A Study of Hardware Efficient Recombination Variants in a Mini Population Genetic Algorithm

Shiva Satya Ramesh Varma Bhupathiraju Venkata
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all
Part of the Computer Engineering Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/961

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
A STUDY OF HARDWARE EFFICIENT RECOMBINATION VARIANTS IN A MINI POPULATION GENETIC ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

By

Shiva Satya Bhupathiraju Venkata.
B.Tech., Jawaharlal Nehru Technological University, 2004

2009
Wright State University

__________________________
John C. Gallagher, Ph.D.
Thesis Director

__________________________
Thomas A. Sudkamp, Ph.D.

Committee on
Final Examination

__________________________
John C. Gallagher, Ph.D.

__________________________
Michael Raymer, Ph.D.

__________________________
Mateen M. Rizki, Ph.D.

__________________________
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies
ABSTRACT

Bhupathiraju Venkata, Shiva Satya, M.S. Department of Computer Science and Engineering, Wright State University, 2009.
A Study of Hardware Efficient Recombination Variants in a Mini Population Genetic Algorithm.

Evolvable Hardware is an emerging sub-field of evolutionary computation in which evolutionary algorithms are employed to create designs for hardware devices. Recent work has combined continuous time recurrent neural networks with the Mini Population (Minipop) Evolutionary Algorithm to create self-configuring device controllers. Standard Minipop eschews recombination operators due to the belief that they increase the size of an algorithm’s on chip implementation without adding significant search power for finding neural network controllers. The focus of this thesis is to challenge that thinking by testing a number of hardware efficient recombination operators against two benchmark problems. We consider variants that recombine at neuron parameter and whole neuron boundaries taking advantage of easily measured neuron output correlation information. Although we conclude that there is no compelling evidence to adopt any of these variants at this time, we have identified interesting opportunities that might be exploited in the future to improve Minipop search over spaces of neurodynamic systems.
# Table of Contents

**Chapter 1 : Introduction**

1.1 Introduction ........................................................................................................................................... 1

1.2 Objective of the Thesis ......................................................................................................................... 3

1.3 Organization of the Thesis ................................................................................................................. 3

**Chapter 2 : Background** ...................................................................................................................... 5

2.1 Evolvable Hardware ............................................................................................................................. 5

2.1.1 Evolutionary Controllers .................................................................................................................. 6

2.2 CTRNN's (Continuous Time Recurrent Neural Network) ................................................................. 8

2.2.1 Definition of CTRNN's ................................................................................................................... 11

2.3 Minipop Algorithm ............................................................................................................................... 12

2.3.1 Motivation for modifying Minipop Algorithm ................................................................................ 15

2.4 Modified Minipop Algorithm ............................................................................................................ 17

2.4.1 Variants of the modified Minipop Algorithm ................................................................................. 23

**Chapter 3 : Control Problems** ............................................................................................................... 25

3.1 Correction of Common Arrhythmias in Simulated Human Heart ..................................................... 26

3.1.1 CTRNN EH Controller Architecture for Heart Problem ............................................................... 33

3.2 Fitness Calculation for Heart Problem ............................................................................................... 33

3.2.1 EA and CTRNN Parameters for Heart Model ............................................................................... 36

3.3 Suppression of Oscillations in Jet Combustion Chamber ................................................................. 37
3.3.1 CTRNN EH Controller Architecture for Jet Combustion chamber problem.............. 41
3.4.1 Fitness Calculation for Jet Combustion chamber problem ................................ 42
3.4.2 EA and CTRNN Parameters for Jet Combustion chamber problem ................... 42

Chapter 4: Experimental Results ................................................................................. 44
4.1 Experimental Results for Heart Problem ............................................................... 44
4.2 Algorithm Assessment Parameters for correction of arrhythmias in simulated Human Heart .................................................................................................................... 46
4.2.1 Experimental results for Time to acceptable for Heart Problem ....................... 47
4.2.2 Experimental results for Time to acceptable for Heart Problem ....................... 49
4.3 Experimental Results for Jet Combustion Chamber Problem ............................... 51
4.4 Assessment Parameters for the Jet Combustion Chamber Problem ..................... 51
4.4.1 Performance assessment in EM1 Configuration ............................................... 52
4.4.2 Performance assessment in EM2 Configuration ............................................... 57
4.4.3 Performance assessment in SM1 Configuration ............................................... 61
4.4.4 Performance assessment in SM2 Configuration ............................................... 66

Chapter 5: Results, Discussion and Future Work ......................................................... 75
5.1 Results Summary and Discussion ......................................................................... 75
5.2 Future Work ......................................................................................................... 77

Bibliography ............................................................................................................... 79
List of Figures

Figure 2.1 : Block Diagram of an Evolvable Hardware Control Device .......................... 9
Figure 2.2 : Architecture of 5 neuron CTRNN ............................................................ 10
Figure 2.3 : Schematic equivalent of indiviual neuron in CTRNN .............................. 10
Figure 2.4 : State equation of Individual neuron in CTRNN ....................................... 11
Figure 2.5: Algorithm for calculating distance vector ............................................... 21
Figure 3.1: Human Heart ............................................................................................ 27
Figure 3.2: Simiplified ECG schematic ....................................................................... 28
Figure 3.3: Time Series plot of 2\textsuperscript{nd} Degree Mobitz type I block .................... 29
Figure 3.4: Time Series plot of 2\textsuperscript{nd} Degree Mobitz type II block .................... 29
Figure 3.5: Time Series plot of 2\textsuperscript{nd} Degree Mobitz type III block .................... 30
Figure 3.6: Schematic representation of coupled oscillator cardiac model ................. 31
Figure 3.7: Interface of CTRNN Feedback Controller for correction of common
arrhythmias problem ................................................................................................... 33
Figure 3.8: Calculation of MSE between for PR interval and RP interval ................. 34
Figure 3.9: Calculation of P and R waves over a period of N ...................................... 35
Figure 3.10: Internal Pressure vs time in uncontroller Jet combustion chamber .......... 38
Figure 3.11: Combustion Chamber with side mounted Speaker .................................. 38
Figure 3.12: Combustion Chamber with End mounted speaker ................................. 39
Figure 3.13: Combustion Chamber with End mounted speaker ................................. 40
Figure 3.14: Interface of CTRNN Feedback controller for Jet Combustion Chamber .... 41
Figure 4.1: Time Series Plot for an assited heart with 2\textsuperscript{nd} degree Mobitz Type I block...45
Figure 4.2: Time Series Plot for an assited heart with 2\textsuperscript{nd} degree Mobitz Type II block...45
Figure 4.3: Time Series Plot for an assited heart with 3\textsuperscript{rd} degree block .........................46
Figure 4.4: Box plot for various algorithms for TTA in Heart Problem.................................48
Figure 4.5: Box plot for various algorithms for Final Fitness values in Heart Problem...50
Figure 4.6: Box plot for various algorithms for TTA for Jet Combustion Chamber in EM1 configuration ..........................................................55
Figure 4.7: Box plot for various algorithms for Final Fitness value for Jet Combustion Chamber in EM1 configuration..........................................................58
Figure 4.8: Box plot for various algorithms for TTA for Jet Combustion Chamber in EM2 configuration ..........................................................60
Figure 4.9: Box plot for various algorithms for Final Fitness value for Jet Combustion Chamber in EM2 configuration..........................................................62
Figure 4.10: Box plot for various algorithms for TTA for Jet Combustion Chamber in SM1 configuration ..........................................................65
Figure 4.11: Box plot for various algorithms for Final Fitness value for Jet Combustion Chamber in SM1 configuration..........................................................67
Figure 4.12: Box plot for various algorithms for TTA for Jet Combustion Chamber in SM2 configuration ..........................................................79
Figure 4.13: Box plot for various algorithms for Final Fitness value for Jet Combustion Chamber in SM2 configuration..........................................................73
List of Tables

Table 3.1: C1 and L1 values for different heart rates.................................................. 32
Table 3.2: Values of PR and RP interval for healthy heart........................................ 35
Table 3.3: Minipop and variants of modified Minipop parameters for heart model...... 36
Table 3.4: CTRNN Parameters for Heart model.......................................................... 37
Table 3.5: Resonant Frequencies for End mount Speaker Configuration....................... 40
Table 3.6: Resonant Frequencies for Side mount Speaker Configuration....................... 41
Table 3.7: Minipop and variants of modified Minipop parameters for Jet Combustion Chamber Problem................................................................................. 43
Table 3.8: CTRNN Parameters for Jet Combustion Chamber Problem ......................... 43
Table 4.1: Average Time to Acceptable solutions for Heart problem ......................... 47
Table 4.2: ANOVA results for the TTA in case of Heart Problem.................................. 49
Table 4.3: Average Final Fitness Values for Heart problem....................................... 50
Table 4.4: ANOVA results for Final Solutions in case of Heart Problem....................... 51
Table 4.5: T-Test for Final solutions for correction of common arrhythmias problem in simulated human heart................................................................. 52
Table 4.6: Time to Acceptable solutions for different algorithms in EM1 configuration 54
Table 4.7: ANOVA results for the TTA in case of Jet Combustion Chamber in EM1 configuration................................................................. 55
Table 4.8: T-Test for TTA solutions, for jet combustion chamber in EM1 Configuration 56
Table 4.9: Final Fitness Values for Jet Combustion Chamber in EM1 Configuration...... 57
Table 4.10: ANOVA results for the Final Solutions in case of Jet Combustion Chamber in EM1 configuration........................................................................... 58
Table 4.11: Time to Acceptable solutions for different algorithms in EM2 configuration ........................................................................................................... 59
ACKNOWLEDGEMENTS

I would like to thank my research advisor Dr. John Gallagher who has given me tremendous support throughout my graduate career. I would also like to thank Dr. Mateen Rizki and Dr. Micheal Raymer for their valuable suggestions and feedbacks about my thesis work. I would also like to acknowledge my friend and colleague Sanjay K. Boddhu for his helpful insights. Finally I would like to thank my Parents for their support and encouragement throughout my educational career.
1. Introduction

1.1 Introduction

Micro Air Vehicles, Micro Robotics and Space Systems have been an evolving discipline in the recent past. Self adaptive and self configuring control devices are an alluring option for the researchers considering the reduced space and power requirements. Our goal as a part of this research project was to combine evolutionary algorithms with reconfigurable computational hardware to construct small and lower power consuming VLSI solutions.

Controllers based on Evolvable Hardware (EH) [1][2], a sub-specialty of Evolutionary Computation (EC) [3][4][5] provide an interesting methodology to help meet the challenging demands of efficacy requirements and resource constraints. In addition to providing a tool to help generate novel designs to specification, EH offers the possibility for generating workable designs against incomplete specifications. Although EH offers distinctly attractive advantages, it is not without flaws. Controllers evolved using EH sometimes may be difficult to explain or understand, making it difficult to employ in situation where a critical validity of the controller is required. Another thing which could be a problem is that EH methods are very opportunistic in nature, i.e., if the controller is evolved on the hardware, the EH may adapt itself to parasitic parameters of the hardware like electromagnetic interferences, to meet this desired objective. Thus sometimes making not only the understanding of the solution more difficult but also very difficult to port across other hardware (platforms). (Evolvable Hardware will be discussed briefly in chapter 2).
Reconfigurable Hardware is one of the main components in EH. CTRNN’s [6][7] are currently our preferred choice as the reconfigurable units in EH for controller applications[8][9][10] (CTRNN’s will be discussed in Chapter 2). CTRNN’s are a special class of artificial neural networks similar to Hopfield continuous model neural networks. One of the reasons for the use of CTRNN’s as the reconfigurable hardware is that CTRNN are known to be universal dynamic approximators [11], given sufficient number of neurons they are capable of approximating any smooth dynamics. Another reason for choice of CTRNN is that it requires very small hardware footprint and have been successfully implemented in several types of hardware [12][13]. CTRNN’s have been successfully used in a wide range of controllers i.e., as controller for walking hexapod robot [14], as controller to control harmful parasitic oscillation in turbine jet engine [15], are also being investigated for use to control a flapping wing robot [12].

Evolvable Algorithm is another important factor in the implementation of Evolvable Hardware. The implementation of Evolutionary Algorithms in hardware for EH is seen in many instances in the literature [16][17][18][19][20][21][22][23]. Minipop is one such algorithm which has been optimized for hardware footprint. The rationale behind optimizing Minipop algorithm for space is that in EH applications [24], where the evaluation of the solutions is done on the hardware, it is not possible to speed up the process of evaluation beyond the time allowed by physics. So, objective evaluation function becomes the bottleneck which cannot be circumvented, thus limiting the speed of execution in an (intrinsic) EH device. Hence optimizing the algorithm for speed makes little sense. Minipop algorithm, which when implemented using VLSI techniques, saves a significant amount of hardware space because of the small population
The Minipop Algorithm, which we will discuss in chapter 2, is a tournament based, mutation driven evolutionary algorithm. In this work, it is the digital EA engine responsible for configuring the analog CTRNN. It was shown in previous work [25] that the Minipop algorithm is sufficiently adept at evolving CTRNN parameters effectively even in noisy environment and under severe size constraints. Minipop, in its canonical form, does not include recombination operators.

1.2 Objective of the Thesis

The basic objective is to modify the mutation and hyper-mutation driven Minipop algorithm to incorporate cross-over operators. It is also the objective of the thesis to evaluate of the performance of the Minipop algorithm against the crossover variants of the Minipop algorithm across two control problems. The choice of the control problems for this thesis is the correction of common arrhythmias in simulated human hearts and the suppression of oscillation in desktop jet combustion chamber engine. These problems were chosen because they had already been treated with traditional EA methods and we already have at least qualitative understandings of their search spaces.

1.3 Organization of the Thesis:

The thesis has been organized into 4 chapters excluding Introduction chapter. The background chapter (chapter 2) deals with giving the reader a brief summary about Evolvable Hardware (Evolvable Controllers), CTRNN’s, Minipop algorithm and Modified Minipop algorithms. The subsequent chapter (chapter 3) is intended to familiarize readers with correction of arrhythmias problem in simulated human heart and suppression of oscillation in jet combustion chamber. The chapter also deals with the
parameters that have been configured for the control problems for the experimental stage.

Chapter 4 deals with the experimental results and the concluding chapter 5 deals with summary of the test results and a discussion of open issues.
Chapter Overview:

This chapter contains an introduction to various background topics that are referenced throughout this thesis. This chapter is split into three major sections: Evolvable Hardware, CTRNN's, Minipop and Modified Minipop Algorithm. Readers who are familiar with these topics may safely skip the respective sections.

2.1 Evolvable Hardware

Evolvable Hardware (EH) is an emerging sub-specialty of Evolutionary computation in which one employs evolutionary algorithms to configure the collections of hardware components into useful forms. There are many promising applications for EH both in the Analog and Digital Fields [26][27][28][29][30][31]. Analog and Digital circuits, evolved via evolutionary algorithms are potentially useful in cases where there is insufficient information available to support traditional design methods. Another potential advantage of using (EH) is that it can dynamically change its configuration to fit the needs of changing plant dynamics or changing operational environments.

In the evolutionary synthesis of hardware, to perform a predefined function, first the population of individuals is randomly generated where each individual is an encoded hardware configuration that could be implemented in the reconfigurable hardware block. Consider a case where objective is to design some digital logic. If FPGA is being used as the reconfigurable hardware, then individual genotype could be a bit string where each bit
in the bit string may be responsible for the connection between two logical units in the FPGA. After generation of the random individuals, their fitness is ascertained by implementing the logic encoded into the each individual on reconfigurable hardware (in this case, by programming an FPGA) and then measuring how closely the circuit performs with respect to a desired objective. After calculation of fitness values, a new generation of individuals is generated from the existing population by allowing current generation’s member representation in the new population as a function of their fitness. The specific generation method used depends upon the specific Evolutionary Algorithm being used. In the new population, some number of the individuals undergoes genomic changes that mimic mutation (the introduction of new traits) and recombination (the combination of traits from more than one individual). The selection of the individuals so treated, and the specifics of the treatment, varies across implementations. The above process of interleaved population generation and population modification continues until some termination condition (eg. one finds an individual that performs well enough) is reached.

### 2.1.1 Evolvable Controllers

Evolvable Controller (EC) refers to using EH methods to evolve a controller for a control problem [8][14][32][33][34][35]. In evolving a controller one has to first decide on the reconfigurable hardware substrate, two common examples are neural networks (CTRNN’s)[8][14] and FPGA circuits [27][26]. Secondly, one has to choose a genotype format that encodes the hardware configurations of potential solutions and an
evolutionary algorithm capable of evolving device descriptions (genomes) to meet a desired objective. The objective is codified as a “fitness function” that the selected EA uses to determine fitness of individual candidate solutions. Figure 2.1 shows a high level block diagram of CTRNN based Evolvable Controller [36] i.e., an evolvable controller which uses CTRNN as reconfigurable hardware.

To evaluate a controller, the evolutionary algorithm configures the CTRNN’s to the configuration encoded in the candidate controller’s genotype. The neural network’s outputs are connected to the effectors of the plant and the inputs to the CTRNN are connected to the sensors monitoring the plant. After setting this configuration, the plant is activated and behavior of the plant is monitored and fitness calculated by the fitness evaluator by measuring the difference between plant’s behavior and plant’s expected behavior. The fitness evaluator reports the measured fitness of the plant to the evolutionary algorithm module as the controller fitness.

Evolvable Hardware based controllers have several advantages when compared to the conventionally designed controllers. Conventional controllers are generally designed by analyzing the approximated mathematical model of the plant to determine what control efforts are required to achieve the desired behavior. Since often times mathematical models are simplified approximations of the actual dynamics of the plant, there could be small changes in mathematically modeled plant behavior and actual plant behavior. These small differences may make little difference for many control problems, and indeed, many control methods adaptively minimize those problems via error feedback. However, there still exist exotic applications that are sufficiently sensitive and require
tuning and modification against the real plant. Additionally, evolution may help expand engineering knowledge by discovering new techniques beyond the bounds of conventional engineering practice. This is because evolutionary algorithms are not constrained by the limits of human understanding and can construct superior devices that are inconceivable to conventional methodologies.

However, there are certain disadvantages in applying evolvable hardware to control devices. For example, one might find artificially evolved controlled devices to be very difficult to describe, characterize or model. This could be difficult as few engineers would tolerate the notion of not able to understand the system dynamics, to understand the potential problem modes and to verify the device operation.

2.2 CTRNN’s (Continuous Time Recurrent Neural Networks)

CTRNN’s (Continuous Time Recurrent Neural Networks) are a superset of Hopfield Continuous Model Neural Networks [37][38]. In CTRNN’s a neuron may receive input from any neuron in the network, including itself, and there are no restrictions of the matrix of interconnections (Hopfield networks are constrained to have only zero diagonal symmetric connection matrices). The architecture of CTRNN’s network, employed in CTRNN-EH is generally a free form network as shown in the figure 2.2. Even though architecture like this can be very difficult to understand, it is sometimes possible to logically modularize the network post evolution and produce behavioral analysis [39]. However unlike modular architecture (where the neurons are arranged in layers), module boundaries may be fuzzy and single neuron may belong to more than one module.
Figure 2.1 Block Diagram of an Evolvable Hardware Control Device.
Fig2.2 Architecture of 5 neuron CTRNN; Each neuron can act as Input or Output or Hidden Neuron.

Fig2.3 Schematic equivalent of individual neuron in CTRNN
2.2.1 Definition of CTRNN’s

CTRNN’s are generalization of the Continuous Hopfield Neural Network [37]. But unlike Hopfield Neural Network, CTRNN’s have self-connections i.e., apart from each neuron being connected to all other neurons, it also has connection to itself. Another characteristic property of CTRNN is unconstrained weight matrices i.e., unlike Hopfield model CTRNN’s doesn’t enforce weight symmetry (i.e., if $W_{ij}$ denotes the strength of the interconnection between neuron $i$ to neuron $j$, then $W_{ij}$ need not be equal to $W_{ji}$). Since no restrictions are imposed on the connectivity and weight matrices of CTRNN’s they are capable of exhibiting richer dynamics [11]. (Figure 2.3, represents an individual neuron).

The mathematical form of a CTRNN can be described by the following state equation (Figure 2.4):

$$\tau \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} W_{ji} \sigma(y_i + \theta_i)$$

Figure 2.4: State equation of Individual neuron in CTRNN

Where $\tau$ in the time constant, $W_{ij} \in \mathbb{R}$ is the weight of the connection between neuron $j$ to neuron $i$, $\theta \in \mathbb{R}$ is the bias, $\sigma$ is the standard sigmoid function, $N$ is the number of neurons in the network and $y_i$ represents the state of the neuron $i$.

Evolutionary computation is the most prevalent method used to configure CTRNN though cases of recurrent versions of traditional feed-forward neural network training
have been explored. The literature of CTRNN’s shows that there are different evolutionary types of algorithms that are usually used to evolve CTRNN’s, the Real Valued Genetic algorithm [41], Net Crawler [42], Minipop [43], and Star-CGA [44].

2.3 Minipop Algorithm:

Minipop Algorithm [43][45], a tournament based, mutation driven compact genetic algorithm, is the digital EA engine responsible for configuring the analog CTRNN. The Minipop Algorithm is characterized by its use of mutation and hyper-mutation and its rejection of recombination operators. The variants introduced and tested in this thesis, in fact, introduce recombination operators to examine if they can add benefit without adding large hardware cost to the digital circuitry that implements it. The population in Minipop Algorithm consists of small number of individuals where each individual is a fixed-precision binary encoding of the real-valued parameters of CTRNN. In the Minipop algorithm, the individuals for the next generation are determined by binary tournament held between each individual and its mutated version. Mutation is accomplished by randomly flipping bits in a parent’s bit-string. The hyper-mutation tournament is a special tournament, run in a fixed ratio to other tournaments, where the worst member of the population competes against a randomly generated individual.

Minipop’s use of bit-string representation is driven by an interest in online evolution of low power environment such as mobile autonomous robots. Bit-string based solution representation and simple mutation and selection operator allow Minipop to be easily implemented in low-power digital VLSI hardware to contemplate low-power analog
VLSI CTRNN implementation [46]. The algorithmic implementation of Minipop is shown below.

**Minipop Algorithm:**

1. **Start**
   
   a. `Max_Evaluations = MAX`; -> Maximum number of evaluations
   
   b. `Population_Size = N;`
   
   c. `population []`; -> Population Array
   
   d. `mut_population[]` -> Array to hold Mutated version of population.
   
   e. `fitness_pop[]` -> Array to hold fitness of the population
   
   f. `fitness_mut_pop[]` -> Array to hold fitness scores of mutated population.
   
   g. `hypermutant` -> Data Structure to hold Hypermutant
   
   i. `Fitness_hypermutant` -> A real valued variable to store fitness of hypermutant.

2. **For** `i = 1` to `N` **do**
   
   a. randomized bit string are generated and stored in `population [i]`.
   
   b. Fitness of `population[i]` is evaluated and stored in `fitness_pop[i]`.

3. **Done //For Loop Ends.**

4. `evaluations = evaluations + N;`

5. **While** `(evaluations <= MAX)` **do**
   
   a. **For** `i = 1` to `N` **do**
i. An Individual Population[i] is selected and mutated, and is stored in mut_population[i].

ii. Fitness of mut_population[i] is evaluated and stored in fitness_mut_pop[i]

iii. If (fitness_mut_pop[i] > fitness_pop[i]) then

1. population[i] = mut_population[i].

2. fitness_pop[i] = fitness_mut_pop[i].

iv. end if.

v. evaluations = evaluations + 1;

b. Done //For loop Ends.

c. A completely randomized bit-string is generated and is stored in hypermutant.

d. evaluations = evaluations + 1;

e. Fitness of the hypermutant is evaluated and stored in fitness_hypermutant.

f. The worst member of the population is determined and its fitness is stored in worst_member_fitness.

g. If (fitness_hypermutant > worst_member_fitness) 

i. pop[worst_member] = hypermutant

ii. fitness[worst_member] = fitness_hypermutant.
6. Done //While loop ends.

7. Determine the best member of the population.

8. Return pop[index_best_member].

9. End

2.3.1 Motivation for Modifying Minipop Algorithm:

Minipop algorithm, which has been optimized for saving space, has been proven to be adequate enough for various problems like evolution of controllers of six-legged insect, controller for suppression of oscillations in jet combustion chamber, etc. But when the algorithm is applied to problems whose fitness landscape has large number of valleys and the algorithm is caught in one such valley, mutation may not be sufficient to steer the algorithm out and algorithm relies solely on hyper-mutation to steer it out of the valley. Introduction of Cross-over operator might be helpful in driving the Minipop algorithm out of these valleys (Since in Minipop algorithm the individuals compete with mutated version of themselves, mutation may not be helpful in driving the algorithm out of the valleys). This has been one of our motivating factors for modifying the Minipop algorithm to accommodate cross-over operator.

The other factor which has driven us to cross-over operator has been our assumption that evolved CTRNN’s may exhibit modularity for some of the control problems, i.e., it may be possible to identify a neuron (group of neurons) that acts (together) to perform specific
function. So it has been thought introduction of the cross-over operator may be good way preserve modular characteristics of the neuron(s) among parents and children so as to increase the chances of producing the better offspring’s.

But introduction of cross-over operator has its own set of challenges for CTRNN architecture. It has been stated and observed that the crossover operator has a tendency of convergence, i.e., after certain number of generations because of selection pressure there would gradual decrease in the diversity of the population. Since the size of the population is too small in Minipop, this might result in the premature convergence of the population. So one of the modifications (rather a restriction) that has been made for better search efficacy using crossover operator is the selection of parents that are as diverse as possible to avoid premature convergence. There are two methods adopted for selection procedure of the parents, in one method the best and worst individuals are selected for cross-over and in second method, which we will discussed in the next section, two individuals are picked from the population which are very much unlike each other for cross over.

Another restriction that has been placed on the crossover operator is the selection of the crossover point, which can happen only at certain fixed points of parameter boundaries of individual neuron in one case and at boundaries of individual neuron in another case,(explained in 2.4.1). One reason for placing restriction on the crossover point, i.e., crossover happening only boundary of neuron or boundary of parameter of individual neuron is to take advantage of the modularity exhibited by the CTRNN architecture. The type of crossover operator that has been selected for crossover operation is Uniform Crossover. The reason for choosing uniform crossover over 1-point or 2-point is that
uniform has no bias associated with it unlike 1-point or 2-point crossover and it also has better recombination potential compared to 1-point or 2-point crossover[47][48].

### 2.4 Modified Minipop Algorithm:

The new modified version of the algorithm uses the cross-over operator, as the secondary search operator in addition to mutation and hyper-mutation. In this new version, cross-over can happen only at certain fixed points (at the boundaries of parameters of individual neuron or at the boundaries of the individual neuron itself.) Another major modification that has been made to the algorithm is calculation of the distance vector that is used to determine the distance (degree) the individuals in the population vary with respect to each other. This parameter is then used for the selection of the individuals which are best for cross-over, i.e., the individuals which are unlike each other are selected for cross-over.

**Modified Minipop Algorithm:**

1. Start

   a. $Max_{Evaluations} = MAX$;

   b. $Population\_Size = N$;

   c. $population[]$ - > Population Array

   d. $mut\_population[]$ - > Array to hold Mutated version of population.
e. fitness_pop[ ] -> Array to hold fitness of the population

f. fitness_mut_pop[ ] -> Array to hold fitness scores of mutated population.

g. distance[ ] -> Array to hold the distance by which the individuals in population vary with respect to each other

h. crossover_child[2] -> Array to hold the individuals created by cross over.

i. betterchild -> Data structure to hold better child of the crossover.

2. For i = 1 to N do

    a. randomized bit string are generated and stored in population[i].

    b. Fitness of population[i] is evaluated and stored in fitness_pop[i].

3. Done //For Loop Ends.

4. While (evaluations <= MAX) do

    a. evaluations = evaluations + 1;

    b. For i = 1 to N do

        i. An Individual Population[i] is selected and mutated, and is stored in mut_population[i].

        ii. Fitness of mut_population[i] is evaluated and stored in fitness_mut_pop[i]

        iii. If (fitness_mut_pop[i] > fitness_pop[i]) then

            1. population[i] = mut_population[i].

            2. fitness_pop[i] = fitness_mut_pop[i].
iv. end if.

v. evaluations = evaluations + 1;

vi. If( cointoss() < crossover_prob)

1. Select two suitable parents. // For parental selection algorithm is described below

2. Perform boundary cross-over

3. better_child = crossover_child[2]

4. if(better_child.fitness < crossover_child.fitness)

   then

   a. better_child = crossover_child[1].

5. end if

6 if(better_child.fitness > f_w_member)

   a. population[index_worst_member] = better_child

   b. fitness_pop[index_worst_member] = better_child.fitness

7. endif

c. Done //For loop Ends.

d. A completely randomized bit-string is generated and is stored in hypermutant.

f. evaluations = evaluations + 1;

g. Fitness of the hypermutant is evaluated and stored in fitness_hypermutant.

h. The worst member of the population is determined and its fitness is stored in worst_member_fitness
i. If (fitness_hypermutant > worst_member_fitness)
   i. pop[worst_member] = hypermutant
   ii. fitness[worst_member] = fitness_hypermutant.

j. End if.

5. Done //While loop ends.

6 Determine the best member of the population.

7. Return pop[index_best_member].

6. End

Parental Selection Algorithm:

1. Start

   a. Parent[1] -> datastructure to hold parent one
   b. Parent[2] -> datastructure to hold parent two
   c. distance_vector -> array to hold distance between individuals
      initialized to zero.
   d. Index -> A variable to hold the index of the individual in the
      population

2. Initialize parent[1] = population[worst_individual]

3. For i = 1 to N do
a. \( \text{if}(\text{distance}_\text{vector} < \text{distance}_\text{between}(...)) \)

i. \( \text{distance}_\text{vector} = \text{distance}_\text{between}(...)) \)

ii. \( \text{index} = i \)

b. \( \text{end if} \)

4. \( \text{end for} \)

5. \( \text{Parent}[2] = \text{pop}[\text{index}] \)

6. \( \text{End} \)

\textbf{Computation of } a_{ij} \text{ of matrix A}

\text{If } i > j
  \text{if } y_i > (1\text{-tolerance}) \text{ (neuron excitation) and } y_j > (1\text{-tolerance}) \text{ (neuron excitation)}
    a_{ij} = a_{ij} + 1;
  \text{If } i < j
    \text{if } y_i > (1\text{-tolerance}) \text{ (neuron excitation) and } y_j < (\text{tolerance}) \text{ (neuron inhibition)}
      a_{ij} = a_{ij} + 1;
    \text{If } i = j
      a_{ij} = 0;

\textbf{Computation of } b_{ij} \text{ of matrix B}

\text{If } i > j
  \text{if } y_i < (\text{tolerance}) \text{ (neuron excitation) and } y_j > (1\text{-tolerance}) \text{ (neuron excitation)}
    b_{ij} = b_{ij} + 1;
  \text{If } i < j
    \text{if } y_i < (\text{tolerance}) \text{ (neuron excitation) and } y_j < (\text{tolerance}) \text{ (neuron low)}
      b_{ij} = b_{ij} + 1;
  \text{If } i = j
      b_{ij} = 0;

\textbf{Figure 2.5 : Algorithm showing calculation of Distance Vector}

The population size for this algorithm has been fixed for small number of individuals, to make the realization of the algorithm in hardware as compact as possible. The individuals
are the real number encoding of the parameters for neurons (like weight matrices, bias, time constant). The initial population is drawn using a uniform random distribution and fitness scores of each individual are computed. The distance vector, which is used for selecting individuals which differs the most, is calculated. The calculation of the distance vector could be problem specific. In our case it is based on the computations of the correlation matrices of the excitation states of each neuron. Correlation matrices are computed as set of two matrices A and B. The algorithm for calculation of the correlation matrices is shown below in Figure 2.5. The distance vector between two individuals is calculated as the Euclidian distance between the corresponding correlation matrices. Now each individual is selected and randomly mutated and is replaced with parent, if mutated version has better fitness than parent. Now depending on outcome of random function it is determined whether crossover has to take place or not. If the crossover takes place then the worst (best) individual and individual which differs the most with respect to worst (best) are chosen for crossover. The crossover could happen at the parameter level of individual neuron or at the boundaries of the neuron itself. After crossover, the individual which is best between the children replaces the worst individual, if its fitness is better than the worst. After this step Hyper-mutation tournament is conducted, i.e., a randomly generated individual is created and it replaces the worst individual in the population if its fitness is better than worst individual. After the hyper-mutation the best individual of the population is returned. The loop of mutation tournament, cross-over tournament and hyper-mutation tournament is repeated until the maximum number of evaluation cycles is reached.
2.4.1 Variants of the Modified Minipop Algorithm

The modified Minipop algorithm can be broadly divided into two categories depending upon the crossover point i.e., crossover happening at the boundary of the individual neuron or crossover happening at the boundary of the parameters of the individual neuron.

**Category I:** The variants of the modified Minipop where crossover is happening at the boundary of the individual neuron are categorized into this group. This group is further divided into 3 different sets depending upon the parental selection for the crossover operation. In the first set, selection of the parents for the crossover is done based on Fitness of the individuals (i.e., the best and the second best individuals are chosen for crossover), the variant henceforth will be referred to as **Fitness_Neuron.** In the second set, the parental selection is done based on the distance vector that is calculated i.e., the worst individual and the individual which differs most with respect to that individual is chosen for cross-over. This variant henceforth will be referred to as **Structure_Neuron.** The third set is combination of the selection methods of set one and two. In these experiments depending on coin-toss selections of the parent could be either based on Fitness of the individuals or based on distance vector that is calculated. This variant will be referred to as **Str_Fitness_Neuron.**

**Category II:** The variants of the modified Minipop where crossover is happening at the boundary of the individual parameters of the neuron are categorized into this group. This group is further divided into 3 different sets depending upon the parental selection for the crossover operation. In the first set, selection of the parents for the crossover is done
based on Fitness of the individuals the variant henceforth will be referred to as 
**Fitness_Parameter_Neuron.** In the second set the parental selection is done based on the distance vector that is calculated. This variant henceforth will be referred to as **Structure_Parameter_Neuron.** The third set is combination of the selection methods of set one and two. In these experiments depending on coin-toss selections, the parent could be either based on Fitness of the individuals or based on distance vector that is calculated. This variant will be referred to as **Str_Fitness_Paramater_Neuron.**
3. CONTROL PROBLEMS

Chapter Overview:

This chapter deals with familiarizing the readers with correction of common arrhythmias problem in a simulated human heart and suppression of oscillation in a jet combustion chamber. The chapter also deals with the parameters that were configured for the control problems for the experimental stage.

Section 3.1 is to describe the correction of common arrhythmias problem in simulated human heart. The section begins with brief description about common arrhythmias problem in human heart and then is followed by explaining the electrical model that is used for simulation of the human heart. Sections 3.1.1 and 3.1.2 explain the CTRNN architecture that is used for this problem and the Fitness function that is used for evaluation. Section 3.2 gives the values of CTRNN and the Evolutionary algorithm parameters that are used.

Section 3.3 describes the suppression of thermo-acoustic oscillations in Jet Combustion Chamber. The section begins with a brief description about thermo-acoustic oscillations and then is followed by explaining about the model that is used for simulation of the thermo-acoustic oscillations. Sections 3.1.1 and 3.1.2 explain the CTRNN architecture that is used for this problem and the Fitness function that is used for evaluation. Section 3.2 gives the values of CTRNN and Evolutionary algorithm parameters that are used.
3.1 Correction of Common Arrhythmias in Simulated Human Heart:

The first evaluation problem will be to evolve the controllers to correct arrhythmias in a simulated human heart. The human heart consists of four chambers as illustrated in Figure 3.1. In a healthy human heart, the left and right atria chambers contract in unison followed shortly by simultaneous contraction of the ventricles. The Sinoatrial (SA) node, which is located in the right atrium of the heart consists of self excitatory tissue that generates regular electrical bursts and generates the heart’s Sinus Rhythm [49]. In a normal heart, the SA node generates 65-85 beats per minute. The SA node impulse travels along the walls of the atria causing the two atria’s to contract in unison [50]. The impulse, in weakened form, also gets relayed to point called Atrio-Ventricular (AV) node on the lower, ventricular portion of the heart. The AV node lies beneath the endocradium of the right atrium, near the inter-ventricular spectrum. AV node is also comprised of self-excitatory tissues. The AV node produces an intrinsic firing frequency that drives the contractions of the ventricles; it however, does entrain to the SA node due to periodic stimulation from the SA node. The SA node fires and starts an electrical impulse, called P wave. The width of the P wave measures the time required for the wave to travel across the atria. The wave next moves down across to AV node where it is delayed by about 0.1 sec before spreading to the walls of the ventricle, causing ventricular contractions. The delay is for ensuring that atria contracts completely. After conducting through the AV node, the wave quickly darts through the ventricles, resulting in sharp up and down waves of the ventricles complex, known as QRS complex (R-Waves).
Cardiac arrhythmias are breakdowns in the normal relationships between atrial and ventricular contractions i.e., in cases where there is an abnormal electrical activity in the heart (P-R wave). Medical assessment (or diagnosis) of the cardiac arrhythmias is done using Electrocardiogram (ECG). Figure 3.2 shows a simplified representation of ECG time series. Arrhythmias can be classified depending upon the heart beat rate, or mechanisms. It can be also be classified into Atrial, Junctional arrhythmias, Atrio-ventricular, Ventricular, Heart blocks depending upon the place of origin of the irregularity. In our case we will be concentrating upon the Heart Block Arrhythmias.
Heart block arrhythmias are commonly referred to as AV blocks, as the vast number of these arise from pathology at the atrio-ventricular node. It is one of the most common causes for Bradycardia. There are three classes AV blocks characterized by specific ECG signatures. A **First degree** AV block or a **PR prolongation** is characterized by a lengthening of the PR interval beyond a certain acceptable interval, generally taken to be 0.2 seconds. A **Second degree** AV block is a conduction block between atrias and ventricles. It can be diagnosed when one or more of the atrial pulses fail to conduct to the ventricles (I.E., it is characterized by the missing of the R-wave after the P-wave). Second degree AV block can be further classified into **Mobitz Type I block and Mobitz Type II block**. Mobitz Type I block (or Wenckebach Block), as shown in figure 3.3, is characterized by continuous increase in the PR interval until one of the R wave is dropped. Mobitz Type II block as shown in figure 3.4 is characterized by constant PR interval in which the R wave is dropped out occasionally. Mobitz Type II block are further subdivided into groups based on the ratio of atrial and ventricular contractions. For example in patient who drops R wave after every 5 P waves would be said to have 5:1 Mobitz Type II block. **Third degree** blocks, also known as complete heart blocks,
are characterized by condition in which the impulse generated in the SA node (atrium) is not propagated into the ventricles. As the impulse is blocked the self-excitatory tissue in the lower chambers will typically activate the ventricles (contractions) i.e., in third degree block as shown in figure 3.5 there is total failure of synchronization between the top and bottom of the heart.

Figure 3.3: Time series plots for an unassisted heart with 2nd degree Mobitz type I block SA node impulse is shown using black thin lines. AV node impulse is shown using bold gray lines.

Figure 3.4: Time series plots for an unassisted heart with 2nd degree Mobitz type II block.
Coupled oscillator model of heart beat generation [52][53] has been adopted for this work. Although this model is simplistic in many ways not and completely accepted by cardiologists as a completely valid explanatory model, it never the less serves as a good descriptive model that possesses interesting dynamics in its own right. In the model, both the AV and the SA nodes are represented by *vanderPol* oscillators (Figure 3.6). The two oscillators share a common coupling "active resistor" (labeled V1 and V2 in Figure) which is capable of both producing and dissipating energy. The value of the resistor R controls the degree of coupling between the oscillators. The value of R can be manipulated to produce all three AV blocks.

*Figure 3.5: Time series plots for an unassisted heart with 3rd block.*
The state equations of the cardiac model as follows

\[ x_1' = \frac{1}{C_1} x_2 \]

\[ x_2' = -\frac{1}{L_1} \left[ x_1 + g(x_2) + R(x_2 + x_4) + S_1 \right] \]

\[ x_3' = \frac{1}{C_2} x_4 \]

\[ x_4' = -\frac{1}{L_2} \left[ x_3 + f(x_4) + R(x_2 + x_4) + S_2 \right] \]

\[ f(x) = V_2(x) = -x + \frac{1}{3} x^3 \]

\[ g(x) = V_1(x) = -x + \frac{1}{3} x^3 + h(x) \]

\[ h(x) = \begin{cases} 
-x^2 - 1/4 & |x| < 1/2 \\
-x & x > 1/2 \\
x & x < -1/2 
\end{cases} \]
Where SA node activation (corresponding to voltage equivalent in electrical model) is represented by $x_2$ and AV node activation is represented by $x_4$. The node currents through SA and AV are represented by $x_1$ and $x_3$ respectively. Simulations are induced into SA and AV nodes through externally controlled voltage sources $S_1$ and $S_2$.

In our experiments, it was assumed that AV node fires at an intrinsical frequency of 40 BPM. To produce realistic AV potentials and also an intrinsic frequency of 40 BPM at AV node the values of $C_2$ and $L_2$ have been chosen as 0.675 and 0.027 respectively. The degree of coupling between SA and AV node is determined by $R$, its value has been chosen as 1.1 to simulate coupling as observed in healthy human heart. The values $C_1$ and $L_1$ reflect the intrinsic frequency of SA node, chosen to produce whole heart intrinsic frequencies of 40 BPM, 60 BPM, 80 BPM, 100 BPM and 120 BPM. The table 3.1 below gives the values of $C_1$ and $L_1$ for different frequencies.

<table>
<thead>
<tr>
<th>Heart Rate</th>
<th>$C_1$</th>
<th>$L_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 BPM</td>
<td>0.395</td>
<td>0.079</td>
</tr>
<tr>
<td>60 BPM</td>
<td>0.250</td>
<td>0.05</td>
</tr>
<tr>
<td>80 BPM</td>
<td>0.15</td>
<td>0.03</td>
</tr>
<tr>
<td>100 BPM</td>
<td>0.144</td>
<td>0.0228</td>
</tr>
<tr>
<td>120 BPM</td>
<td>0.09125</td>
<td>0.01825</td>
</tr>
</tbody>
</table>

Table 3.1 Heart Model Parameters($C_1$ and $L_1$) for different heart rates
3.1.1. CTRNN EH Controller Architecture for Heart Problem

We have chosen 5-node CTRNN architecture as the reconfigurable hardware to evolve the electrical circuit for correcting the common arrhythmias problems in the simulated human heart as shown in the figure 3.7. The output nodes of the oscillator circuit $x_2$ and $x_4$ (simulated heart) are given as sensory input to the CTRNN and two outputs of the CTRNN is given as input to the heart (through $S_1$ and $S_2$).

![CTRNN Architecture Diagram](image)

Figure 3.7: Interface of CTRNN Feedback Controller for correction of common arrhythmias problem.

3.1.2 Fitness Calculation for Heart Problem

Fitness of an individual CTRNN controller generated by the Evolutionary algorithm is evaluated by simulating CTRNN augmented heart for fifteen simulated seconds and evaluating the errors between CTRNN augmented heart behavior and
healthy, non-augmented (normal) heart behavior for ten selected test conditions (heart beat rates from 40BPM to 120BPM ). The total error for each test condition was the mean squared error between the augmented heart and normal heart PR interval, the mean square error between the augmented heart and normal heart RP interval, the mean square error between the augmented heart and normal heart atrial rate, and the mean square error between the augmented heart and normal heart ventricular rate. The total error for a CTRNN controller was taken to be the sum of the errors of each of the test conditions. The figure 3.8 illustrates the calculation of mean square error between healthy and augmented PR interval and RP interval. The equation 3.1 represents the calculation of this mean square error. \( T_{pr}(hr) \) and \( T_{pr}(ar) \) represents the time between R wave and P wave in healthy and augmented human heart respectively. \( T_{pr}(hr) \) and \( T_{pr}(ar) \) represents the time between P wave and R wave in healthy and augmented human heart respectively. The table 3.2 shows the values of \( T_{pr}(hr) \) and \( T_{pr}(hr) \) (PR interval and RP interval for an healthy heart) for different heart beat rates.

\[
E_1 = \frac{1}{N} \left[ \sum \{Trp(hr) - Trp(ar)\}^2 + \sum \{Tpr(hr) - Tpr(ar)\}^2 \right] \quad -- \text{3.1}
\]

Figure 3.8 shows the calculation of MSE between normal and augmented PR interval and mean square error between normal and augmented RP interval.
<table>
<thead>
<tr>
<th>Heart Beat Rate</th>
<th>PR interval in Healthy Heart</th>
<th>PR interval in Healthy Heart</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 BPM</td>
<td>0.631</td>
<td>0.846</td>
</tr>
<tr>
<td>60 BPM</td>
<td>0.435</td>
<td>0.556</td>
</tr>
<tr>
<td>80 BPM</td>
<td>0.336</td>
<td>0.406</td>
</tr>
<tr>
<td>100 BPM</td>
<td>0.278</td>
<td>0.319</td>
</tr>
<tr>
<td>120 BPM</td>
<td>0.239</td>
<td>0.259</td>
</tr>
</tbody>
</table>

Table 3.2 Values of PR interval and RP interval values for healthy heart

Figure 3.9: Calculation of P and R waves over a period of N

\[ E^2 = \frac{1}{N} \left[ (mP(hr) - nP(ag))^2 + (mR(hr) - nR(ag))^2 \right] \]  

----- 3.2

The figure 3.9 along with the equation 3.2, illustrates the calculation of MSE between healthy and augmented arterial rates and ventricular rates. mP(hr) and nP(ag) represents the number of arterial waves over a period of N. mR(hr) and nR(ag) represents the number of ventricular waves over a period of N. (In our case N has been selected as 60
simulated seconds). The total error value is calculated as summation of $E_1$ and $E_2$ for all test conditions consisting of each of the five heart beat rates as defined in table 3.1 under normal condition i.e., $R=1.1$ and under 3rd block condition.

### 3.2 EA and CTRNN Parameters for Heart Model:

For the heart model the Minipop algorithm and the (six) variants of Modified Minipop algorithm have been configured to run with population size of 8, so as to keep the algorithm as compact as possible, for a maximum of 100000 evaluation cycles. The parameters for Minipop and variants of modified Minipop algorithm are listed down in the table 3.3 and table 3.4 lists CTRNN parameters for the Heart Controller.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>8</td>
</tr>
<tr>
<td>Genome Length</td>
<td>320</td>
</tr>
<tr>
<td>Mutation Rate (for Minipop and for variants for modified Minipop)</td>
<td>0.005</td>
</tr>
<tr>
<td>Cross-over Rate (for variants for modified Minipop)</td>
<td>0.05</td>
</tr>
<tr>
<td>Maximum Number of evaluations</td>
<td>100000</td>
</tr>
<tr>
<td>Seed Value</td>
<td>Random</td>
</tr>
</tbody>
</table>

Table 3.3: Minipop and Variants of modified Minipop parameters for Heart Model
Parameter | Value
--- | ---
Number of neurons | 5
Number of Inputs | 2
Number of Outputs | 2
Weight Range | -13.0 to 13.0

Table 3.4 CTRNN parameters for Heart Model

### 3.3 SUPPRESSION OF THE OSCILLATIONS IN JET COMBUSTION CHAMBER:

The second control problem that has been selected for evaluation of the Minipop algorithms and variants is the control of thermo-acoustic instabilities in combustion chamber of simple, non-turbine, jet engine.

Thermo-acoustic instability can potentially arise in the combustion chamber of any jet engine, but predominantly in those designed to run in Lean Pre-mixed conditions. Due to the two-way coupling between the acoustics and the flame dynamics in a combustion chamber [54][55]. The instabilities could result in damaging vibrations and parasitic heat transfers and the loss of propulsion efficiency. Despite these problems LP based jet engine are still desirable as they not only operate at low fuel to air ratio but also produce fewer pollutants. Many power generation systems, propulsion and heating process systems that use lean premixed continuous combustion are prone to thermo-acoustic instabilities. Though passive suppression of the oscillation is possible under certain circumstances, active control of suppression of the oscillations provides far more
flexibility and robustness. Designs of the active controllers for suppression of oscillations rely on accurate modeling of the underlying mechanisms governing combustion dynamics and a good understanding of the tight and often subtle coupling between actuators, combustion dynamics, and acoustics.

The graph in the figure 3.10 illustrates the engine pressure with respect to time for the first 0.04 seconds of uncontrolled operation of unstable desktop combustion chamber. It can be observed from the graph that the pressure amplitude grows exponentially in the jet combustion chamber due to thermo-acoustic instabilities. It can be seen that the pressure amplitude rapidly reaches unsafe levels. The approach for design of the active controller for suppression of the oscillation in LP based jet combustion chamber can be broadly classified into two categories. They are Experimental based and Model based active controllers. In the Experimental based approach a feedback mechanism is selected and experimentally tuned to suppress instabilities by adjusting effectors on the fly. Tuning may be done with respect to a real engine. In the Model based approach, initially an analytical model of the combustor is constructed and studied to find instabilities. Feedback controllers that remove these instabilities are developed via mathematical study. These are constructed and added to the engine.

![Graph](image)

Figure 3.10 Internal Pressure Vs time in uncontrolled jet combustion chamber (where X-axis represents time and Y axis represents pressure in Pascal’s)
A simple combustion chamber with audio speaker, used as control actuators, and microphone, used as control sensor is shown in Figure 3.11. A fuel air mixture introduced through the closed end of the chamber is ignited in the flame holder. Heat and thrust are produced when the combustion products are expelled through the open end. Lean-Premixed fuels contribute to the flame instability in the combustor but they are of particular interest as they release less harmful emissions from the combustor. The flame instabilities might result in a shortened life or might be instrumental in the disruption of the operation of the chamber instantly due to positive feedback between the flame and acoustic vibrations commonly known as TA instability.

A full development of simulation state equations for the propane-fueled combustor based on the figure 3.11 is given in [55]. The four simulated engine configurations, designated EM1, EM2 (EM1 and EM2 configurations are shown in Figure 3.12) which represent speaker end-mount configurations resonant at 357 Hz and 714 Hz and SM1, SM2 (EM1 and EM2 configurations are shown in Figure3.13) which represent side-mount configurations resonant at 542 Hz and 708 Hz respectively. The exponential growth of
the pressure amplitude inside the jet combustion chamber is typical for all the four unstable configurations.

A possible solution to the TA instability problem is the introduction of closed-loop control [56]. For simple combustor model as shown in Figure 3.11, a closed loop controller could be placed that would observe engine vibrations through attached microphone and vibrations are controlled by exciting the speakers effectors. In [31] the closed controller was hand designed Linear Quadratic Regulator (LQR) controller.

![Combustion Chamber with End mounted Speaker](image)

Figure 3.12: Combustion Chamber with End mounted Speaker

<table>
<thead>
<tr>
<th>End Speaker Mount Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM1</td>
</tr>
<tr>
<td>EM2</td>
</tr>
</tbody>
</table>

Table 3.5: Resonant Frequencies for End mount Speaker Configuration
3.3.1 CTRNN-EH Controller Architecture for Jet Combustion Chamber:

We have chosen 5-node CTRNN architecture as the reconfigurable hardware to evolve the controller for controlling the thermo-acoustic oscillations in combustion chamber of jet engine as shown in the figure 3.11. Each neuron in the CTRNN receives a raw microphone value as the sensory input. The output from two neurons controls the amplitude and frequency of the oscillator that is given as input to the speaker.
3.3.2 Fitness Function for Jet Combustion Chamber Problem:

The objective function to be minimized is the area under the amplitude curve of the microphone for one second of simulated time. Ideally, this value would be zero for a combustor that never vibrates and is not possible to achieve in real world.

3.4 EA and CTRNN Parameters for Jet Combustion problem:

For the heart model the Minipop algorithm and the (six) variants of Modified Minipop algorithm have been configured to run with population size of 8, to keep the algorithm as compact as possible, for a maximum of 30000 evaluation cycle. The parameters for Minipop and variants of modified Minipop algorithm is listed in the table 3.7. and table
3.8 lists CTRNN parameters for the Heart Controller. To make a fair assessment of Minipop and its derivatives (modified Minipop variants) each experiment is carried out 40 times.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>8</td>
</tr>
<tr>
<td>Genome Length</td>
<td>320</td>
</tr>
<tr>
<td>Mutation Rate(for Minipop and for variants for modified Minipop)</td>
<td>0.005</td>
</tr>
<tr>
<td>Cross-over Rate(for variants for modified Minipop)</td>
<td>0.05</td>
</tr>
<tr>
<td>Maximum Number of evaluations</td>
<td>100000</td>
</tr>
<tr>
<td>Seed Value</td>
<td>Random</td>
</tr>
</tbody>
</table>

Table 3.7 Minipop and Variants of modified Minipop parameters for Jet Combustion Chamber Problem

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of neurons</td>
<td>5</td>
</tr>
<tr>
<td>Number of Inputs</td>
<td>2</td>
</tr>
<tr>
<td>Number of Outputs</td>
<td>1</td>
</tr>
<tr>
<td>Weight Range</td>
<td>-13.0 to 13.0</td>
</tr>
</tbody>
</table>

Table 3.8 CTRNN Parameters for Jet Combustion Chamber problem
4 Experimental Results

4.1 Experimental Results for Heart Problem:

Figure 4.1 through 4.3 illustrate the correction of common arrhythmias in simulated human heart by one of the evolved controllers which has a fitness score of 1.67. Figure 4.1 shows typical correction of a 2nd degree Mobitz type I block. Figure 4.1a shows time series plots of SA and AV activations over a period of fifteen seconds for a model heart coupled with the evolved controller beating at 60 BPM. The value of R has been changed to 0.09 instead of 1.1, the other parameters being constant as described in section 3.2, to simulate 2nd degree Mobitz type I block.

Figure 4.2 illustrates typical correction of 2nd degree Mobitz Type II 2:1 block. Figure 4.2 illustrates SA and AV activation time series for a damaged heart beating at 100 BPM coupled with evolved controller whose fitness value is 1.65. The value of R has been changed to 0.26 instead of 1.1; the other parameters are kept constant as described in section 3.2, to simulate 2nd degree Mobitz type II 2:1 block.

Figure 4.3 shows typical correction of 3rd degree AV block. The value of R has been changed to 0 instead of 1.1, the other parameters being constant as described in section 3.2 for 120 BPM heart rate, to simulate 3rd degree AV block.
Figure 4.1 Time series plots for an assisted heart with 2nd degree Mobitz type I block (heart coupled with the evolved controller).

Figure 4.2: Time series plots for an assisted heart with 2nd degree Mobitz type II block.
4.2 Algorithm Assessment Parameters for correction of Arrhythmias in Simulated Human Heart.

The performance of each variant algorithm is assessed by two factors

i) **Time to acceptable Solutions (TTA)** TTA is the number of evaluation cycles required for the algorithm to come across an acceptable solution. Based on our prior knowledge of the control problem, coupled with our experiments it has been found that, when the error value of the evolved controller was at less than or equal to 1.65 then evolved controller was stable and acceptable. Hence we have chosen a target error value of 1.65 to calculate TTA.

ii) **Final Fitness Value (FFV)** FFV is determined by allowing the algorithm is allowed to run for predefined number of evaluations before the fitness value is recorded. For this control problem the algorithm is allowed to run for 100,000 evaluation cycles. The FFV is the error score of the best individual after 100,000 evaluation cycles.

Figure 4.3: Time series plots for an assisted heart with 3rd degree block
50 trails of experiments were conducted for each algorithm to compare the performance of the Minipop and variants of modified Minipop on the evolution of controller for correction of common arrhythmias in simulated human heart.

4.2.1 Experimental results Time to Acceptable Solutions:

From the experiments conducted it has been found that the average number of evaluation cycles required for the Minipop algorithm to reach an acceptable solution (solution with a fitness value of 1.65) is 2964. The variants of the modified Minipop algorithm took between 3957 to 6700 evaluation cycles. The results are tabulated in table 4.1. By observing the data, one may infer that Minipop algorithm performs better in terms of time to acceptable solutions compared to variants of modified Minipop. Among the modified Minipop variants Str_Fitness_Neuron does better than the rest. Figure 4.4 shows the box plot for all the variants of the algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Eval. Cycles)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>2964</td>
<td>1601</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>6309</td>
<td>15551</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>4191</td>
<td>6590</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>6700</td>
<td>12416</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>5916</td>
<td>13456</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>6683</td>
<td>11357</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>3957</td>
<td>5023</td>
</tr>
</tbody>
</table>

Table 4.1 Average Time to Acceptable Solution for Heart Control Problem
Figure 4.4: Shows the Box plot for the Time to Acceptable Solutions in Heart problem case. Y-axis represents the evaluation cycles, and X-axis A - Minipop, B - Fitness_Parameter_Neuron, C - Structure_Parameter_Neuron, D - Str_Fitness_Parameter_Neuron, E - Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

To increase our confidence that the Minipop algorithm is better than all variants of modified Minipop in terms of time to acceptable solutions for the heart control problem, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in table 4.2. Even though we anecdotally believe that unadorned Minipop is indeed the best variant for the heart problem in general, from the ANOVA results (P = 0.58), we fail to reject the hypothesis that means of all the algorithms is the same, in other words we
need to conduct experimental trials to be able say anything with any reasonable level of statistical confidence.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>5.3299E+08</td>
<td>8.8832E+07</td>
<td>0.7931</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>2.9570E+10</td>
<td>1.1201E+08</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.0103E+10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: ANOVA results for the TTA case in heart problem.

### 4.2.1 Experimental results for Final Fitness Value:

The table 4.3 shows the final fitness value after 100000 evaluations for all the algorithms along with standard deviation. We observe from the table that the unadorned Minipop algorithm performs better than all variants of modified Minipop. Figure 4.5 shows the box plot for all the variants of the algorithm. To increase our confidence that the Minipop algorithm is better than all variants of modified Minipop in terms of final fitness value for the heart control problem, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in Table 4.4. From the ANOVA results (P = 0.0006) it can be concluded that there is significant difference between the means of the algorithms. To ascertain the means of which algorithm differ with respect to one another T-Tests have been conducted, the results of the T-test are tabulated in Table 4.5. From the T-Test results it can be conclusively said that Minipop algorithm is better than all the crossover variants.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Fitness. Value)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>1.317</td>
<td>0.096</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>1.4227</td>
<td>0.1215</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>1.3686</td>
<td>0.1180</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>1.4014</td>
<td>0.1185</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>1.3903</td>
<td>0.0948</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>1.3831</td>
<td>0.1612</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>1.4318</td>
<td>0.0885</td>
</tr>
</tbody>
</table>

Table 4.3: Average Final Fitness values for different algorithms for the Heart Problem case.

Figure 4.5: Shows the Box plot for the Time to Acceptable Solutions in Heart problem case. Y-axis represents the evaluation cycles, and X-axis represents the algorithms, A - Minipop, B - Fitness_Paramater_Neuron, C - Structure_Parameter_Neuron, D-
Str_Fitness_Paramater_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>0.3385</td>
<td>5.6414E-02</td>
<td>4.132</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>3.605</td>
<td>1.3654E-02</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.943</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.0006

Table 4.4: ANOVA results for the Final Solutions in heart problem

4.3 Experimental Results for Jet Combustion Chamber Problem:

The controllers that are evolved using the Minipop and variants of modified Minipop algorithms are able to suppress the oscillations in Jet Combustion Chamber.

To compare the performance of Minipop and variants of modified Minipop algorithms on the suppression of oscillations in Jet Combustion Chamber, 70 trails were conducted for each algorithm on each of the four engine configuration (SM1, SM2, EM1, and EM2).

4.4 Assessment Parameters for the Jet Combustion Chamber problem:

The performance of the algorithms is assessed by two factors

i) **Time to acceptable Solutions (TTA)** i.e., number of evaluation cycles required for the algorithm to come across an acceptable solution. Based on our prior knowledge of the control problem, coupled with our experiments it has been found that, when the error
<table>
<thead>
<tr>
<th>Minipop</th>
<th>Structure_Parameter_Neuron</th>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
<th>Str_Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.001</td>
<td>P = 0.0012</td>
<td>P = 0.001</td>
<td>P = 0.0299</td>
<td>P = 0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Minipop]</td>
<td>[Minipop]</td>
<td>[Minipop]</td>
<td>[Minipop]</td>
<td>[Minipop]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structure_Parameter_Neuron</th>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
<th>Str_Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.047</td>
<td>P = 0.3674</td>
<td>P = 0.6474</td>
<td>P = 0.0102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Fitness_Parameter_Neuron]</td>
<td>[Structure_Neuron]</td>
<td>[Fitness_Neuron]</td>
<td>[Structure_Parameter_Neuron]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.4436</td>
<td>P = 0.6548</td>
<td>P = 0.8083</td>
<td>P = 0.1047</td>
</tr>
<tr>
<td>[Fitness_Parameter_Neuron]</td>
<td>[Structure_Neuron]</td>
<td>[Fitness_Neuron]</td>
<td>[Fitness_Neuron]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.2306</td>
<td>P = 0.5713</td>
<td>P = 0.0542</td>
</tr>
<tr>
<td>[Str_Fitness_Parameter_Neuron]</td>
<td>[Fitness_Neuron]</td>
<td>[Structure_Neuron]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 0.1585</td>
<td>P = 0.1047</td>
</tr>
<tr>
<td>[Structure_Neuron]</td>
<td>[Fitness_Neuron]</td>
</tr>
</tbody>
</table>

Indicates the Statistically significant (Confidence level 95 %)

Indicates the Statistically significant (Confidence level 90 %)

[ ] --------------- Winner of the two algorithms

Table 4.5: T-Test for Final solutions for correction of common arrhythmias problem in simulated human heart.
value of the evolved controller was at less than or equal to 400 then evolved controller was stable and acceptable. Hence we have chosen an error value of 400 to calculate TTA.

ii) **Final Fitness Value** (FFV) , In this case the algorithm is allowed to run for predefined number of evaluations before the fitness value is recorded. For this control problem the algorithm is allowed to run for 30,000 evaluation cycles.

### 4.4.1 Performance assessment in EM1 configuration:

#### 4.4.1.2 Experimental results for Time to acceptable Solutions (TTA)

From the experiments conducted it has been found that average number of evaluation cycles required for the Minipop algorithm to reach an acceptable solution (solution with a fitness value of 400) is 686 evaluation cycles. The variants of the modified Minipop algorithm took between 633 to 1388 cycles. The results are tabulated in the table 4.6. By observing the data, one may infer that **Structure_Neuron** algorithm performs slightly better in terms of TTA compared to Minipop and variants of modified Minipop. Minipop algorithm fairs better with respect to rest of the algorithms. Figure 4.6 shows the box plot for all the variants of the algorithm.

To increase our confidence that that the **Structure_Neuron** is better than all variants of modified Minipop for time to acceptable solution in case of Jet Combustion chamber problem in EM1 configuration, we conducted ANOVA (Analysis of Variance) test on population distributions tabulated in table 4.7. From the ANOVA results (P = 0.042) it can be concluded that there is significant difference between the means of the algorithms. To ascertain the means of which algorithm differ with respect to one another T-Tests have been conducted, the results of the T-test are tabulated in Table 4.8. From the T-Test
results it can be conclusively said that Minipop differs statistically only with Str_Fitness_Neuron, Str_Fitness_Parameter differs statistically with Str_Fitness_Neuron and Structure_Neuron differs statistically with Str_Fitness_Neuron. Even though we believe that Structure_Neuron is better than Minipop and other variants of modified Minipop, we need to run more experiments to be able to say that with reasonable degree of statistical confidence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Eval. Cycles)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>686</td>
<td>638</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>778</td>
<td>807</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>942</td>
<td>1092</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>728</td>
<td>596</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>633</td>
<td>186</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>1047</td>
<td>175</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>1388</td>
<td>1630</td>
</tr>
</tbody>
</table>

Table 4.6: TTA results for different algorithms for Jet Combustion Chamber in EM1 configuration
Figure 4.6: Shows the Box plot for the Time to Acceptable Solutions in Jet Combustion Chamber in EM1 configuration case. Y-axis represents the evaluation cycles, and X-axis represents the algorithms A - Minipop, B - Structure_Parameter_Neuron, C – Fitness_Parameter_Neuron, D - Str_Fitness_Parameter_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>1.4927E+07</td>
<td>2.4878E+06</td>
<td>2.211</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>2.9147E+08</td>
<td>1.1254E+06</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.0640E+08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.042

Table 4.7: ANOVA results for the TTA in case of Jet Combustion Chamber in EM1 configuration.
<table>
<thead>
<tr>
<th>T-Test</th>
<th>Structure_Parameter_Neuron</th>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
<th>Str_Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>P = 0.5799 [Minipop]</td>
<td>P = 0.217 [Minipop]</td>
<td>P = 0.7632 [Minipop]</td>
<td>P = 0.7300 [Minipop]</td>
<td>P = 0.4378 [Minipop]</td>
<td>P = 0.0169 [Minipop]</td>
</tr>
<tr>
<td>Structure_Parameter_Neuron</td>
<td>P = 0.4608 [Structure_Parameter_Neuron]</td>
<td>P = 0.7599 [Str_Fitness_Parameter_Neuron]</td>
<td>P = 0.3989 [Structure_Neuron]</td>
<td>P = 0.6867 [Structure_Parameter_Neuron]</td>
<td>P = 0.0435 [Structure_Parameter_Neuron]</td>
<td></td>
</tr>
<tr>
<td>Fitness_Parameter_Neuron</td>
<td>P = 0.2905 [Str_Fitness_Parameter_Neuron]</td>
<td>P = .1442 [Structure_Neuron]</td>
<td>P = 0.8577 [Fitness_Parameter_Neuron]</td>
<td>P = 0.1653 [Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str_Fitness_Parameter_Neuron</td>
<td>P = 0.5181 [Structure_Neuron]</td>
<td>P = 0.5361 [Str_Fitness_Parameter_Neuron]</td>
<td>P = 0.0234 [Str_Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>P = 0.3346 [Structure_Neuron]</td>
<td>P = 0.0112 [Structure_Neuron]</td>
<td>P = 0.1639 [Fitness_Neuron]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Indicates the Statistically significant (Confidence level 95 %)

[ ] --------------- Winner of the two algorithms

Table 4. 8: T-Test for Time to acceptable solutions, for suppression of oscillations in Jet Combustion Chamber in EM1 Configuration.
4.4.1.2 Final Fitness Value:

The table 4.9 shows the final fitness value after 30000 evaluations for all the algorithms along with standard deviation. The results show that Minipop has an average fitness of 181.46 and the average of the fitness of variants of modified Minipop ranges from 175.85 to 194.21. Fitness_Neuron performs better in terms of Final Fitness averages compared to rest. Figure 4.7 shows the box plot for all the variants of the algorithm. To increase our confidence that that the Fitness_Neuron algorithm is better than all variants of modified Minipop in terms of final fitness value for jet combustion chamber in EM1 configuration, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in table 4.10. Even though we anecdotally believe that Fitness_Neuron is indeed the best variant for the heart problem in general, from the ANOVA results (P = 0.37), we fail to reject the hypothesis that means of all the algorithms is the same, in other words we need to conduct experimental trails to be able say anything with any reasonable level of statistical confidence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Fitness. Value)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>181.46</td>
<td>32.1</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>188.2</td>
<td>42.2</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>183.8</td>
<td>45.2</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>194.21</td>
<td>46.608</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>186.67</td>
<td>47.2</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>175.85</td>
<td>40.53832</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>189.99</td>
<td>47.355</td>
</tr>
</tbody>
</table>

Table 4.9: Final Fitness values for different algorithms for EM1 configuration
Figure 4.7: Shows the Box plot for the Final Solutions in Jet Combustion Chamber in EM1 configuration case. Y-axis represents the evaluation cycles, and X-axis represents the algorithms, A - Minipop, B - Structure_Paramater_Neuron, C – Fitness_Parameter_Neuron, D - Str_Fitness_Paramater_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>1.2264E+04</td>
<td>2044.0</td>
<td>1.095</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>5.5430E+05</td>
<td>1866.</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>5.6657E+05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.37

Table 4.10: ANOVA results for the Final Solutions in case of Jet Combustion Chamber in EM1 configuration.
4.4.2 Performance assessment in EM2 configuration:

4.4.2.1 Time to Acceptable Solutions:

From the experiments conducted it has been found that average number of evaluation cycles required for the Minipop algorithm to reach an acceptable solution (solution with a fitness value of 400) is 632 evaluation cycles. The variants of the modified Minipop algorithm took between 675 to 1388 cycles. The results are tabulated in the table 4.11. So by looking at these experiments, one may infer that Minipop algorithm performs slightly better in terms of TTA compared to variants of modified Minipop. Figure 4.8 shows the box plot for all the variants of the algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Eval. Cycles)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>632</td>
<td>668.17</td>
</tr>
<tr>
<td>Structure_Parameter_neuron</td>
<td>675</td>
<td>483.3</td>
</tr>
<tr>
<td>Fitness_Parameter_neuron</td>
<td>915</td>
<td>1257.4</td>
</tr>
<tr>
<td>Str_Fitness_Parameter_neuron</td>
<td>1388</td>
<td>1630.554</td>
</tr>
<tr>
<td>Structure_neuron</td>
<td>763</td>
<td>1032.455</td>
</tr>
<tr>
<td>Fitness_neuron</td>
<td>1047</td>
<td>1239.08</td>
</tr>
<tr>
<td>Str_Fitness_neuron</td>
<td>1017</td>
<td>1384.132</td>
</tr>
</tbody>
</table>

Table 4.11: TTA results for different algorithms in EM2 configuration.

To increase our confidence that the Minipop algorithm is better than all variants of modified Minipop in terms of time to acceptable solutions for jet combustion chamber problem in EM2 configuration, we conducted ANOVA (Analysis of Variance) test on
population distributions, tabulated in table 4.12. Even though we anecdotally believe that \textbf{Minipop} is indeed the best variant for the heart problem in general, from the ANOVA results ($P = 0.24$), we fail to reject the hypothesis that means of all the algorithms is the same, in other words we need to conduct experimental trails to be able say anything with any reasonable level of statistical confidence.

![Figure 4.8: Shows the Box plot for the Time to Acceptable Solutions in Jet Combustion Chamber in EM2 configuration case. Y-axis represents the evaluation cycles, and X-axis represents the algorithms A - Minipop, B - Structure_Parameter_Neuron, C – Fitness_Parameter_Neuron, D- Str_Fitness_Parameter_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.](image-url)
### Table 4.12: ANOVA results for the TTA in case of Jet Combustion Chamber in EM2 configuration.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>1.1125E+07</td>
<td>1.8541E+06</td>
<td>1.331</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>3.6075E+08</td>
<td>1.3928E+06</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.7187E+08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.24

#### 4.4.2.2 Final Fitness Value:

The table 4.13 shows the final fitness value after 30000 evaluations for all the algorithms along with standard deviation. The observation of the results shows that Minipop performs better than variants of modified Minipop with fitness value of 162.97. Among the crossover variants Fitness_Parameter_Neuron is better with fitness value of 171.33. Figure 4.9 shows the box plot for all the variants of the algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Fitness. Value)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>162.97</td>
<td>36.700</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>172.073</td>
<td>38.74</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>171.33</td>
<td>36.55</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>189.99</td>
<td>47.355</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>181.423</td>
<td>47.3683</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>175.85</td>
<td>40.53832</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>178.23</td>
<td>42.321</td>
</tr>
</tbody>
</table>

Table 4.13: Final Fitness values for different algorithms in EM2 Configuration
To increase our confidence that that the Minipop algorithm is better than all variants of modified Minipop in terms of final solutions for jet combustion chamber problem in EM2 configuration, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in table 4.14. Even though we anecdotally believe that unadorned Minipop is indeed the best variant for the heart problem in general, from the ANOVA results \( P = 0.60 \), we fail to reject the hypothesis that means of all the algorithms is the same, in other words we need to conduct experimental trails to be able say anything with any reasonable level of statistical confidence.

![Figure 4.9: Shows the Box plot for the Final Solutions in Jet Combustion Chamber in EM2 configuration case. Y-axis represents the evaluation cycles, and X-axis A - Minipop, B - Structure_Parameter_Neuron, C – Fitness_Parameter_Neuron, D - Str_Fitness_Parameter_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.](image-url)
<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>7915.0</td>
<td>1319.0</td>
<td>0.760</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>4.4955E+5</td>
<td>1736.0</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.5746E+05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.60

Table 4.14: ANOVA results for the Final Solutions in case of Jet Combustion Chamber in EM2 configuration.

### 4.4.3 Performance assessment in SM1 configuration:

#### 4.4.3.1 Time to Acceptable Solutions:

From the experiments conducted it has been found that average number of evaluation cycles required for the Minipop algorithm to reach an acceptable solution (solution with a fitness value of 400) is 501 evaluation cycles. The variants of the modified Minipop algorithm took between 653 to 1104 cycles. The results are tabulated in the table 4.15. So by looking at these experiments, one may infer that Minipop algorithm performs better in terms of TTA compared to variants of modified Minipop. Among the modified Minipop variants Fitness_Neuron does better than the rest. Figure 4.10 shows the box plot for all the variants of the algorithm.

To increase our confidence that that the Minipop algorithm is better than all variants of modified Minipop for time to acceptable solutions for jet combustion chamber in SM1 mode, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in 4.16. Even though we anecdotally believe that Minipop is indeed the best variant for the heart problem in general, from the ANOVA results (P = 0.24), we fail to reject the hypothesis that means of all the algorithms is the same, in other
words we need to conduct experimental trails to be able say anything with any reasonable level of statistical confidence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Eval. Cycles)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>501</td>
<td>493.433</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>1104</td>
<td>1319.78</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>653</td>
<td>840.26</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>768</td>
<td>1091.173</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>812</td>
<td>913.13</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>659</td>
<td>840.26</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>965</td>
<td>1061.2</td>
</tr>
</tbody>
</table>

Table 4.15: TTA results for different algorithms for Jet combustion chamber in SM1 configuration
Figure 4.10: Shows the Box plot for the Time to Acceptable Solutions in Jet Combustion Chamber in EM2 configuration case. Y-axis represents the evaluation cycles, and X-axis A - Minipop, B - Structure_Parmaeter_Neuron, C – Fitness_Parameter_Neuron, D- Str_Fitness_Parmaeter_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>1.2226E+07</td>
<td>2.0376E+06</td>
<td>1.817</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>2.9039E+08</td>
<td>1.1212E+06</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.0262E+08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.096

Table 4.16: ANOVA results for the TTA in case of Jet Combustion Chamber in EM2 configuration.
4.4.3.2 Final Fitness Value:

The table 4.17 shows the final fitness value after 30000 evaluations for all the algorithms along with standard deviation. The observation of the results it may be inferred that Minipop algorithm performs better than the variants of modified Minipop (although there is not much of difference between Minipop and Fitness_Neuron). Figure 4.11 shows the box plot for all the variants of the algorithm. To increase our confidence that that the **Minipop** algorithm is better than all variants of modified Minipop in terms of final solutions for jet combustion chamber problem in SM1 configuration, we conducted ANOVA (Analysis of Variance) test on population distributions. Even though we anecdotally believe that **Minipop** is indeed the better than cross-over variants for the heart problem in general, from the ANOVA results (P = 0.79), we fail to reject the hypothesis that means of all the algorithms is the same, in other words we need to conduct experimental trails to be able say anything with any reasonable level of statistical confidence.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Fitness. Value)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>161.77</td>
<td>28.69</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>171.64</td>
<td>50.023</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>170.298</td>
<td>43.54</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>173.458</td>
<td>52.413</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>172.9155</td>
<td>40.694</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>162.59</td>
<td>36.06</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>175.200</td>
<td>47.48</td>
</tr>
</tbody>
</table>

Table 4.17: Final Fitness values for different algorithms in SM1 configuration
Figure 4.11: Shows the Box plot for the Final Solutions in Jet Combustion Chamber in SM1 configuration case. Y-axis represents the evaluation cycles, and X-axis represents algorithm, A - Minipop, B - Structure Parameter Neuron, C - Fitness Parameter Neuron, D - Str_Fitness_Parameter_Neuron, E - Structure Neuron, F - Fitness Neuron, and G - Str_Fitness Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>5989.0</td>
<td>998.2</td>
<td>0.5272</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>4.9036E+05</td>
<td>1893.0</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.9635E+05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.79

Table 4.18: ANOVA results for the Final Solutions in case of Jet Combustion Chamber in SM1 configuration.
4.4.4 Performance assessment in SM2 configuration:

4.4.4.1 Time to Acceptable Solutions:

From the experiments conducted it has been found that average number of evaluation cycles required for the Minipop algorithm to reach an acceptable solution (solution with a fitness value of 400) is 745 evaluation cycles. The variants of the modified Minipop algorithm took between 653 to 1272 cycles. From the results tabulated in the table 4.19 one can observe that **Fitness_Neuron** algorithm performs slightly better in terms of TTA compared to others. Figure 4.12 gives the box plot for the population distribution of all the algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Eval. Cycles)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>745</td>
<td>485.713</td>
</tr>
<tr>
<td>Structure_Paramater_Neuron</td>
<td>969</td>
<td>865.561</td>
</tr>
<tr>
<td>Fitness_Paramater_Neuron</td>
<td>1099</td>
<td>1325.498</td>
</tr>
<tr>
<td>Str_Fitness_Paramater_Neuron</td>
<td>1036</td>
<td>1265.565</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>812</td>
<td>913.138</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>653</td>
<td>840.260</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>1273</td>
<td>1421.56</td>
</tr>
</tbody>
</table>

Table 4.19: TTA results for different algorithms, for Jet Combustion Chamber problem in SM2 configuration.

To increase our confidence that that the **Fitness_Neuron** algorithm is better than all other variants of modified Minipop in terms of time to acceptable solutions for the Jet Combustion Chamber problem in SM2 configuration, we conducted ANOVA (Analysis
of Variance) test on population distributions, tabulated in 4.20. From the ANOVA results ($P = 0.026$) it can be concluded that there is significant difference between the means of the algorithms. To ascertain the means of which algorithm differ with respect to one another, T-Tests have been conducted, the results of the T-test are tabulated in Table 4.21. From the t-test results it has been observed that we need to run more trails in order to be able to say that Fitness_Neuron is better variant for the average time to acceptable solutions for the jet combustion chamber problem in SM2 configuration with any reasonable level of statistical confidence.

![Box plot](image)

Figure 4.12: Shows the Box plot for the Time to Acceptable Solutions in Jet Combustion Chamber in EM2 configuration case. Y-axis represents the evaluation cycles, and X-axis represents algorithms, A - Minipop, B - Structure_Parameter_Neuron, C -
Fitness_Parameter_Neuron, D- Str_Fitness_Paramater_Neuron, E – Structure_Neuron, 
F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>2.1786E+07</td>
<td>3.6310E+06</td>
<td>2.441</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>3.8531E+08</td>
<td>1.4877E+06</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.0709E+08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.026

Table 4.20: ANOVA results for the TTA in case of Jet Combustion Chamber in SM2 configuration.

4.4.4.2 Final Fitness Value:

The table 4.22 shows the final fitness value after 30000 evaluations for all the algorithms along with standard deviation. The observation of the results shows Fitness_Neuron performs a bit better than Minipop and other variants. Figure 4.13 shows the box diagram of the population distribution for all the algorithms.

To increase our confidence that that the Fitness_Neuron algorithm is better than all other variants of modified Minipop in terms of final solution for the Jet Combustion Chamber problem in SM2 configuration, we conducted ANOVA (Analysis of Variance) test on population distributions, tabulated in 4.23. From the ANOVA results (P = 0.021) it can be concluded that there is significant difference between the means of the algorithms. To ascertain the means of which algorithm differ with respect to one another, T-Tests have been conducted, results of the T-test are tabulated in Table 4.24. From the T-tests it can be statistically said that Minipop algorithm is better than Structure_Neuron, Str_Fitness_Neuron and Str_Fitness_Paramater_Neuron, Fitness_Neuron is better than Structure_Neuron, Str_Fitness_Parameter_Neuron & Str_Fitness_Neuron and Structure_Neuron is better than Str_Fitness_Paramater_Neuron. For the rest number of
<table>
<thead>
<tr>
<th>T-Test</th>
<th>Structure_Parameter_Neuron</th>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
<th>Str_Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>P = 0.2958 [Minipop]</td>
<td>P = 0.2645 [Minipop]</td>
<td>P = 0.2336 [Minipop]</td>
<td>P = 0.1725 [Minipop]</td>
<td>P = 0.9133 [Fitness_Neuron]</td>
<td>P = 0.0264 [Minipop]</td>
</tr>
<tr>
<td>Structure_Parameter_Neuron</td>
<td>P = 0.9743 [Structure_Parameter_Neuron]</td>
<td>P = 0.8778 [Structure_Parameter_Neuron]</td>
<td>P = 0.9036 [Structure_Neuron]</td>
<td>P = 0.3688 [Fitness_Neuron]</td>
<td>P = 0.8773 [Structure_Parameter_Neuron]</td>
<td></td>
</tr>
<tr>
<td>Fitness_Parameter_Neuron</td>
<td>P = 0.8457 [Fitness_Parameter_Neuron]</td>
<td>P = 0.8675 [Structure_Neuron]</td>
<td>P = 0.3458 [Fitness_Neuron]</td>
<td>P = 0.8431 [Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str_Fitness_Parameter_Neuron</td>
<td>P = 0.9599 [Structure_Neuron]</td>
<td>P = 0.2968 [Fitness_Neuron]</td>
<td>P = 0.9952 [Str_Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>P = 0.2457 [Fitness_Neuron]</td>
<td>P = 0.9631 [Structure_Neuron]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>P = 0.2724 [Fitness_Neuron]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Indicates the Statistically significant (Confidence level 95%)
- Indicates the Statistically significant (Confidence level 90%)

[ ] --------------- Winner of the two algorithms

Table 4.21: T-Test for Time acceptable solutions for suppression of oscillations in Jet Combustion Chamber in SM2 Configuration.
experiments carried out not significant enough to make any conclusive statements about the performance of the algorithms with respect one another.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average (Fitness Value)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>170.323</td>
<td>29.286</td>
</tr>
<tr>
<td>Structure_Parametern</td>
<td>185.398</td>
<td>32.8636</td>
</tr>
<tr>
<td>Fitness_Parametern</td>
<td>170.0791</td>
<td>38.13</td>
</tr>
<tr>
<td>Str_Fitness_Parametern</td>
<td>191.102</td>
<td>42.585</td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>172.916</td>
<td>40.694</td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td>162.593</td>
<td>36.06</td>
</tr>
<tr>
<td>Str_Fitness_Neuron</td>
<td>196.315</td>
<td>45.002</td>
</tr>
</tbody>
</table>

Table 4.22: Final Fitness values for different algorithms in SM2 configuration
Figure 4.13: Shows the Box plot for the Final Solutions in Jet Combustion Chamber in SM1 configuration case. Y-axis represents the evaluation cycles, and X-axis represents, A - Minipop, B - Structure_Parameter_Neuron, C – Fitness_Parameter_Neuron, D - Str_Fitness_Parameter_Neuron, E – Structure_Neuron, F - Fitness_Neuron, G - Str_Fitness_Neuron.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Mean Squares</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatments (between columns)</td>
<td>2.4050E+04</td>
<td>4008.0</td>
<td>2.539</td>
</tr>
<tr>
<td>Residuals (within columns)</td>
<td>4.0896E+05</td>
<td>1579.</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.3301E+05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.021

Table 4.23: ANOVA results for the Final Solutions in case of Jet Combustion Chamber in EM2 configuration.
<table>
<thead>
<tr>
<th>T-Test</th>
<th>Structure_Parameter_Neuron</th>
<th>Fitness_Parameter_Neuron</th>
<th>Str_Fitness_Parameter_Neuron</th>
<th>Structure_Neuron</th>
<th>Fitness_Neuron</th>
<th>Str_Fitness_Neuron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minipop</td>
<td>P = 0.0381 [Minipop]</td>
<td>P = 0.9752 [Minipop]</td>
<td>P = 0.0158 [Minipop]</td>
<td>P = 0.6345 [Minipop]</td>
<td>P = 0.2152 [Minipop]</td>
<td>P = 0.0040 [Minipop]</td>
</tr>
<tr>
<td>Structure_Parameter_Neuron</td>
<td>P = 0.0647 [Fitness_Parameter_Neuron]</td>
<td>P = 0.5155 [Structure_Parameter_Neuron]</td>
<td>P = 0.2138 [Structure_Neuron]</td>
<td>P = 0.6332 [Fitness_Neuron]</td>
<td>P = 0.2314 [Structure_Parameter_Neuron]</td>
<td></td>
</tr>
<tr>
<td>Fitness_Parameter_Neuron</td>
<td>P = 0.0263 [Fitness_Parameter_Neuron]</td>
<td>P = 0.6479 [Fitness_Parameter_Neuron]</td>
<td>P = 0.2486 [Fitness_Neuron]</td>
<td>P = 0.0077 [Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str_Fitness_Parameter_Neuron</td>
<td>P = 0.0929 [Structure_Neuron]</td>
<td>P = 0.3194 [Fitness_Neuron]</td>
<td>P = 0.6056 [Str_Fitness_Parameter_Neuron]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure_Neuron</td>
<td>P = 0.5033 [Fitness_Neuron]</td>
<td>P = 0.0332 [Structure_Neuron]</td>
<td>P = 0.1420 [Fitness_Neuron]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fitness_Neuron</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indicates the Statistically significant (Confidence level 95 %)**

**Indicates the Statistically significant (Confidence level 90 %)**

[ ] --------------- Winner of the two algorithms

Table 4.20: T-Test for Final Solutions for suppression of oscillations in Jet Combustion chamber in SM2 Configuration.
5. Results, Discussion and Future Work

This chapter contains an overview of the results presented in chapter 4, including a discussion on the results as well as directions for future work.

5.1 Results Summary and Discussion:

The results presented in the previous chapter illustrates that variants of Minipop algorithm have not be been able to improve the performance compared to Minipop in most cases for both Time to acceptable solutions and Final solutions( although this cannot be backed statistically for most cases for the presented control problems). Table 5.1 shows the winner for the Time to acceptable solutions for both the control problems without statistical considerations. From the table it can be seen that Minipop performs better than the proposed modified Minipop algorithm for some of the configurations and for some other configurations modified Minipop algorithm is better.

<table>
<thead>
<tr>
<th>Control Problem</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction of common arrhythmias in simulated human heart</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in EM1 configuration</td>
<td>Structure_Neuron</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in EM2 configuration</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in SM1 configuration</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in SM2 configuration</td>
<td>Fitness_Neuron</td>
</tr>
</tbody>
</table>

Table 5.1 Winner for Time to acceptable solutions for the two control problems
Table 5.2 shows the winner for final solutions for both the control problems without statistical considerations. From the table 5.1 and 5.2 it can be seen that variant of modified Minipop algorithm is better than Minipop algorithm for Suppression of Oscillations in Jet combustion chamber in EM1 and SM2 configuration (cannot be statistically corroborated).

<table>
<thead>
<tr>
<th>Control Problem</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction of common arrhythmias in simulated human heart</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in EM1 configuration</td>
<td>Fitness_Neuron</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in EM2 configuration</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in SM1 configuration</td>
<td>Minipop</td>
</tr>
<tr>
<td>Suppression of oscillations in Jet Combustion Chamber in SM2 configuration</td>
<td>Fitness_Neuron</td>
</tr>
</tbody>
</table>

Table 5.2 Winner for Final Solutions for the two control problems

By observing the results it leads us to believe that there may not be modularity in the architecture of the evolved CTRNN controllers for these control problems (in some configuration) i.e., the control problems that have been chosen for the purpose of evaluation may be better suited for mutation driven Minipop than cross-over variant modified Minipop algorithm. Another reason for Minipop performing fairly better than crossover variant may be because of the assumption that the parameters that were good
for Minipop for a control problem is good for the modified Minipop algorithm which may not be the case in general.

It was also seen that the most of the results cannot be backed statistically i.e., number of experiments is less than sufficient to make valid conclusion. Theoretically additional experiments can be carried out but the fact of availability of limited resources is also to be considered. It was observed that evolution of controller for correction of common arrhythmias took on average 50 hours for each experiment (on Intel Pentium 4 based 2.0GHz PC) i.e., close 14000 hrs to run all the experiments. For evolution of controller for suppression of oscillation in Jet combustion chamber took on average 4hrs i.e, close to 6720 hrs to complete all experiments.

5.2 Future Work

An area of interest that might be worth the pursuit would be to modify the crossover probability as function of exponential decaying function instead of static values. This is because crossover operator has a tendency of convergence, so applying crossover operator as function of exponential decaying function might help in avoiding premature convergence. Another way of modifying the crossover probability is to represent it as a function of square pulse instead of static values (where the crossover operator is switched on and off alternatively over a certain number of evaluation cycles). When the crossover operator is switched on for a certain duration, period crossover operator might help in convergence of the solutions and when the crossover operator is switched off, other evolutionary operators (Mutation and Hyper mutation) might help in diversifying the solutions. These changes might be worth pursuing to answer the
questions of whether crossover operator is really helpful for CTRNN control problems and at what stage in the evolutionary algorithm is the crossover operator really helpful? Another aspect that is worth experimenting is evaluating modified Minipop algorithm for other control problems like evolution of controller for six-legged insect. This might help us in understanding the type of problems that this algorithm can be better applied to (as applications might differ in the amount of modularity in their CTRNN architecture), as in evolutionary algorithms there is no free lunch i.e., algorithm which is good for one set of problems may not necessarily be good for different set of problems. These proposed extensions to the thesis work provide an opportunity to further understand the type of control problems where the crossover operator might be helpful and at what stage in evolutionary algorithms the crossover operator is really useful.
Bibliography


50. Phibbs, Brendan. (2007) The human heart, a basic guide to heart disease, 2nd Ed


