Evolutionary Computation with Variable length Chromosome for the Design of Fuzzy Logic System

R. RAJESH
Department of Computer Science, Central University of Bihar, Patna, Bihar, India
E-mail: kollamrajeshr@ieee.org

Abstract: Evolutionary computing (EC) is an optimization technique inspired by the Darwinian’s concept of evolution subject to crossover and mutation in a selective environment where only the fittest will survive. EC is one of the major components of softcomputing. EC has seen its application in all most all fields of science and social science due to its ability for providing simple, powerful, and derivative free, optimization technique. Once we think about fuzzy systems having a set of rules, there is a high need of optimizing (i) the number of membership functions (MFs), (ii) position of MFs, (iii) left/right spread of MFs and, (iv) the number of rules. Hence there is a need of having variable length chromosome to optimize the number of membership functions or to optimize the number of rules. This paper presents the current state of art in evolutionary computation with variable length chromosome for the design of fuzzy logic system. Simulation of benchmarking example for the optimization of number of membership functions (which intern optimizes the number of rules) shows it effectiveness.

Keywords: - Evolutionary algorithm, Variable length chromosome, Fuzzy logic Systems, Evolutionary algorithm with Variable Length Chromosome (EAVLC)

1. Introduction:
Evolutionary computation (EC) has roots in nature and most of the functionalities of EC resembles with that of the biological evolution. The applications of EC can be seen in almost all fields of science and social science because of its optimization capability. For certain applications like rule optimization, number of membership function optimization [6], circuit design, topology design optimization [1][2][3][4][5] etc. variable length chromosomes are needed. This paper deals with evolutionary computation with variable length chromosome (ECVLC) for the design of fuzzy logic systems. This paper is organized as follows. Section 2 deals with a review of variable length chromosome in evolutionary algorithms. Section 3 deals with the design of fuzzy logic system using evolutionary computation with variable length chromosome (ECVLC). Section 4 deals with simulation/results and Section 5 concludes the paper.

2. Variable length chromosome in evolutionary algorithms – a review
Evolutionary algorithms will normally have steps like selection, crossover, mutation and reinsertion and EA’s have been used for learning fuzzy systems for quite number of years [7][8][9]. But when we have variable length chromosome, there needs much better operators to handle it. Hence few concepts have been introduced in [10]. Any authors who makes reference to this paper or cites this paper should also cite the paper “GAVLC: GA with Variable Length Chromosome for the Simultaneous Design and Stability Analysis of T-S Fuzzy Controllers” [10]. The design and implementation of fuzzy logic controller using the concepts of [10] is the main theme of this paper. The various operators used in ECVLC are explained below.

2.1 Various type of Mutations:
Gene Mutation: - Mutation of the values of the genes is done on random positions with a probability $P_{GM}$. This is based on the Darwin’s principles of evolutions.

Gene Deletion (Pruning): Genes will be deleted randomly from a chromosome with a probability $P_{GD}$. There will also be a threshold (indicative parameter about the number of genes) above which the genes should be deleted.

![Figure 1: Illustration for gene deletion (also called pruning)](https://example.com/image.png)
Gene Inclusion (Feeding): Genes will be included in a chromosome with a probability of $P_{GD}$. There will also be a threshold (indicative parameter about the number of genes) below which the genes should be included.

2.2 Various types of crossovers/recombination

Gene Donation: Two chromosomes will be selected randomly and the one chromosome which is having more genes will be donating one or more of its genes to another.

**Gene exchange:** This operator is approximately equivalent to the crossover operator in GA/EA, but the only difference is the variable length chromosome used in gene exchange.

Two Side Sharing: Here two selected chromosomes will share some of its genes with one another without losing the genes.

One Side Sharing: Here one of the two selected chromosome will share some of its genes with other without losing the genes.
Evolutionary Computation with Variable length Chromosome for the Design of Fuzzy Logic System

2.3 Additional Operators:
If the number of genes exceeds the maximum limit, then there is an operator which randomly deletes genes. Moreover due to crossover and mutation, if some genes are repeated then there is an operator which deletes those genes which are repeated.

3. Design of fuzzy logic system using EAVLC
Inverted pendulum is a benchmarking control problem. This section describes the design of fuzzy logic controller for inverted pendulum using EAVLC.

3.1 Structure of Chromosome:
The chromosome will have ‘n’ number of genes. Each gene will correspond to a rule. Initial population will be created randomly.

3.2 Membership functions:
The variable theta (θ) is allowed to take values in [-75°, 75°] and seven membership functions are designed with [left spread, centre, right spread] given as [Inf, -75, 0], [15, 0, 15], [75, 75, Inf], [30, 0, 30], [45, 0, 45], [60, 0, 60].

The variable thetadot (dθ/dt) takes the values in [-1000, 1000] and 30 membership functions are designed as [100, 0, 100], [200, 0, 200], [300, 0, 300], [400, 0, 400], [500, 0, 500], [600, 0, 600], [700, 0, 700], [800, 0, 700], [900, 0, 900], [1000, 0, 1000], [Inf, -100, 100], [Inf, -200, 200], [Inf, -300, 300], [Inf, -400, 400], [Inf, -500, 500], [Inf, -600, 600], [Inf, -700, 700], [Inf, -800, 800], [Inf, -900, 900], [Inf, -1000, 1000], [100, 100, Inf], [200, 200, Inf], [300, 300, Inf], [400, 400, Inf], [500, 500, Inf], [600, 600, Inf], [700, 700, Inf], [800, 800, Inf], [900, 900, Inf], [1000, 1000, Inf].
The output variable, force, takes 21 singleton values, namely, -100, -90, -80, -70, -60, -50, -40, -30, -20, -10, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.

3.3 Rules:
Based on the membership functions, 7x30x21=4410 rules are designed. Now it is the duty of EAVLC to choose the minimum number of best rules from 4410 rules for the fuzzy logic controller.

3.4 EAVLC simulation parameters
The parameters of EAVLC are given in table 1. Two different simulations are carried out with initial number of rules as 1, and 100.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>Mutation’s</td>
<td></td>
</tr>
<tr>
<td>(i)</td>
<td>gene mutation rate</td>
<td>0.2</td>
</tr>
<tr>
<td>(ii)</td>
<td>new gene inclusion rate</td>
<td>0.2</td>
</tr>
<tr>
<td>(iii)</td>
<td>gene deletion rate</td>
<td>0.2</td>
</tr>
<tr>
<td>(iv)</td>
<td>new gene inclusion threshold</td>
<td>20 rules</td>
</tr>
<tr>
<td>(v)</td>
<td>gene deletion threshold</td>
<td>5 rules</td>
</tr>
<tr>
<td>3</td>
<td>Recombination’s</td>
<td></td>
</tr>
<tr>
<td>(i)</td>
<td>Donation rate</td>
<td>0.20</td>
</tr>
<tr>
<td>(ii)</td>
<td>Two side sharing rate</td>
<td>0.20</td>
</tr>
<tr>
<td>(iii)</td>
<td>One side sharing rate</td>
<td>0.20</td>
</tr>
<tr>
<td>(iv)</td>
<td>Gene exchange/crossover rate</td>
<td>0.20</td>
</tr>
<tr>
<td>4</td>
<td>Maximum Generation allowed</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>Maximum number of rules allowed</td>
<td>100</td>
</tr>
</tbody>
</table>
4. Simulations and Results

Figure 6: The performance of the inverted pendulum starting with 40° and -40° for the best set of rules obtained after 100th generation

Figure 6 shows performance of the inverted pendulum starting with 40° and -40° for the best set of rules obtained after 100th generation. It can be seen that the pendulum balances quickly for both the cases. Figure 7 shows the fitness of the best chromosome in each generation for simulations starting with 1 rule (Case 1, straight line), 100 rules (Case 2, dotted line), 25 rules (Case 3, dash dot), 50 rules (Case 4, dash), 75 rules (Case 5, circle). It can be seen that in most of the cases it converges to the optimal fitness values within 20 generations.

Figure 8 shows the number of rules of the best chromosome in each generation for simulations starting with 1 rule (Case 1, straight line), 100 rules (Case 2, dotted line), 25 rules (Case 3, dash dot), 50 rules (Case 4, dash), 75 rules (Case 5, circle). In all most all cases the number of rules converges to a value less than 10 which shows the rule optimization capability of the proposed method.

Figure 7: Fitness of the best chromosome in each generation for simulations starting with 1 rule (Case 1, straight line), 100 rules (Case 2, dotted line), 25 rules (Case 3, dash dot), 50 rules (Case 4, dash), 75 rules (Case 5, circle)
5. Conclusion:
This paper has proposed evolutionary computation with variable length chromosome for the design of fuzzy logic system. The performance of the algorithm is tested using a benchmark marking control problem and the results are promising.

Acknowledgement:
The author would like to thank Central University of Bihar for administration and research support.

References: