

Genetic Algorithm Approach for cognitive radio technology

B.Vijayalakshmi¹, K. Harika², A.Srujana³, J.Mounika⁴, K.Darshana⁵
Assistant Professor¹, Student^{2,3,4,5}

^{1,2,3,4,5}Department of Electronics and Communication Engineering,
^{1,2,3,4,5}Gayatri Vidya Parishad College of Engineering for Women,
Visakhapatnam, Andhra Pradesh, India

Abstract: In this paper we have gone through a very brief idea on Genetic Algorithm, which is a very new approach for problems related to Optimization. There are many techniques used to optimize a function but in case of optimizing Multimodal Functions most of these techniques face a common problem of robustness. This can be overcome by using Genetic algorithm. Through this paper we will learn how the Genetic Algorithm actually works with proper explanation and with some real time examples based on MATLAB.

I. INTRODUCTION

Nature follows a very interesting path to select an optimum solution of any problem. By generations it chooses and keeps the best and fittest solutions as survivor and discards the rest other options. The fittest solutions again evolve continuously as the test ground of the nature gets more difficult. This is a continuous and almost flawless process to find out the most optimum result. We all know this process under the heading "Survival of the Fittest". In the year 1975 a person named John Holland, from the University of Michigan, first attempted to apply this natural selection technique to optimize artificial problems. His goal was to find out one robust technique to optimize all kind of functions, one robust technique which can provide global optimization to any function. His approach was the building steps of Genetic Algorithm.

2 Genetic Algorithm approach:

The Genetic Algorithm is an example of a search procedure that uses a random choice as a tool to guide a highly exploitative search through a coding of a parameter space.

The principle and procedure of Genetic algorithm can be summarized under the following,

- Outline of the Algorithm.
- Initial population.
- Creating the Next Generation.
- Plots of Later Generation.
- Stopping Condition for the Algorithm.

Let's have a brief idea on each section.

1. OUTLINE OF THE ALGORITHM: The following outlines summarized how the Genetic Algorithm works.

- a. The algorithm begins by creating a random initial population.
- b. The algorithm then creates a new generation or population by using the individuals of the current population.
- c. To create new generation the algorithm performs crossover and mutation in between the individuals of the current generation.
- d. Replace the current generation by the children of the next generation.
- e. The children of the next generation are chosen on the basis of their best fitness value.
- f. The algorithm stops when one of the stopping criteria is met.

2. INITIAL POPULATION:

The algorithm begins by creating a random initial population for a given problem.

3. CREATING NEXT GENERATION: At each step, the genetic algorithm uses the current population to create the children that makes up the next generation. The algorithm selects a group of individuals in the current population, called 'Parents', who contribute their genes (The Entries Of Their Vectors) to their children. The parents are selected on the basis of 'Fitness' to create better children. Genetic Algorithm creates three types of children, which are,

- a. Elite Children: The individual in the current generation with best fitness value will automatically survive to the next generation.
- b. Crossover Children: The children created by combining the genes from a pair of parents selected on count of Fitness.
- c. Mutation Children: The children created by introducing random change or mutation in the gene of a single parent.

As the number of generation increases the average fitness value of the entire generation increases and the individuals in the population get closer to the global minimum point [0,0] of optimization.

4. PLOTS OF LATER GENERATION: As the algorithm goes on the graphical plot of each successive generation shows the increasing closeness of the outcome to the global optimum point.

5.STOPPING CONDITION FOR THE ALGORITHM:The Genetic algorithm uses the following five conditions to determine when to stop.

- i. Maximum Number of Generation.
- ii. Maximum Time Limit.
- iii. Maximum Fitness limit.
- iv. Stall Generations.
- v. Stall Time Limit

The Algorithm stops as soon as any one of these six conditions met.

1. Construct the proper chromosome structure.
2. Generate the initial population from random chromosomes.
3. Calculate the fitness measure for each chromosome.
4. Construct a new population using the following steps

- Selection: Choose the fit chromosomes based on specified fitness measure and discard others, among the fit
- chromosomes select the best two and save them as elite chromosomes.
- Crossover: Apply the crossover function on the selected chromosomes other than the elite chromosomes with
- a determined crossover function and rate.
- Mutation: The chromosomes are mutated with a determined probability of mutation. In mutation every gene
- of the chromosomes is represented in form of binary bit strings. Put the chromosomes into the new generation.

5. Return to Step 3 until the new generation contains the desired individuals. Replace the old population with the newly generated population.

6. Terminate: When a good enough solution is achieved or a number of defined generations is produced.

Procedure:

- Start
- Enter nodes less than 200 and re run the program
- Generating x and y co ordinates of the nodes
- Displaying the x co ordinates of the radio nodes
- Variable for the node ids
- Generating node ids and checking that whether any node does not contain any similar id
- ID s of all the nodes are generated
- Calculating the coverage set of each node
- Saving the file to coverage set
- Finding the source node from which the data has to be send
- Finding the destination where the data has to be send

- Data packets to be send
- First column consists of the data
- Calculating the uncovered neighbor sets of n_i
- Finding the common set of two nodes
- Calculate TER for PU and SU
- Values for Heterogeneity
- %Percentage of nodes than are advanced
- Number of Sensors in the box
- Find false positive with various speed and connections
- Secondary user detection
- RSS values generated
- Changing x and y coordinates to genetic elements
- Calculating the distance between genetic elements
- Error rate with every genetic element in finding secondary nodes
- Apply GA
- Calculate
- Identification rate
- Throughput
- Normal users
- Secondary Users
- Secondary Users with GA
- Recognition RATE
- Time
- Stop
-

RESULTS:

1.Ackley Function:

Using GA TOOL:

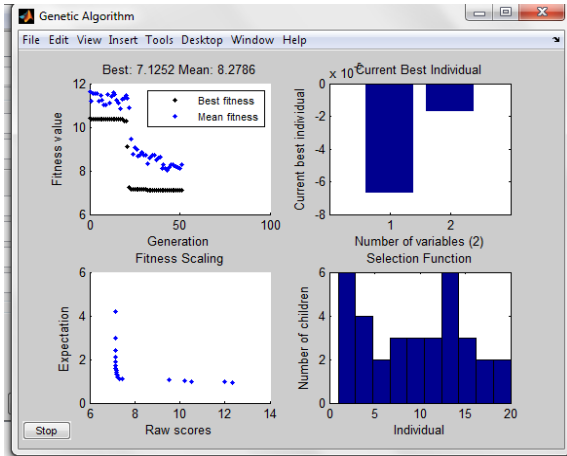
1. Open the gatool window.
2. In the 'Fitness Function' field enter '@ft_ackley'.
3. In the 'number of variable' field enter '2'.
4. Set the following options.
 - Population Type- Double Vector.
 - Population Size- Use Default:20
 - Creation Function- Use Default
 - Initial Scores- Use Default
 - Initial Range- Use Default
 - Selection Function
 - Stochastic Uniform.
 - Elite Count- Use Default:2

5. Click the 'start' button.

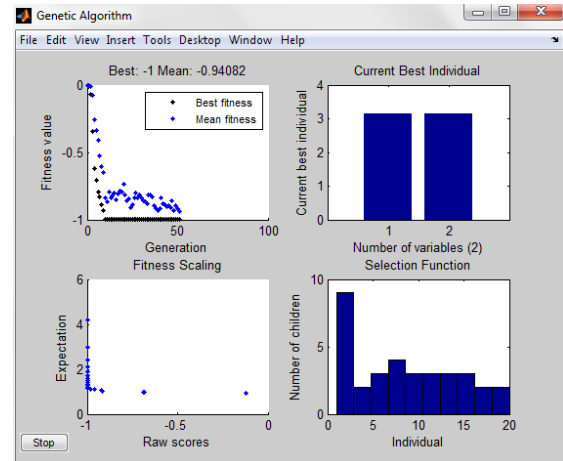
6. To plot the output select the plot interval 1 and mark the 'Best Fitness'.

Output (for 'best fitness', 'best individuals' and 'stall generations'=100)

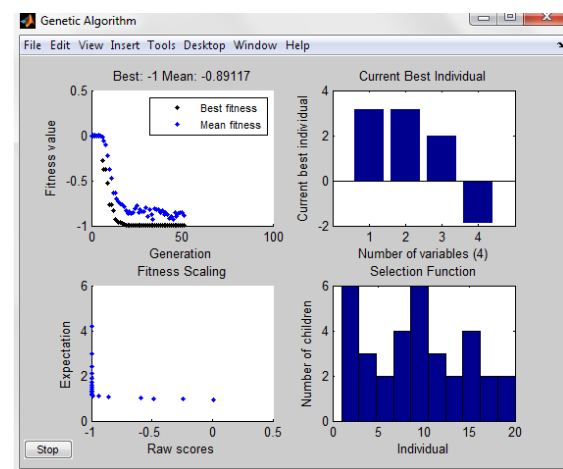
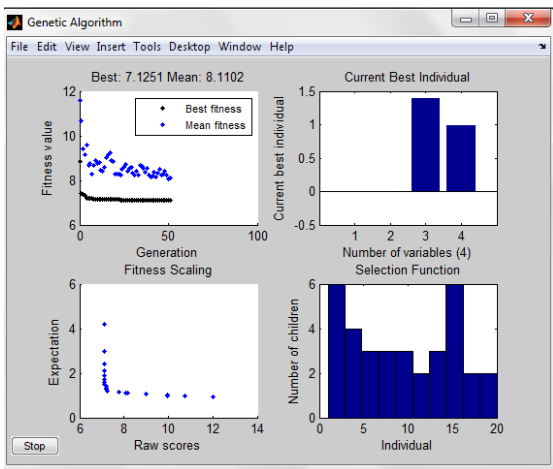
Ackley function using 2 variables:



Ackley function Using 4 variables:



Easom function using 4 variables:



Easom Function:
Using GA TOOL:

1. Open the GA TOOL window.
 2. In the 'fitness function' field enter '@easom'.
 3. In the 'number of variables' field enter '2'.
 4. Click the 'start button'.
 5. When the algorithm ends, the 'status and result' pane shows the result.
- Output (for 'best fitness', 'best individuals' and 'stall generations'=100)

Easom function using 2 variables:

CONCLUSION:

Parallel processing - Simple calculations because they use just the values of the function optimize Continuous or discrete variables can be optimized with the GA . It does not only provide a single solution but a list of optimum solutions.

REFERENCES:

- [1] 'Tutorial on Genetic Algorithm'- Dr. Adel Abdennour(Electrical Engineering Department).
- [2] 'Teaching Genetic Algorithm Using Matlab'- Y.Z.CAO and Q.H.WU
- [3] 'A Genetic Algorithm for Function Optimization: A MATLAB IMPLEMENTATION'- Christopher.R.Houek
- [4] 'Global Optimization Genetic Algorithm'- Olesya Peshks
- [5] Xiaorong Zhu, L. S.-S. "Analysis of Cognitive Radio Spectrum Access with Optimal Channel Reservation", *Communications Letters, IEEE* , 11 (4), 2007, pp. 304 - 306.
- [6] Xie Jiang, H. I. "Cognitive Radio Resource Management Using Multi Agent Systems", *Consumer Communications and Networking Conference, 2007*.
- [7] 'Genetic Algorithm For Optimization'- AndreyPopov(Hamburg 2005)
- [8] 'A quick look at Genetic Algorithm Methods for optimization'- Reza Khademekbari