AI Project Report-1

Academic Year- 2020-21

On

<Analysing the Convergence Rate of Genetic Algorithms >

Submitted by

1. Arjun Bakshi BT18GEC134 CSE
2. Somanshu Singh BT18GCS111 CSE
3. Yukta Sharma BT18GCS181 CSE

I. INTRODUCTION

Inspired by the theory of evolution given by Charles Darwin, John Holland in 1960 proposed to use this 'natural' approach in order to solve complex problems. In theory the chromosome is juxtaposed with an individual in a generation. These chromosomes with their respectives genetic order together form a generation(of solutions). Evolution is embodied in a loop where each iteration represents a population, a reproduction operation followed by a deliberate mutation. Prior to the reproduction a selection function selects the best chromosomes which are then mated. The 'reproduction' is implemented by cross-mutating the selected chromosomes(solutions) in a specific way and thereby populating the next generation. After this we introduce some 'mutations' into the new population deliberately to ensure that the same population is not repeated(to avoid local optima). This loops on till our termination condition is met which is generally a lower limit on the amount of change noticed between consecutive generations.

II. PROBLEM STATEMENT AND OBJECTIVE

The genetic algorithm converges when the goal state is reached, this convergence depends on the genetic operators like fitness function, selection function, crossover function, mutation function, and removal function. The convergence rate determines the effectiveness of the algorithm, now since we are testing out a variety of genetic operators, the convergence rate will be affected either positively or negatively. Therefore determining the best possible genetic operators for the algorithm is crucial.

Hence, the objective is to analyze combinations of various genetic operators and select the combination which results in the best convergence rate.

III. LITERATURE REVIEW

In [1] the author explains the genetic algorithm and specifies as to how it works differently than other solving techniques. A game is also explained, "The Prisoner's Dilemma " where the GA concludes a maximum gain strategy. [2] Gives a concise working of the GA, explaining at each step the different variations that are possible in implementing each step. [5] and [3] are more survey oriented and provide information on the classification and application range of GA. [4] uses the GA to solve the problem of Job Shop scheduling, a problem widely researched by the GA approach. Here, the author has not explicitly mentioned the encoding technique of the problem but the remaining analysis is detailed. [6] is also an interesting application of GA to the problem of the minimum dominating set of queens. The author here explains the technique of encoding the problem into binary(most generally used).

In [7] we came across multiple standard test functions against which optimization algorithms are

tested. Out of the many listed algorithms, Rosenbrock Function and Rastrigin Function, are the standard test functions and also mostly used test functions to test out any optimization algorithm.

IV . CHALLENGES

Encoding is the first challenge we come across. The most general encoding technique is binary encoding, hence the first challenge we face is to encode the possible solutions of a problem in binary. Selection of initial population: The size of the population plays the major role in affecting the quality of solution. According to the research the large population means that algorithms take more computation time. However, the small population may lead to poor solutions. Selection of efficient fitness function: It plays an important role in selecting the fittest individual in every generation of an algorithm. If the number of iterations are small, then costly fitness functions can be adjusted. The number of iterations increases may increase the computational cost. The selection of fitness depends upon the computational cost as well as their suitability. Degree of mutation and crossover: The degree of these operators greatly affect the performance of GAs. If the mutation is not considered during evolution, then there will be no new information available for evolution. If crossover is not considered during evolution, then the algorithm can result in local optima. Therefore, the proper balance between these operators is required to ensure the global optima.

VII. REFERENCES

[1] Genetic Algorithms - Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand by John H. Holland

[2] A study on Genetic Algorithms and its Applications

[3] A review on genetic algorithms, past, present and future

[4] Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems

[5] Genetic Algorithm - Survey Paper

[6] A Genetic Algorithm Based Approach for Solving the minimum Dominating Set of Queens Problem

[7] A Genetic Algorithm on Optimization Test Functions