three sectors, and in each sector the value of members are biased to some extent.

Symmetric Codes

Symmetric codes includes Horizontal Symmetric Code and Vertical Symmetric Code. They are applied successfully in solving problems having large algorithm complexity, such as robot dynamic control and path planning.

How Genetic Algorithms Work?

Genetic algorithm maintains a population of individuals, say $S(t)$, for generation t . Each individual represents a potential solution for the problem at hand. Each individual is analyzed to give some measure of its fitness. There are two type of transformation followed by the individuals:-

- 1) Mutation, which creates new individuals by making changes in a single individual.
- 2) Crossover, which creates new individuals by combining parts from two individuals.

The new individuals, called offspring $R(t)$, are then evaluated. A new population is formed by selecting the more fit individuals from the parent population and offspring population.

3) After several generations, genetic algorithm converges to the best individual, which hopefully represents an optimal or suboptimal solution to the problem. The general structure of the Genetic algorithms is as follow:

```

Begin
{
t=0;
Initialize  $S(t)$ ;
Evaluate  $S(t)$ ;
While (not for termination condition) do
Begin
{
Apply crossover and mutation to  $S(t)$  to
produce  $R(t)$ ;
Evaluate  $R(t)$ ;
Select  $S(t+1)$  from  $S(t)$  and  $R(t)$ ;
t=t+1;
}
End
}
End
    
```

Genetic Operator Technique

Spears in 1995 has introduced the use of combination between uniform crossover and two-point crossover in the same population. Each individual, which is coded in binary form, has an extra bit attached to its chromosome. This extra bit is used to indicate which kind of crossover will be used by an individual when it mates. This extra bit will not be effected by mutation operation. The tag bit of the offspring is depended upon the tag bit of its parents. If the parents have the same tag bit, the tag bit of the offspring will be the same as that of its parents. On the other hand, if the parents have different tag bits, the type of crossover and the tag bit of the offspring will be defined by using some different scheme.

Developmental Mechanism in Genetic Algorithms.

Hart et al. in 1994 have stressed the importance of developmental mechanisms in genetic algorithms. The term "development" is used in the context of a process by which genotypes are transformed into phenotypes. Developmental mechanisms in genetic algorithms contain two major parts: maturation and learning. In the Maturation process, a genotype is mapped into a phenotype and learning refers to a local search. A number of advantages are gained from the use of maturation and local search. Hart et al. (1994) have explained these advantages in three different issues: fitness transformations, time complexity and evolutionary bias.

Genetic Algorithm Applications

Optimization:

GAs have been used in a wide variety of optimization tasks, including numerical

optimization, and combinatorial optimization problems such as traveling salesman problem (TSP), circuit design [Louis 1993], job shop scheduling [Goldstein 1991] and video & sound quality optimization.

Automatic Programming: GAs have been used to evolve computer programs for specific tasks, and to design other computational structures, for example, cellular automata and sorting networks.

Economic model: GAs have been used to model processes of innovation, the development of bidding strategies, and the emergence of economic markets[4].

Pattern Recognition Applications:

In any pattern recognition application, it is desirable to obtain a minimum feature set which yields a maximum classification accuracy. Imam and Vafaie (1994) have performed a comparative study between the use of genetic algorithm and important score method for reducing the number of feature that are used in pattern recognition application making the use of learning system[6]. Genetic algorithm is used to search the space of all possible subsets of the complete set which includes all features. The feature set is coded as a binary string in which each bit represents presence or absence of a particular feature. The classification accuracy will be used as fitness. Genetic algorithm gives a better classification accuracy as compared to score method with the expenses of a higher number of iterations required and a higher number of features required in the reduced feature set.

Robotics and Artificial Life Applications:

In 1994 a steady-state genetic algorithm was used to evolve sensory characteristics

of artificial organism in an environment with controlled complexity. The environment model used is called a latent energy environment (LEE). Feed-forward neural networks are used to simulate organisms. Two types of sensors used in this study are - contact and ambient. Contact sensors are presented in the organism which is required to learn avoidance tasks. Reinforcement learning is used to train motor actions which are the outputs from neural network[2]. Ambient sensors are presented in organism in which approaching task is required. Back-propagation learning is used to train sensors prediction outputs of the network. Steady-state genetic algorithm is used in this study as follows:

Each individual that is represented by neural network, must acquire energy from atoms in the environment beyond a predefined threshold before it can asexually reproduce. If the energy level within an individual is lower than threshold value, that individual will die[5]. Chromosome of each individual contains two parts, one is in floating-point format, and the other in binary format. The connection weights in neural network are coded into the floating-point section of chromosome. Mutation is done by randomly added uniformly distributed noise to the chromosome.

Expert System Applications:

Genetic algorithm can be used to validate an expert system. The objective of testing an expert system is to determine input combinations which will cause the expert system to result in inappropriate responses, so that changes can be made to the expert system[1]. It become exhaustive to test an expert system with all possible input combinations. Genetic algorithm can be used to generate test inputs for the expert

system. This results in an optimal number of test cases that yield a good coverage of all possible input combinations to the expert system

Applications in Biology and Medicine.

In 1995, genetic algorithm were used to model an evolution in an immune system. Chromosome of each individual are used to represents libraries of genetic material in the immune system. These genetic materials are used to construct antibodies in order to recognize antigens. Unlike other applications using genetic algorithm, phenotype of an individual which is represented by antibodies is not a complete mapping from every gene in the chromosome. Chromosome is divided into four libraries of genetic material. Each library contains eight elements. An antibody is produced by combing one element from every library [2]. This study has demonstrated that genetic algorithm is capable of improving fitness of the population even if only partial information about each individual is given to the algorithm during each generation. Further research in this field have introduced a method involving the use of genetic algorithm in gene sequencing. The simulation is based on an actual sequencing method called as short-gun sequencing method. In this method firstly, DNA is replicated many times and then individual strands of the double helix are broken randomly into fragments. These fragments (from both strands) are then used in pair-wise relationship computation process. Each pair of fragments is compared and their similarities are determined, resulting in an overlap strength. All possible orientations and alignments are tried in order to

maximize the overlap strength. The overlap strength will be the key to fitness value which is used in genetic algorithm[2]. All fragments are then totally ordered. Different possible permutations of fragment order will be used to represent individuals in genetic algorithms. Once the ordering process starts, a number of fragments will form a continuous layout known as contig. Initially, there will be many short length contigs presented in the process. Moving toward the end of the process, contigs should increase in their length and reduce their number. Finally, the end sequence should contain only one contig.

Advantages and Disadvantage of Genetic Algorithms:

Perhaps it isn't obvious why such an algorithm should lead to accurate solutions for optimization problems. Crossover is a crucial aspect of any genetic algorithm, but it may seem that it will dramatically change parents with a high fitness function so that they will no longer be fit. However, this is not the case. As in biology, crossover can lead to new combinations of genes which are more fit than any in the previous generations. Other offspring will be created which are less fit but these will have a low probability of being selected to go on to the next generation. Creating new variants is the key to genetic algorithms, as there is a good chance of finding better solutions. This is why mutation is also a necessary part of the genetic algorithms. It will create offspring which would not have arisen otherwise, and may lead to a better solution.

Other optimization algorithms have the disadvantage that some kind of initial guess is required and this may bias the final result. GAs on the other hand only require a search

range, which need only be constrained by prior knowledge of the physical properties of the system. Effectively they search the whole of the solution space, without calculating the fitness function at every point. This can help avoid a danger in any optimization problem which is being trapped in local

maxima or minima. There are two main reasons for this:

1) The initial population, being randomly generated, will sample the whole of the solution space, and not just a small area.

2) Variation inducing tactics, i.e. crossover and mutation, prevent the algorithm being trapped in one part of the solution space. Genetic algorithms can be employed for a wide variety of

optimization problems[3]. They perform very well for large scale optimization problems which may be very difficult or impossible to solve by other traditional methods.

The disadvantage of genetic algorithms is that it, Sometimes, have trouble finding the exact global optimum because there is no guaranty to find best solution. Another drawback that GAs require large number of response (fitness) function evaluations depending on the number of individuals and the number of generations. Therefore, genetic algorithms may take long time to evaluate the individuals.

CONCLUSION

Genetic encoding techniques have significant impact on GAs performance in solving some problems with big algorithm complexity. Our study shows that, in some problems some special codes can be used, to get convergence or the solution. This study reveals that GAs with the special codes can find solutions with better quality in shorter

time than some classical GAs..In summary, genetic encoding techniques, such as Bias Code, Uniform Code, Tri-sector Code and Symmetric Code, are presented in this paper. This paper summarizes a number of current definitions that are used in genetic algorithms. It includes both theoretical aspects of genetic algorithms and some potential applications which makes the use of genetic algorithms.

REFERENCES

- [1] Andre, D., Bennett 111, F. H. and Koza, J. R. (1996b). Evolution of intricate long-distance communication signals in cellular automata using genetic programming. Artificial Life V: Proceedings of the Fifth International Workshop on the Syntlwsis and Simulation of Living Systems, Cambridge, MA MIT Press.
- [2] Das, R. Mitchell, M. and Crutchfield, J. P. (1994). A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H. P. Schwefel and R. Manner (Eds.), Lecture Notes in Computer Science 866 - Parallel Problem Solving from Nature - PPSNIII, International Conference on Evolutionary Computation, The Third Conference on Parallel Problem Solving from Nature (pp. 344-353). Berlin, Germany: Springer-Verlag.
- [3] D. E. Goldberg, Genetic Algorithm in Search,Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc 1989.
- [4] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithm", IEEE Trans. On Systems, Man, and Cybernetics, Vol. 16, NO. 1, 1986.

- [5] K. F. Man, Member, IEEE, K. S. Tang, and S. Kwong, Genetic Algorithms: Concepts and Applications, IEEE Transactions on Industrial Electronics, Vol. 43, No. 5, October 1996.
- [6] Rizki, M.M., Zmuda, M.A., & Tamburino, L.A. (2002). "Evolving pattern recognition systems." IEEE Transactions on Evolutionary Computation, 6(6), 594-609
- [7] Q. C. Meng, Genetic Algorithms and Their Application, Jinan, Publishing Company of Shandong University, August, 1995.
- [8] Q. C. Meng, "An Approach on Genetic Algorithm with Symmetric Codes", Journal of ACTA Electronic Sinica. Vol. 24, No. 10, 1996.
- [9] Q. C. Meng, et al, "The Effects of Partial Genetic Inheritance Technique on Genetic Algorithm with Symmetric Codes", Journal of Yantai University, Vol. 10, No. 1, 1997.
- [10] Q. C. Meng and T.J. Feng.,Z.Chen "Genetic Algorithm encoding study and a sufficient convergence condition of GAS.in Proc. Of IEEE Transactions.
- [11] Wikipedia (2010). Genetic Algorithm. Available from
URL:http://en.wikipedia.org/wiki/Genetic_algorithm.