

Optimization of Ethanol Production Process from Sugarcane Bagasse using Genetic Algorithm

By:

Jacob Chol Garang

15797

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

JANUARY 2015

Universiti Teknologi PETRONAS
32610 Bandar Seri Iskandar,
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

**Optimization of Ethanol Production Process from Sugarcane
Bagasse using Genetic Algorithm**

By:

Jacob Chol Garang

15797

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

Approved by:

Dr. Timothy Ganesan

**UNIVERSITI TEKNOLOGI PETRONAS
BANDAR SERI ISKANDAR, PERAK**

January

2015

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

JACOB CHOL GARANG

ABSTRACT

The search for new technologies for converting plant biomass into alternative biofuels is leveraged by many social and environmental problems associated with the use of fossil fuels and their exploration. Bioethanol has emerged as a potential alternative for fossil fuels. Bioethanol production from lignocellulosic biomass obtained from sugarcane bagasse using *Saccharomyces cerevisiae* comprises the hydrolysis of cellulose and sugar fermentation. This research was focused on the optimization of ethanol production from sugarcane bagasse using genetic algorithm (GA). The relationship between the process variables were modelled by Response Surface Methodology (RSM) and found that the maximum ethanol concentration to be 17.44 g/l. GA was employed to find the optimal values for process variables that gives the maximum ethanol concentration. It was found that the optimal values of the variables are 20.14g/l for substrate concentration, 4.5 for pH and 24hours for fermentation time and 0.15 g/l for Na_2HPO_4 . These optimal variables lead to an improved ethanol concentration of 47.85g/l which is 174% increase from the predicted concentration obtained by empirical testing.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my beloved supervisor Dr Timothy Ganesan, who has shown the attitude and the substance of a genius. He continually and persuasively conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to educating. Without his supervision and constant willingness to help, this research would not have been accomplished.

Deepest gratitude goes to my beloved family, Relatives and friends whose support came to me in many different forms. Special thank goes to my beloved mother Rabecca Athiek Kon who has entirely exceeded expectations for a wonderful mother. Her words of encouragement and support throughout my academic and entire life are priceless, for all this work belongs to her.

Most importantly, I would like to thank Almighty God for his protection and guidance throughout this Final year project Thesis. The Almighty father has been assuring me health and peace of mind throughout the period. He has been showing the directions and crucial areas for this project. This is the greatest contribution of all and I am thankful for his mercy. For without him, I wouldn't have moved this mountain.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	i
CERTIFICATE OF ORIGINALITY.....	ii
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction	1
1.2 Genetic Algorithms/Computational intelligence introduction	2
1.3 Problem statement	2
1.4 Project Objectives	3
1.4.1 The objectives for this project are	3
1.4.2 Scope of study	3
CHAPTER 2: LITERATURE REVIEW.....	4
Conclusions	9
CHAPTER 3: RESEARCH METHODOLOGY	10
3.1 Key Milestone for FYP I	10
3.2 Gantt chart for FYP I.....	11
3.3 Key Milestones for FYP II	11
3.4 Gantt chart for FYP II.....	12
3.5 Project flow chart	13
3.6.0 Tools used.....	13
3.6.1 Genetic Algorithm (GA) Toolbox for optimization in Mat lab	14
3.7 Optimization Procedures for Genetic Algorithm	14
3.7.1 Function declaration	14
3.7.2 Constraints and Boundary declaration	15
3.7.3 Genetic algorithm execution from the command window.....	16
3.7.4 Calling GA optimization tool box from the command	16
3.80 Genetic Algorithm Parameters	17
3.90 Genetic Algorithm Parameters Selection	23
3.10.0 Genetic Algorithm Parameters Tuning.....	24
3.10.1 Genetic algorithm parameter screening	24
3.10.2 Selection of significant parameters.....	24
3.10.3 Best Parameters Tuning	24
3.11 Overall Methodological Flowchart	25

CHAPTER 4: RESULTS AND DISCUSSION	26
4.0 Data collection.....	26
4.1 Genetic Algorithm Parameter Screening.....	26
4.2 Best, mean and worst function values selection.....	29
4.3 Best Parameters tuning	32
4.4 Plots of results from the optimization toolbox	34
CHAPTER 5: CONCLUSIONS & RECOMMENDATIONS	36
REFERENCES	
APPENDICES	

LIST OF FIGURES

Figure 1: Pareto chart of Plackett Burman design (Diptarka, <i>et al.</i> , 2013).....	5
Figure 2: Optimization conditions for maximizing ethanol yield predicted by reliasoft DOE (Diptarka, <i>et al.</i> , 2013).....	7
Figure 3 shows process flow of GA (Marco, <i>et al.</i> , 2012).....	14
Figure 4 shows declaration of m-file on matlab.....	15
Figure 5 shows declaration of boundaries on matlab.....	16
Figure 6 shows execution of the GA from the command window	16
Figure 7 shows calling for GA Optimization tool box from the Command window	16
Figure 8 shows GA Optimization window toolbox display.....	17
Figure 9 shows overall Methodology flowchart	25
Figure 10 shows the optimization results display on the tool box	27
Figure 11: shows the graph of best function values for different initial ranges and fitness limit.....	29
Figure 12: shows the graph of Mean function values against initial ranges	30
Figure 13: shows the graph of worst function values against fitness limits	31
Figure 14: Function value against crossover fraction	33
Figure 15: Best fitness function against generation	34
Figure 16: Best individual variable (1=substrate concentration, 2=pH, 3 = fermentation time and 4 = Na ₂ HPO ₄)	34

LIST OF TABLES

Table 1: Box behnken design (Diptarka, <i>et al.</i> , 2013).	6
Table 2: Significance of term coefficients for BBD (Diptarka, <i>et al.</i> , 2013).....	7
Table 3 shows the default parameters of the GA	26
Table 4 shows the parameters table with specified initial range and default fitness limit	27
Table 5: Parameters table with specified initial ranges and fitness limits	28
Table 6: The best function values for different initial ranges and fitness limits.....	29
Table 7: Mean function value for different initial ranges and fitness limits.....	30
Table 8: The worst function value for different initial ranges and fitness limits.....	31
Table 9: Table for the best parameters of the GA.....	32
Table 10: Crossover fraction against best initial range and fitness limit against.....	33

CHAPTER 1: INTRODUCTION

1.1 Introduction

Dependence on petroleum remains the most important factor affecting the distribution of wealth in the entire world. The deteriorating environmental quality and world's conflicts are a result of petroleum fuels. The increasing world's population and fuel demand as well as restriction on environment by regulations have taken a toll in the research and development of renewable energy feedstock to substitute for and/or to complement fossil fuel sources (Vargas *et al.*, 2009).

The increase in the world's energy demand and continuous depletion of oil reserves has ignited the search for alternative energy resources. These alternative energy resources are derived from renewable materials such as biomass.

Ferreira, *et al.* (2009) suggests that the search for new technologies for converting plant biomass into alternative biofuels is intensified by many social and environmental problems associated with the use of fossil fuels and their exploration. Fermentation-derived ethanol can be produced from sugar, starch or lignocellulosic biomass. Sugar and starch based feed stocks are currently predominant at the industrial level and they are so far economically favourable.

The need for alternative energy sources came about also as a result of changing climatic conditions and consequent need to reduce greenhouse emission of gases. These have raised the demand for bioethanol as a perfect replacement for gasoline.

The global scenario shows that biggest portion of research in last three decades have been focused on technological development for bioethanol after its emergence as a potential fuel additive (Dasgupta *et al.*, 2013).

These have called for the ways to increase production of renewable fuels such as ethanol in order to meet the demand. Optimization of ethanol production from available quantity of sugarcane bagasse plays an important role. Process variables such as temperature, pH, fermentation time and many more are optimized to maximize ethanol production.

Ethanol production via fermentation route comprises of a series of biochemical reactions with numerous factors involved in the process (Dasgupta, 2013). Conversion of lignocellulosic sugar hydrolysate into ethanol requires many other micro and macro elements apart from fermentable nitrogen which in right balance can always give optimum product yield.

1.2 Genetic Algorithms/Computational intelligence introduction

Genetic algorithms are optimization algorithms that are used to find the optimal solutions to a given computational problem either to maximize or to minimize a particular function. These algorithms belong to evolutionary computation branch field of study. Genetic Algorithms search for fittest solution by imitating the biological processes of reproduction and natural selection (Abimbola and Josiah, 2011).

The crucial factors for the conversion of carbohydrates to ethanol are the cost and substrate availability. Consequentially it is worthy to develop an economical process which allows the use of cheap substrates for successive conversion to ethanol. Hence, there is still need for cutting edge research to be done on an effective, economical, and efficient conversion process (Walia *et al.*, 2014).

1.3 Problem statement

Optimization of bioprocesses has involved several statistical experimental methods recently. Many scientific approaches have been employed on several occasions to improve and optimize processes.

Optimization is conventionally carried out heuristically or empirically to find the optimum function values but most of the times it's not attained due to unknown parametric interaction, the presence of multiple local maxima, or non linearities. In the real-world situation, optimization using this method may not be as viable as thought due to competing objectives, complexities and variation in raw materials.

Genetic algorithms are powerful and efficient tools than random search and exhaustive search algorithms that require no extra information about the given problem and this feature enables them to find solutions to the problems of non-linearities, competing objectives and variations in raw materials.

1.4 Project Objectives

1.4.1 The objectives for this project are

- To identify important process variables that affects the production process of ethanol from sugarcane bagasse.
- To optimize the production of ethanol from sugarcane bagasse by optimization of process variables
- To find the interaction effects of process variables on the objective function value using genetic algorithm.
- To tune the genetic algorithm parameters that work best for ethanol production optimization

1.4.2 Scope of study

The scope of this project is to study the optimization of ethanol production from sugarcane bagasse. This study involves the implementation of Genetic Algorithm for optimization of ethanol production. The optimum process variables for ethanol production are identified by tuning the parameters of the GA. Comparative analyses and comprehensive discussion are carried out based on the computational results obtained in this work and the work of Diptarka *et al.* (2013).

CHAPTER 2: LITERATURE REVIEW

According to Maria *et al.* (2013), lignocellulosic biomass which is used for bioethanol production using *Saccharomyces cerevisiae* comprises the hydrolysis of cellulose and sugar fermentation. The cellulose can be saccharified using either acids or enzymes so as to obtain fermentable sugars from sugarcane bagasse.

The crucial factors for the conversion of carbohydrates to ethanol are the cost and substrate availability as stated by Navpreet, *et al.* (2014). Cellulosic substances appeared to be striking as raw materials for some time but as of today are not competitive as substrates for ethanol production. Starchy materials have been therefore proposed and have proved as viable for production of ethanol. Optimization conditions of substrate concentration is a key to success of conversion of starch into sugars as well as the availability of highly active enzymes, suitable strain and, temperature and pH (Navpreet, *et al.*, 2014).

Navpreet (2014) suggests that Response Surface Methodology (RSM) is a major process optimization tool that is used to determine the optimum values of a variety of factors significant for the process. He continued that it is a compilation of statistical techniques to design experiments, evaluate the effects of variables and thereby, seeking the optimum conditions. It is widely used in optimization of different types of fermentations and bioprocesses. The robustness of RSM is the confined sets of experimental runs that are required to provide sufficient information for statistically acceptable results, in addition, its suitability for multiple factor experiments and examination of common relationship between various factors under experiment towards finding the most suitable production conditions for the bioprocess and forecast response. Therefore, it is a group of techniques that are used to study the reaction between one or more measured dependent factors (responses) and a number of input (independent) factors.

Diptarka *et al.* (2013) carried out Respond Surface Methodology (RSM) study of ethanol fermentation with thermotolerant yeast *Kluyveromyces* to find optimum conditions for maximizing ethanol production by two step approach. Plackett-Burman Design (PBD) was the first approach which was used for initial factors

screening to sort out important parameters dictating the yield of ethanol. The second approach was the Box-Behnken Design (BBD) technique which was used to determine to what degree based on their interaction effect.

PBD identified the major variables out of the selected ones using Pareto chart as illustrated in Figure 1. Factors such as pH, fermentation time, substrate concentration and Na_2HPO_4 with T values above threshold and P values lower than 0.05 as represented by regression analysis had a substantial effect on ethanol yield and were considered for further evaluation by BBD, while rest of the variables did not have a significant contribution to ethanol production.

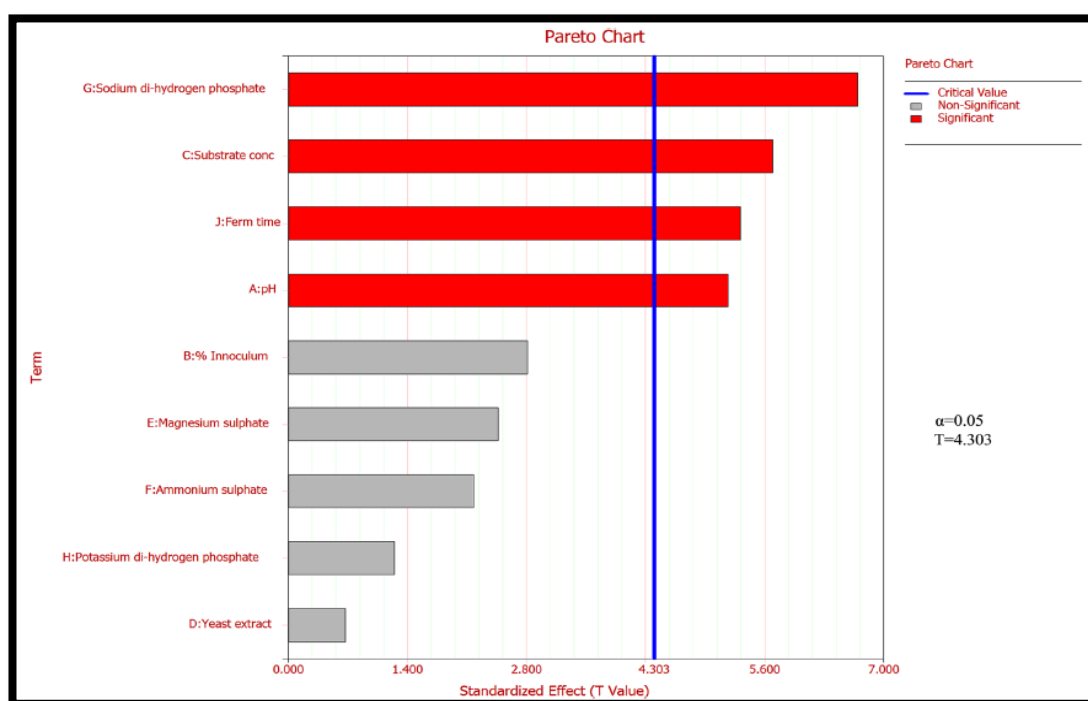


Figure 1: Pareto chart of Plackett Burman design (Diptarka, *et al.*, 2013).

BBD matrix with response is shown in Table 1. A second order polynomial model fit to the experimental data for optimizing ethanol production via response surface method (RSM) predicts response as a function of four variables and their interactions in terms of their coded values.

Table 1: Box behnken design (Diptarka, *et al.*, 2013).

Run order	Random	Substrate conc. (g/L) (A)	pH (B)	Fermentation time (h) (C)	Na ₂ HPO ₄ (g/L) (D)	Experimental ethanol conc. (g/L)	Model predicted ethanol conc. (g/L)
1	15	0	-1	0	+1	10.90	10.32
2	14	0	+1	0	-1	8.60	7.79
3	9	-1	0	-1	0	4.20	3.95
4	12	+1	0	+1	0	11.55	12.07
5	4	+1	+1	0	0	7.10	6.91
6	17	-1	0	0	-1	9.30	9.34
7	19	-1	0	0	+1	4.40	4.35
8	20	+1	0	0	+1	11.50	11.47
9	24	0	+1	+1	0	7.90	7.45
10	7	0	0	-1	+1	7.25	7.37
11	25	0	0	0	0	7.60	8.01
12	11	0	0	+1	0	8.32	7.65
13	2	+1	-1	0	0	14.60	13.53
14	13	0	-1	0	-1	12.70	12.61
15	8	0	0	+1	+1	8.00	8.45
16	10	+1	0	-1	0	8.10	8.37
17	23	0	-1	+1	0	12.00	12.27
18	3	-1	+1	0	0	3.60	4.29
19	1	-1	-1	0	0	7.50	7.31
20	21	0	-1	-1	0	7.90	8.57
21	5	0	0	-1	-1	6.80	7.04
22	6	0	0	+1	-1	12.80	13.36
23	18	+1	0	0	-1	11.00	11.06
24	26	0	0	0	0	8.00	8.01
25	16	0	+1	0	+1	5.40	5.50

Diptarka, *et al.* (2013), used student's t-test and p test values to determine the statistical significance of the model term coefficients as illustrated in Table 2. It was observed that main effects were significant for each of four coded factors whereas interactions among pH, substrate concentration fermentation time & Na₂HPO₄ concentration were important as indicated by their high T and low P values.

ANOVA analysis of the model was performed to evaluate statistical significance of the coefficients of four variables as in table 2. The resulting model is as shown in equation 1 below.

$$\begin{aligned}
 \text{Response} = & 7.67 + 2.21 \times A - 2.40 \times B + 1.81 \times C - 1.14 \times D - 0.90 \times A \times B \\
 & - 0.1675 \times A \times C + 1.35 \times A \times D - 0.07 \times B \times C - 0.35 \times B \times D \\
 & - 1.31 \times C \times D + 0.19 \times A^2 + 0.34 \times B^2 - 0.08 \times C^2 \\
 & + 1.13 \times D^2 \dots \dots \dots \text{equation 1}
 \end{aligned}$$

Where, A, B, C and D are the independent variables which represent Substrate concentration, pH, fermentation time and Na₂HPO₄ respectively (Diptarka, *et al.*, 2013).

Table 2: Significance of term coefficients for BBD (Diptarka, *et al.*, 2013)

Term	coefficient	Standard error	T-value	P-value
Intercept	7.7667	0.3196	24.3025	1.42E-11
A	2.2108	0.1598	13.8358	9.75E-09
B	-2.4083	0.1598	-15.0717	3.68E-09
C	1.8517	0.1598	11.588	7.13E-08
D	-1.1458	0.1598	-7.1708	1.13E-05
A*B	-0.9	0.2768	-3.2518	0.0069
A*C	-0.1675	0.2768	-0.6052	0.55663
A*D	1.35	0.2768	4.8778	0.0004
B*C	-0.075	0.2768	-0.271	0.791
B*D	-0.35	0.2768	-1.2646	0.23
C*D	-1.3125	0.2768	-4.7423	0.0005
A²	0.1996	0.2397	0.8327	0.4213
B²	0.3408	0.2397	1.422	0.1805
C²	-0.0817	0.2397	-0.3407	0.7392
D²	1.1346	0.2397	4.7336	0.0005

Using the polynomial model, maximization study for ethanol production was done and the maximum concentration of ethanol predicted by the model was 17.4g/l. The variables were 40g/l for substrate concentration, pH 4.5, 48h fermentation time and 0.15g/l for Na₂HPO₄ (Figure 2 below). The data was further validated in a shake flask study where the experiment was carried out under optimized condition (Diptarka, *et al.*, 2013).

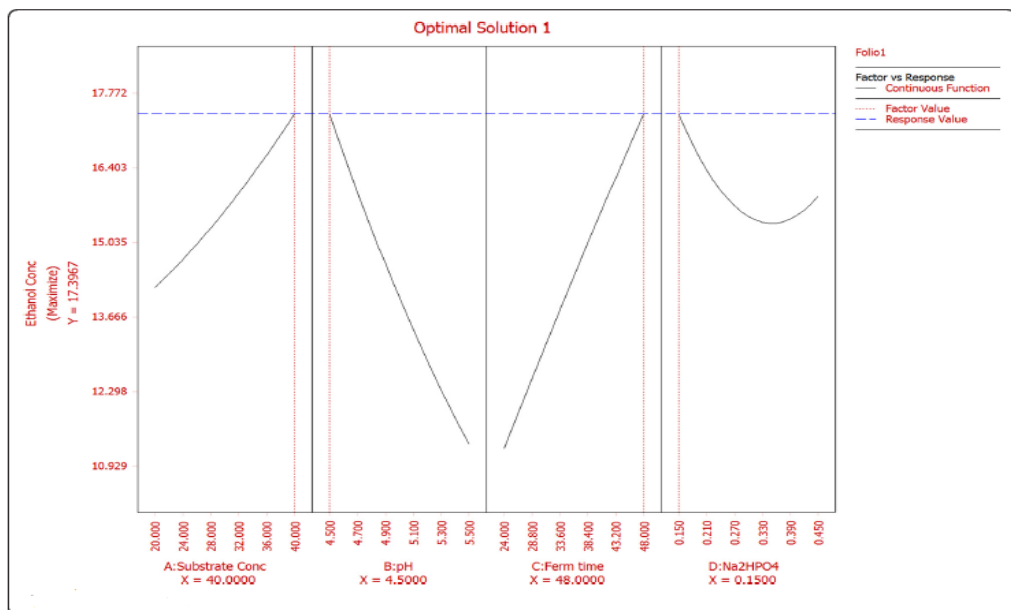


Figure 2: Optimization conditions for maximizing ethanol yield predicted by reliasoft DOE (Diptarka, *et al.*, 2013).

However, the modelled response predicted by Respond Surface Methodology (RSM) has got some limitations in certain situations due to unknown system parameters, the presence of multiple local maxima and non-differentiable systems. In the real-world situation, optimization using this method may not be as viable as thought due to competing objectives, complexities and variation in raw materials. Variety of optimization problems is not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, stochastic, or highly nonlinear.

Abimbola and Josiah (2011) assert that many real-world problems have multiple often competing objectives. The optimization of food processing operations may not be an easy task due to complexities and variations in the raw materials. Evolutionary algorithms as a class of direct search algorithms have proved to be an important tool for difficult search and optimization problems and have received increased interest during the last decade due to the ease way of handling multiple objective problems.

A constrained optimization problem or an unconstrained multi-objective problem may in principle be two different ways to pose the same underlying problem and can be solved by Evolutionary algorithms (Abimbola and Josiah, 2011). Evolutionary algorithms are of interest to finding solution to real world problems because they are proving robust in delivering global optimal solutions which help in resolving limitations encountered in traditional methods.

The GA operates over a population of individuals also called chromosomes that represent possible solutions to the investigated problem. At the beginning, a population of individuals is randomly generated and evolved to one solution by means of applying of GA operators (selection, crossover and mutation). In each generation, the chromosomes are evaluated and the most able ones among them are selected and crossed each other, generating new chromosomes better than those ones of the previous population. It makes to increase the probability of one or more individuals are a solution of the problem. The GA operates over the population in parallel, yielding various solutions at a time. Hence, this method has been used to solve many problems involving complex combinatorial optimization (Marco, *et al.*, 2012).

According to Chipperfield and Fleming (n.d.), GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce successively better approximations to a solution. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from which they were created, just as in natural adaptation.

Mahdavi *et al.* (2009) proposed a GA approach for solving a cell formation problem in cellular manufacturing by developing a model to solve the real size problems, which involve nonlinear terms and integer variables.

Conclusions

In conclusion, Respond Surface Methodology (RSM) has suggested a model that represents the response of various process variables like Substrate concentration, pH, fermentation time and Na_2HPO_4 to the ethanol production as seen in equation 1. Therefore, Genetic Algorithm was used to find optimum solutions for maximum production of ethanol as it has been dubbed to be a powerful tool in cases where there are competing objectives, complexities and variation in raw materials.

CHAPTER 3: RESEARCH METHODOLOGY

In this chapter, the details of the methodological framework of this project are presented. The validity of the study of this project is judged and the steps as well as the procedures under taken on the way in fulfilment of the research objectives are presented. There were basically two important questions answered in this section which are how the data was collected or generated and how the generated data was analysed.

With regards to this section, Genetic Algorithm was the optimization toolbox that was used to generate solutions for the maximization of ethanol production. The results obtained from the GA were tabulated and analysed in the results section. Therefore, the procedures employed to reach the solution to the problem are illustrated in this section.

3.1 Key Milestone for FYP I

No	Activity	Week
1	Submission of the extended proposal	Week 7
2	Proposal Defence (oral presentation)	Week 8,9
3	Submission of Interim Draft Report	Week 13
4	Submission of Interim Report	Week 14

3.2 Gantt chart for FYP I

		Weeks													
No.	Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project title finalizing														
2	Project Introduction														
3	Literature review and methodology														
4	Extended Proposal Submission														
6	Proposal defense														
9	Draft Interim Report Submission														
10	Interim Report submission														

3.3 Key Milestones for FYP II

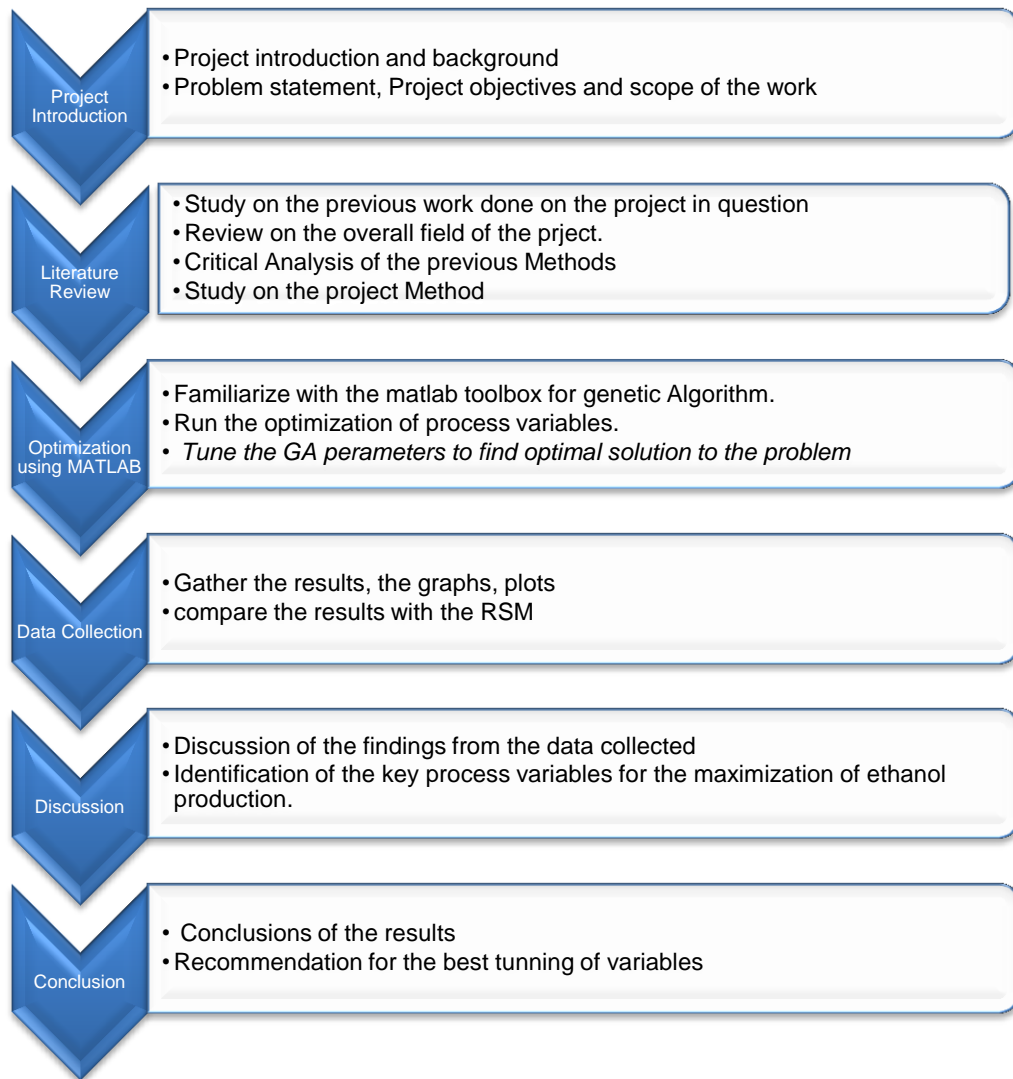
No.	Activity	Week
1	Submission of Progress Report	Week 8
2	Pre Sedex	Week 11
3	Submission of Draft Report	Week 12
4	Submission of Project Dissertation (Soft Bound)	Week 13
5	Submission of Technical Paper	Week 13
6	Oral Presentation	Week 14
7	Submission of Project Dissertation (Hard Bound)	Week 15

3.4 Gantt chart for FYP II

Activity	Week no.														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project Work Continues	Briefing		S1		S2		S3								
Submission of Progress Report								6th March							
Project Work Continues									S4		S5				
Pre-edex											25th March				
Submission of draft report												3rd April			
Project work continuous: Analysis and reporting													S6		
Submission of soft bound														10th April	
Submission of technical paper														10 April	
Oral presentation															15th April
Submission of hard bound															24th April

S- Seminar

3.5 Project flow chart



3.6.0 Tools used

The major tool that was used in this project is the MATLAB which is optimization and simulation software. MATLAB simply means matrix laboratory, a fourth generation of programming language and a multi-paradigm numerical computing environment. It was developed by Math work and allows matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, such as C++, C, Java, Fortran and Python.

3.6.1 Genetic Algorithm (GA) Toolbox for optimization in Mat lab

Genetic algorithm is a stochastic optimization technique that searches for an optimal value of a complex objective function and are used to solve complicated optimization problems by simulation or mimicking a natural evolution process (Abimbola and Josiah, 2011) It involves repeated procedures with an initial population of potential solutions, a fitness evaluation via the application of genetic operators and the development of a new population.

Abimbola and Josiah (2011) continued that GA has been successfully used as a tool in computer programming, artificial intelligence, optimization and neural network training and information technology since its introduction by Holland (1975) to improve the performance of simple GA.

The Figure 3 shows the process flow of GA

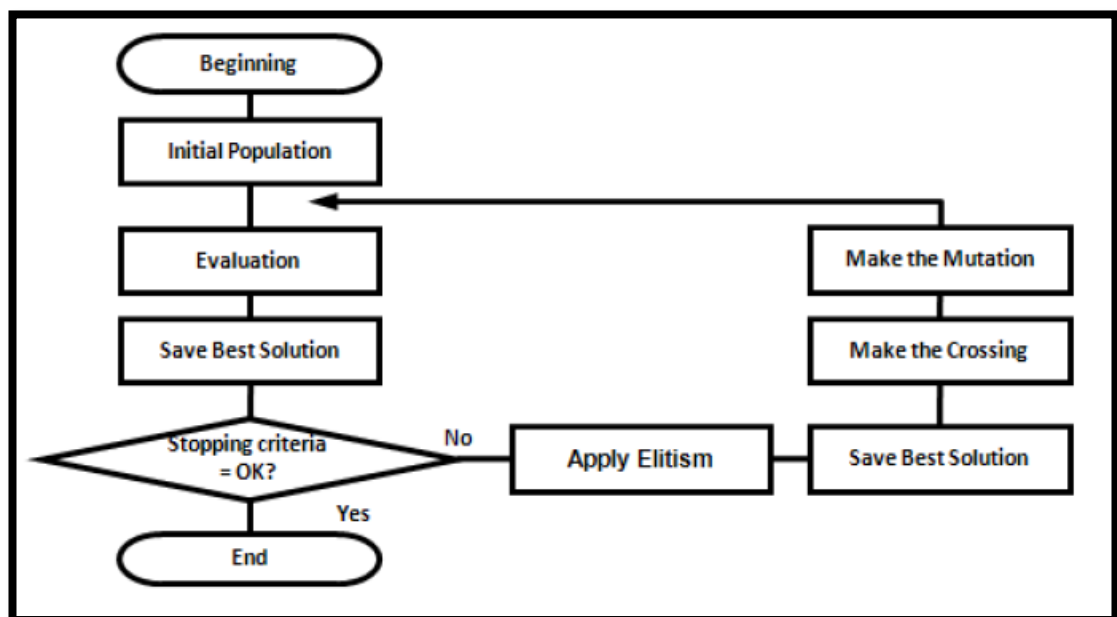


Figure 3 shows process flow of GA (Marco, et al., 2012).

3.7 Optimization Procedures for Genetic Algorithm

3.7.1 Function declaration

The objective function was first declared on the matlab so as to be solved by the toolbox to find optimal solution. This was done by the following steps.

Given the function as

$$y = 7.67 + 2.21 \times x_1 - 2.40 \times x_2 + 1.81 \times x_3 - 1.14 \times x_4 - 0.90 \times x_1 \times x_2 - 0.1675 \times x_1 \times x_3 + 1.35 \times x_1 \times x_4 - 0.07 \times x_2 \times x_3 - 0.35 \times x_2 \times x_4 - 1.31 \times x_3 \times x_4 + 0.19 \times x_1^2 + 0.34 \times x_2^2 - 0.08 \times x_3^2 + 1.13 \times x_4^2$$

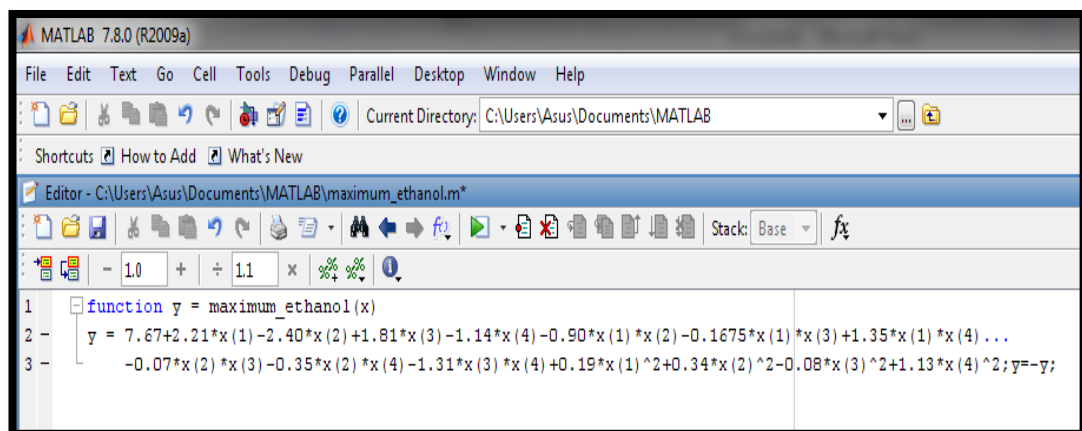
The same function or equation above is written as below in matlab

```
y=7.67+2.21*x(1)-2.40*x(2)+1.81*x(3)-1.14*x(4)-0.90*x(1)*x(2)-
0.1675*x(1)*x(3)+1.35*x(1)*x(4)-0.07*x(2)*x(3)-0.35*x(2)*x(4)-
1.31*x(3)*x(4)+0.19*x(1)^2+0.34*x(2)^2-0.08*x(3)^2+1.13*x(4)^2
```

Where x (1), x (2), x (3) and x (4) are A, B, C and D and are the independent variables which represent Substrate concentration, pH, fermentation time and Na₂HPO₄ respectively.

x = vector of variables

It was therefore declared in a new script and save as m-file with the name “maximum_ethanol” as in the Figure 4 below.



```
function y = maximum_ethanol(x)
y = 7.67+2.21*x(1)-2.40*x(2)+1.81*x(3)-1.14*x(4)-0.90*x(1)*x(2)-0.1675*x(1)*x(3)+1.35*x(1)*x(4)...
-0.07*x(2)*x(3)-0.35*x(2)*x(4)-1.31*x(3)*x(4)+0.19*x(1)^2+0.34*x(2)^2-0.08*x(3)^2+1.13*x(4)^2; y=-y;
```

Figure 4 shows declaration of m-file on matlab

3.7.2 Constraints and Boundary declaration

The constraints and boundaries were also declared the same way as the objective function as below.


```

Editor - C:\Users\Asus\Documents\MATLAB\lower_upperboundaries.m*
1 function = lower_upperboundaries(x)
2 c = [];
3 ceq = [];

```

Figure 5 shows declaration of boundaries on matlab

3.7.3 Genetic algorithm execution from the command window

In order to insert the boundaries and execute the GA, the following syntax was written in the command window.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
fx >> ObjectiveFunction = @maximum_ethanol;
nvars = 4; % Number of variables
lb = [20 4.5 24 0.15]; % Lower bound
ub = [40 5.5 48 0.45]; % Upper bound
ConstraintFunction = @lower_upperboundaries;
[x, fval] = ga(ObjectiveFunction, nvars, [], [], [], [], lb, ub)

```

Figure 6 shows execution of the GA from the command window

Where,

$[x, fval] = ga(\text{ObjectiveFunction}, nvars, [], [], [], [], LB, UB, \text{ConstraintFunction})$

$x = ga(\text{fitnessfcn}, nvars, A, b, Aeq, beq, LB, UB)$ defines a set of lower and upper bounds on the design variables, x , so that a solution is found in the range $lb \leq x \leq ub$. (Set $Aeq = []$ and $beq = []$ if no linear equalities exist.) $nvars$ = number of variables

3.7.4 Calling GA optimization tool box from the command

Ga optimization toolbox was called from the command window as below

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
fx >> optimtool

```

Figure 7 shows calling for GA Optimization tool box from the Command window

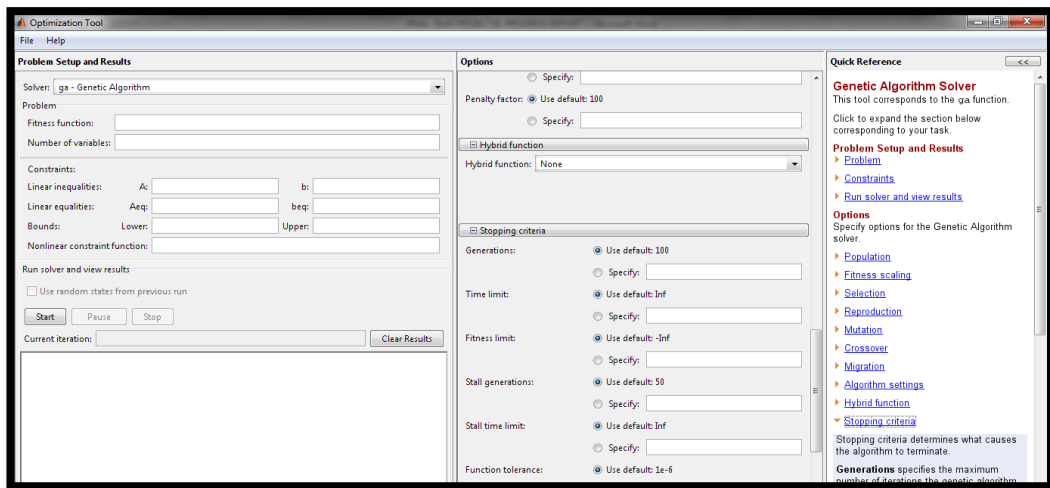


Figure 8 shows GA Optimization window toolbox display

3.80 Genetic Algorithm Parameters

There are several parameters of genetic algorithm that are manipulated in order to get the best performance of the software however, not all the parameters are significant enough to affect the results. There were basically few GA parameters that were tuned in this project which are as explained below.

Population options: this specifies options for the population of the genetic algorithm.

Population type

This specifies the type of the input to the fitness function. The population type can be set to be double vector or Bit string, or Custom. If custom is selected, creation, mutation, and crossover functions that work with the selected population type must be written. These functions must be specified in the fields Creation function, mutation function and Crossover function respectively.

Population size

This specifies how many individuals are there in each generation. If Population size is set to be a vector of length greater than 1, the algorithm creates multiple subpopulations. Each entry of the vector specifies the size of a subpopulation.

Creation function

This specifies the function that creates the initial population. The constraint dependent default chooses:

- Uniform if there are no constraints
- Feasible population otherwise

Uniform creates a random initial population with a uniform distribution.

Feasible population creates a random initial population that satisfies the bounds and linear constraints.

Initial population

This specifies an initial population for the genetic algorithm. The default value is [], in which case GA uses the default Creation function to create an initial population. If a nonempty array in the Initial population field is entered, the array must have no more than Population size rows, and exactly Number of variables columns. In this case, the genetic algorithm calls a Creation function to generate the remaining individuals, if required.

Initial scores

This specifies initial scores for the initial population. The initial scores can also be partial. Do not specify initial scores with integer problems because GA overrides any choice you make.

Initial range

This specifies the range of the vectors in the initial population that is generated by the GA creation uniform creation function. The Initial range is set to be a matrix with two rows and Number of variables columns, each column of which has the form [lb;ub], where lb is the lower bound and ub is the upper bound for the entries in that coordinate. If Initial range is specified to be a 2-by-1 vector, each entry is expanded to a constant row of length Number of variables. If an initial range is not specified, the default is [-10; 10] ([-1e4+1; 1e4+1] for integer-constrained problems), modified to match any existing bounds.

Fitness scaling

The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. Scaling function specifies the function that performs the scaling. The following scaling functions can be chosen from:

Rank

Rank scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

Proportional

This makes the expectation proportional to the raw fitness score. This strategy has weaknesses when raw scores are not in a "good" range.

Top

This scales the individuals with the highest fitness values equally. If this option is selected, quantity can be specified, the number of fittest individuals that produce offspring. Quantity must be an integer between 1 and Population Size or a fraction between 0 and 1 specifying a fraction of the population size. Each of these individuals has an equal probability of reproducing.

Shift linear

This scales the raw scores so that the expectation of the fittest individual is equal to a constant, which you can specify as Maximum survival rate, multiplied by the average score.

Selection Function

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function.

The function that performs the selection in the Selection function field is specified. The following functions can be chosen:

Stochastic; This lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

Remainder; This assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part.

Uniform; this selects parents at random from a uniform distribution using the expectations and number of parents. This results in an undirected search. Uniform selection is not a useful search strategy, but you can use it to test the genetic algorithm.

Roulette; this simulates a roulette wheel with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability equal to its area.

Tournament; this selects each parent by choosing individuals at random, the number of which you can specify by Tournament size, and then choosing the best individual out of that set to be a parent.

Reproduction option

Reproduction options determine how the genetic algorithm creates children at each new generation.

Elite count; specifies the number of individuals that are guaranteed to survive to the next generation. Set Elite count to be a positive integer less than or equal to Population size.

Crossover fraction; specifies the fraction of the next generation that crossover produces. Mutation produces the remaining individuals in the next generation. Crossover fraction can be set to be a fraction between 0 and 1 either by entering the fraction in the text box, or by moving the slider.

Mutation functions

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the genetic algorithm to search a broader space.

Specify the function that performs the mutation in the Mutation function field. The following functions can be chosen from:

Use constraint dependent default chooses:

- Gaussian if there are no constraints
- Adaptive feasible otherwise

Gaussian adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centred on zero. The standard deviation of this distribution can be controlled with two parameters. The Scale parameter determines the standard deviation at the first generation. The Shrink parameter controls how standard deviation shrinks as generations go by. If the Shrink parameter is 0, the standard deviation is constant. If the Shrink parameter is 1, the standard deviation shrinks to 0 linearly as the last generation is reached.

Uniform is a two-step process. First, the algorithm selects a fraction of the vector entries of an individual for mutation, where each entry has the same probability as the mutation rate of being mutated. In the second step, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry.

Adaptive feasible randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. A step length is chosen along each direction so that linear constraints and bounds are satisfied.

Crossover Options

Crossover options specify how the genetic algorithm combines two individuals, or parents, to form a crossover child for the next generation. Crossover function specifies the function that performs the crossover. The following functions can be chosen from:

Scattered; this is the default crossover function for problems without linear constraints, creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second

parent, and combines the genes to form the child. For example, if p1 and p2 are the parents

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8] and the binary vector is [1 1 0 0 1 0 0 0], the function returns the following child:

child1 = [a b 3 4 e 6 7 8]

Stopping Criteria

- i. Fitness limit: The algorithm stops if the best fitness value is less than or equal to the value of Fitness limit.
- ii. Stall generations: the algorithm stops if the average relative change in the best fitness function value over Stall generations is less than or equal to Function tolerance.
- iii. Stall time limit: The algorithm stops if there is no improvement in the best fitness value for an interval of time in seconds specified by Stall time limit, as measured by CPU time.
- iv. Function tolerance: The algorithm stops if the average relative change in the best fitness function value over Stall generations is less than or equal to Function tolerance.

Selection Options

Selection options specify how the genetic algorithm chooses parents for the next generation. The function the algorithm uses can be specified in the Selection function field in the Selection options pane. The options are;

- i. Stochastic uniform: The default selection function, stochastic uniform, lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.
- ii. Remainder: Remainder selection assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part. For example, if the scaled value of an individual is 2.3, that individual is listed twice as a parent because the integer part is 2. After parents have been assigned according to the integer parts of

the scaled values, the rest of the parents are chosen stochastically. The probability that a parent is chosen in this step is proportional to the fractional part of its scaled value.

- iii. Uniform: Uniform selection chooses parents using the expectations and number of parents. Uniform selection is useful for debugging and testing, but is not a very effective search strategy.
- iv. Roulette: Roulette selection chooses parents by simulating a roulette wheel, in which the area of the section of the wheel corresponding to an individual is proportional to the individual's expectation. The algorithm uses a random number to select one of the sections with a probability equal to its area.
- v. Tournament: Tournament selection chooses each parent by choosing Tournament size players at random and then choosing the best individual out of that set to be a parent. Tournament size must be at least 2.

3.90 Genetic Algorithm Parameters Selection

The following parameters of genetic algorithm were randomly selected during the initial execution in the GA optimization toolbox before they were tuned.

- ❖ Population type: Double
- ❖ Population size: 20
- ❖ Creation function: constraint dependent
- ❖ Initial population: Default []
- ❖ Initial scores: Default []
- ❖ Initial Range: [0; 1]
- ❖ Fitness scaling: Rank
- ❖ Selection function: Stochastic
- ❖ Reproduction Option: Elite count with default value of 2 and default crossover fraction of 0.8
- ❖ Mutation function: Adaptive feasible
- ❖ Fitness limit

3.10.0 Genetic Algorithm Parameters Tuning

The following section 3.10.0 explains the tuning of the GA parameters that significantly influence the performance of the algorithm towards optimization of the objective function value.

3.10.1 Genetic algorithm parameter screening

This was done firstly by running the Optimization when the GA parameters were maintained at their default states, then by specifying the initial range and fitness limit to observe their significance.

Three sets of simulations were done to observe the best possible conditions and parameter settings that the GA works. All the other parameters were kept at default except the three below.

- i. Default initial range at default fitness limit
- ii. Specified initial ranges against default fitness limits
- iii. Specified initial ranges against fitness limits

These two parameters were varied to improve the performance of the software, Five initial ranges of [10; 30], [10; 20], and [15; 30], [15; 20] and [15; 21] were varied with three fitness limits of 50, 55 and 60 each and then run for four times. Crossover fraction was kept constant at 0.4 for all the simulation.

3.10.2 Selection of significant parameters

Best, mean and worst function values were selected and recorded separately for further screening and tuning.

3.10.3 Best Parameters Tuning

The initial range of [15; 21] and fitness limit of [55] were very significant at giving the best performance of the toolbox after the screening, therefore, these two parameters were run at varied crossover fractions of 0.2, 0.4, 0.6 and 0.8 in order to find the best function value at a particular crossover fraction.

3.11 Overall Methodological Flowchart

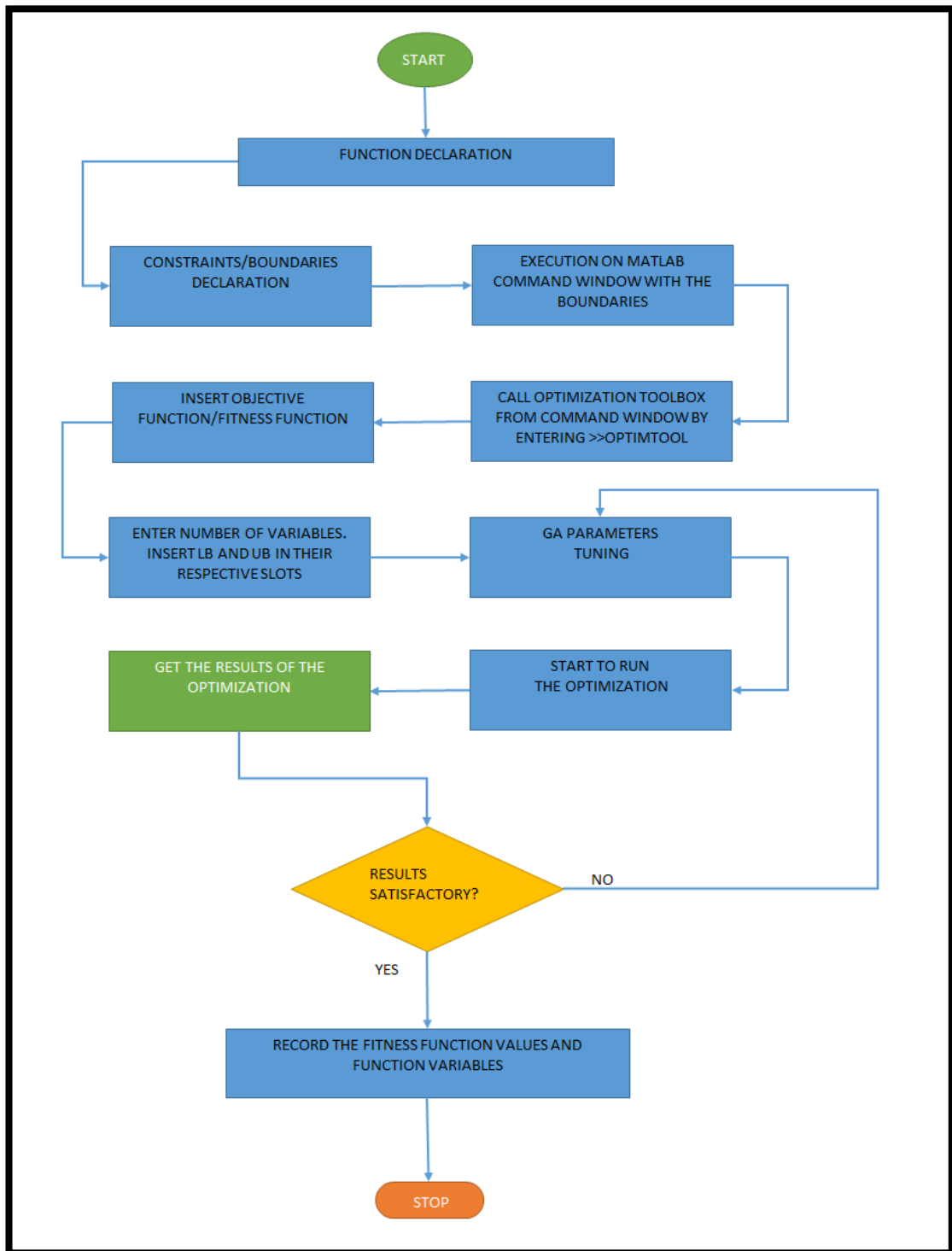


Figure 9 shows overall Methodology flowchart

CHAPTER 4: RESULTS AND DISCUSSION

4.0 Data collection

This section records all the results from various sets of parameter settings. As explained from the methodology part.

4.1 Genetic Algorithm Parameter Screening

This section gives the findings of the parameters screening which was done in the following approaches.

- i. Default initial range at default fitness limit

Table 3 shows the default parameters of the GA

Parameter	Variable value
Population type	Double
Population size	20
Creation function	constraint dependent
Initial population	Default []
Initial scores	Default []
Initial Range	[0; 1]
Fitness scaling	Rank
Selection function	Stochastic
Reproduction Option	Elite count with default value of 2 and default crossover fraction of 0.8
Mutation function	Adaptive feasible
Fitness limit	Default

The optimization was run using the above table of default parameters.

The results obtained are as below;

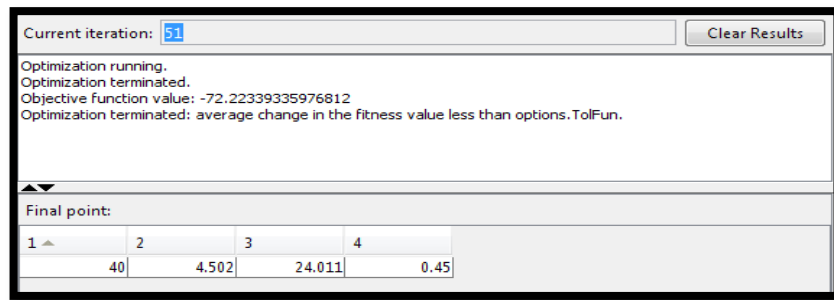


Figure 10 shows the optimization results display on the tool box

Function value = -72.22 g/l

Point values are $x_1 = 40$, $x_2 = 4.5$, $x_3 = 24$ and $x_4 = 0.45$

The negative function value of results obtained from the default parameters values of the optimization toolbox are not reasonable since it is a maximization problem and the results expected are strictly to be positive values. This shows that the toolbox doesn't perform well under only default settings for this particular problem.

ii. Specified initial ranges against default fitness limits

Secondly, the initial ranges were specified and there were positive values for the function value but experiences high deviation/variation each time the optimization was run. The results for this were not recorded as it was a randomly done.

Table 4 shows the parameters table with specified initial range and default fitness limit

Parameter	Variable value
Population type	Double
Population size	20
Creation function	constraint dependent
Initial population	Default []
Initial scores	Default []
Initial Range	[10; 20], [15; 20], [20; 30]...
Fitness scaling	Rank
Selection function	Stochastic
Reproduction Option	Elite count with default value of 2 and default crossover fraction of 0.8
Mutation function	Adaptive feasible
Fitness limit	Default

iii) Specified initial ranges against fitness limits

The initial ranges and fitness limits were specified. The initial ranges of [10; 30], [10; 20], and [15; 30], [15; 20] and [15; 21] were varied with three fitness limits of 50, 55 and 60 each and then run for four times. The results were recorded and tabulated in the appendix (i), (ii), (iii) and (iv) and appendix (v).

Table 5: Parameters table with specified initial ranges and fitness limits

Parameter	Variable value
Population type	Double
Population size	20
Creation function	constraint dependent
Initial population	Default []
Initial scores	Default []
Initial Range	[10; 30], [10; 20], [15; 30], [15;20] and [15;21]
Fitness scaling	Rank
Selection function	Stochastic
Reproduction Option	Elite count with default value of 2 and default crossover fraction of 0.8
Mutation function	Adaptive feasible
Fitness limit	[50], [55] and [60]

From section (iii) of the results, the initial ranges and fitness limits were specified as seen in table above. The other parameters were held constant. The results were recorded. Please refer to the appendices (i) to (v).

It can be observed that for initial range of [10; 30], the best fitness value is when the fitness limit is set at [55], the results for [10; 20] initial range are all negatives and therefore are far away from the optimal function value, meanwhile the best function value was obtained at initial range of [15; 30] and at [50] fitness limit. Initial ranges of [15; 20] gives negative function values at all three fitness limits. Initial range of [15; 21] give the best function value at fitness limit of [55].

The reason for significant variations in the results at different initial ranges is that, there are several local maxima found in the search space and therefore, if approximately where the solution to the problem lies is known, the initial range

should be specified so that it contains the guess for the solution. However, the genetic algorithm can find the solution even if it does not lie in the initial range, provided that the populations have enough diversity.

4.2 Best, mean and worst function values selection

After recording all the results obtained in the section (iii) above, the best mean and worst function value for different initial ranges at different fitness limits were selected and recorded in the following tables.

- a) Best function values for different initial ranges at different fitness limits

Table 6: The best function values for different initial ranges and fitness limits

Initial Range	Function Values (g/l)		
	Fitness limits		
	50	55	60
[10, 30]	24.04	27.07	18.95
[10;20]	-12.52	-33.23	-26.92
[15;30]	23.70	23.39	18.98
[15;20]	-23.70	-30.17	-33.23
[15;21]	49.39	50.2	50.09

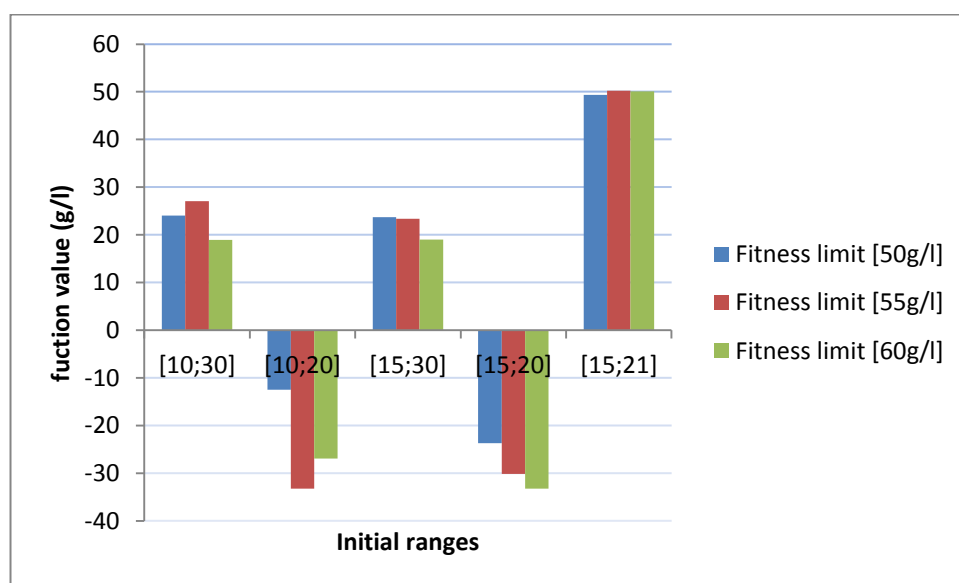


Figure 11: shows the graph of best function values for different initial ranges and fitness limit

b) Mean function values for different initial ranges and fitness limits

Table 7: Mean function value for different initial ranges and fitness limits

		Function Values (g/l)		
		Fitness limits		
Initial Range		50	55	60
[10;30]		18.475	23.4075	16.7525
[10;20]		-33.075	-39.7675	46.1325
[15;30]		19.3	19.9125	17.1975
[15;20]		-35.4575	-43.31	49.5475
[15;21]		48.4225	48.8575	47.9475

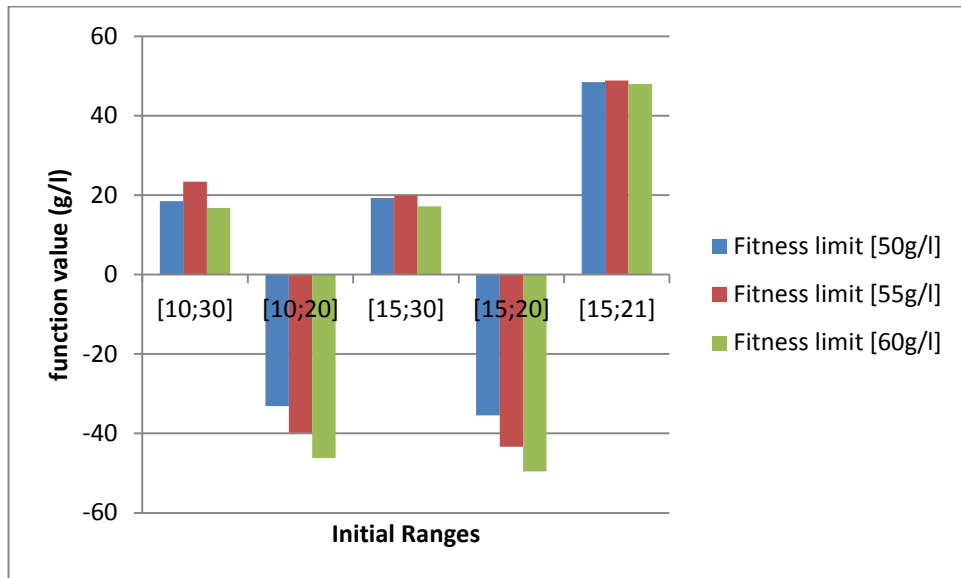


Figure 12 shows the graph of Mean function values against initial ranges

c) Worst function value for different initial ranges and fitness limits

Table 8: The worst function value for different initial ranges and fitness limits

Initial Range	Function Values (g/l)		
	Fitness limits		
	50	55	60
[10;30]	12.25	19.62	14.24
[10;20]	-43.42	-48.42	-57.99
[15;30]	16.12	15.81	15.81
[15;20]	-45.32	-63.75	-60.46
[15;21]	47.12	46.46	46.46

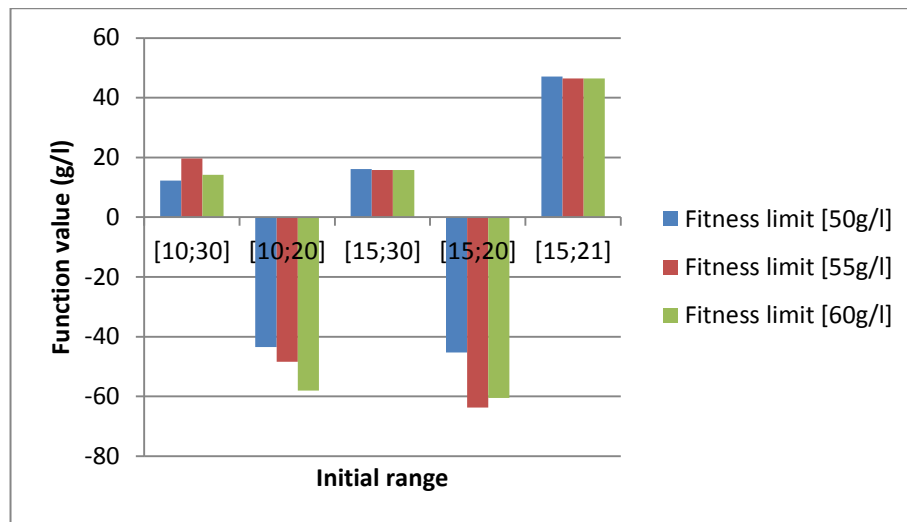


Figure 13 shows the graph of worst function values against fitness limits

Tables (6), (7) and (8) shows the best, mean and worst function values obtained when initial ranges and fitness limits are varied against each other. The results from the three indicate that, the initial ranges of [10; 20] and [15; 20] are the parameters that give the worst performance of the GA. Initial ranges of [10; 30], [15; 30] and [15; 21] proves very significant as far as performance of the GA is concern. However, the initial range of [15; 21] has the best function values as compared to the other two.

Similarly, Figures 11, 12 and 13 represent the performance of the GA under different initial ranges and fitness limits. The graphs show that initial ranges of [10; 20] and [15; 20] give negative function values which are not feasible since the main target is to maximize the function values. Initial ranges of [10; 30], [15; 30] and [15; 21] from

the graphs give apposite function values and therefore, they are considered significant for the performance of the GA.

It can be seen from these three initial ranges that, initial range of [15; 21] gives the best function values as compared to the other initial ranges of [10; 30] and [15; 30]. Looking at the best combination of parameters between initial ranges and fitness limits, the results shows that initial range of [15;21] and fitness limit of [55] gives the best function values as it can be seen from graphs 1, 2 and 3. The fitness limits are differentiated by three colours of blue, red and green. The blue colour represents fitness limit of [50], red and green colours represents the fitness limit of [55] and [60] respectively.

4.3 Best Parameters tuning

Initial range of [15;21] and fitness limit of [55] were identified to be the most significant values for optimal function value, therefore, these two parameters and their values were maintained constant meanwhile the crossover fractions were varied so as to tune for the best crossover fraction. The results were recorded in the below table (9) and their graph was plotted as shown in Figure (14). The table below displays the best parameters of the GA that led to the optimum function value.

Table 9: Table for the best parameters of the GA

Parameter	Variable value
Population type	Double
Population size	20
Creation function	constraint dependent
Initial population	Default []
Initial scores	Default []
Initial Range	[15;21]
Fitness scaling	Rank
Selection function	Stochastic
Reproduction Option	Elite count with default value of 2 and different crossover fractions
Mutation function	Adaptive feasible
Fitness limit	[55]

Table 10: Crossover fraction against best initial range and fitness limit against

Cross over fractions	Fitness values (g/l)						
	Runs				Worst	Mean	Best
	1	2	3	4			
0.2	46.46	46.03	48.59	47.1	46.03	47.05	48.59
0.4	46.46	48.92	48	48	46.46	47.85	48.92
0.6	48.59	48	46.46	46.46	46.46	47.38	48.59
0.8	48.59	46.46	46.46	48	46.46	47.38	48.59

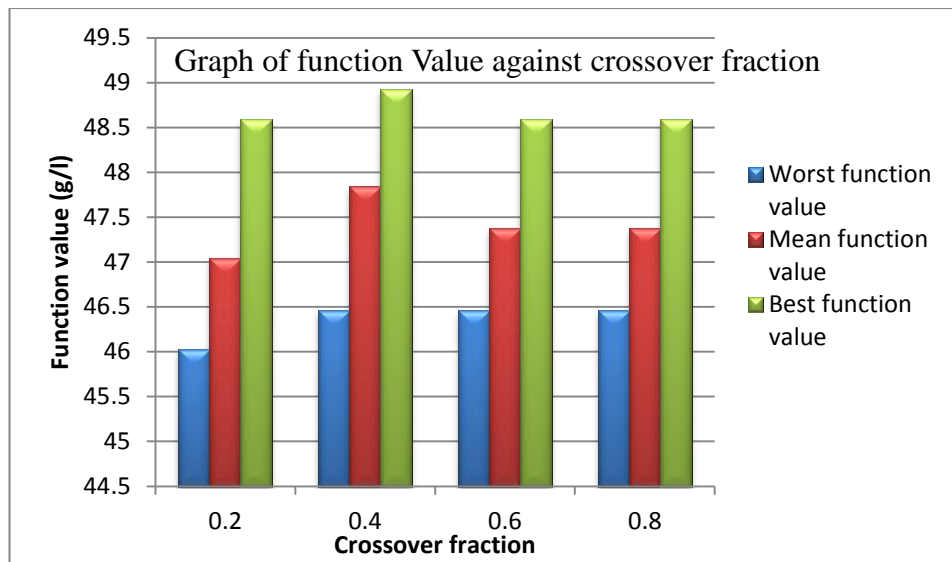


Figure 14 shows Function value against crossover fraction

Table (10) represent the performance of GA under the best parameter values of initial range [15; 21] and fitness limit [55] at different crossover fraction. This is done to investigate for the crossover fraction that gives the best function value. Figure (14) indicates that the crossover fraction of 0.4 gives the best function values at four different runs of the simulation. The worst function value at these particular conditions is 46.46g/l, mean function value is 47.85g/l and the best function value is 48.92g/l. This is the best performance of the GA so far as far as optimization of ethanol production is concern. Given that the worst concentration of ethanol obtained to be 48.92 g/l by GA is far higher than the empirical testing value obtained by Diptarka *et al.* (2013), there is enough evidence that this algorithm works very well for this particular problem.

4.4 Plots of results from the optimization toolbox

The toolbox plots the results based on the researcher's discretion. The plots are generated by the software after they have been selected on their checkboxes by the researcher. The plots selected are best fitness function, best individual and genealogy as shown in the results part.

The following plots show the best fitness value, best individual and genealogy plots in Figures (15), (16) and appendix (vi) respectively.

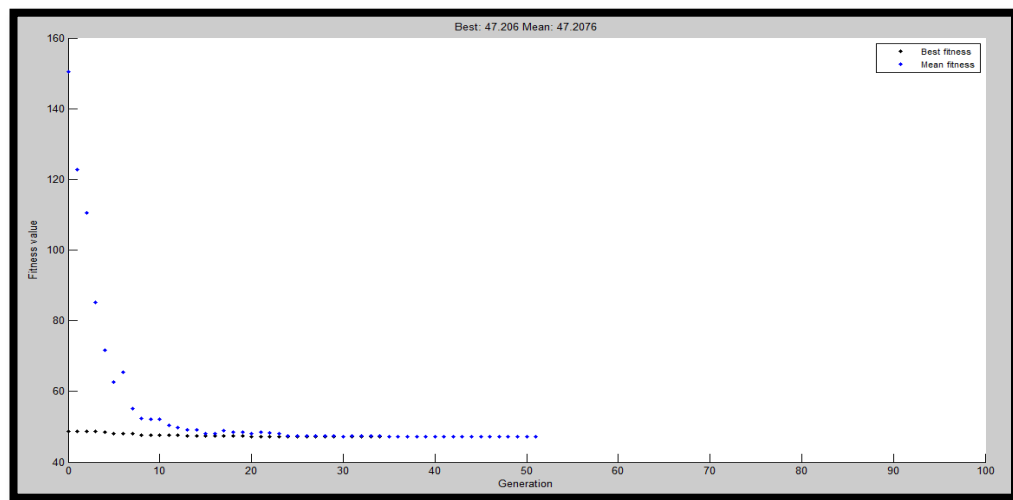


Figure 15 shows Best fitness function against generation

Figure 15 shows the best function value in each generation versus iteration number. The figure indicates that the best function value is got at about 47g/l after 50 generations.

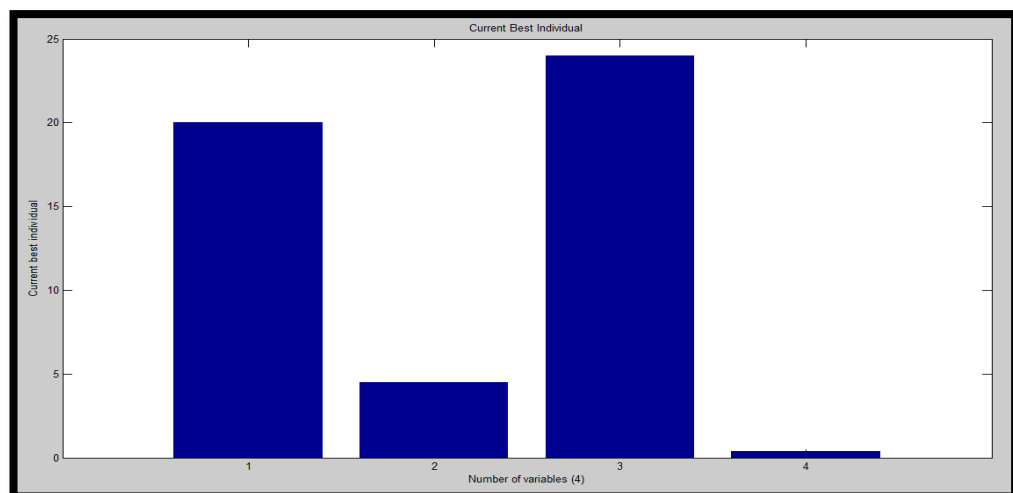


Figure 16: Best individual variable (1=substrate concentration, 2=pH, 3 = fermentation time and 4 = Na_2HPO_4)

Figure 16 shows the vector entries of the individual with the best fitness function value in each generation. The individuals in this case are the variables where variables 1, 2 and 3 and 4 represent substrate concentration, pH and fermentation time and Na_2HPO_4 respectively. It can be seen that among the four variables, fermentation time plays an important role towards maximization of ethanol production in contrast to Na_2HPO_4 which has the least contribution. Variable 1 which is the substrate concentration comes next to fermentation time as far as significance of each variable is concern followed by pH in the optimization of ethanol production.

The genealogy of individuals where lines from one generation to the next are color-coded such that the red lines indicate mutation children, blue lines indicate crossover children and black lines indicate elite individuals as shown in appendix (vi).

Genetic algorithm has got other several parameters which were also manipulated in order to find optimal solution for the problem. However, there was no significant effect on the function value as compared to initial range, fitness limit and crossover fraction. These parameters include Population size, initial scores and Initial population.

CHAPTER 5: CONCLUSIONS & RECOMMENDATIONS

The main objectives for this project were to identify important process variables that affect the production process of ethanol by fermentation of sugarcane bagasse, to find the interaction effects of process variables on the objective function value using genetic algorithm and to tune the genetic algorithm parameters that work best for ethanol production optimization and importantly to optimize ethanol production from sugarcane bagasse using genetic algorithm.

From the results of the project, it can be seen that the objectives have been achieved. The main significant variable for the optimization of ethanol production is found to be fermentation time followed by substrate concentration. pH and Na₂HPO₄ also important variables but they contribute significantly low to the function value. The optimal values of the variables are 20.14g/l for substrate concentration, 4.5 for pH and 24hours for fermentation time and 0.15 g/l for Na₂HPO₄. The best function value at these variable values is 50.2 g/l of ethanol.

The genetic algorithm parameters that give the best performance and stability of GA are initial range of [15; 21], fitness limit of [55] and crossover fraction of 0.4. These conditions give a mean function value of 47.85g/l, worst function value of 46.46g/l and the best function value of 48.92g/l. Therefore, the average function value of 47.85g/l is considered as outcome of this project and will be compared to the predicted value of Respond Surface methodology (RSM) model of 17.44g/l. This shows a significant robustness of GA over RSM with a difference of 30.41g/l which is a percentage increase of 174%.

There are other several evolutionary algorithms available which could work best on this given problem. Therefore another evolutionary computation technique known as particle swarm optimization (PSO) is recommended.

Particle swarm optimization (PSO) is a population- based search algorithm which is initialized with a population of random solutions known as particles. In PSO, each particle is associated with a velocity then particles fly through the search space with velocities according to their dynamic adjustment based of historical behaviours. The

particles will then continuously fly towards the better search area over the course of search process.

REFERENCES

- Adeyemo, A. M. (2011). Food processing optimization using evolutionary. *African Journal of Biotechnology Vol. 10(72), pp. 16120-16127, 16 November, 2011.*
- Ahmed, F. M., Rahman, S. R., and Gomes, D. J., 2012, "Saccharification of Sugarcane Bagasse by Enzymatic Treatment for bioethanol production," *Malaysian Journal of Microbiology*, Department of Microbiology, Jahangirnagar University, Savar, Dhaka-1342 and Bangladesh Department of Microbiology, University of Dhaka, Dhaka-1000
- Bai, Y. Y., Xiao, L. P., Zheng-Jun Shi, Z. J. and Sun, R. C., 2013, "Structural variation of bamboo lignin before and after ethanol Organosolv Pretreatment," *International Journal of Molecular Sciences*, Beijing Key Laboratory of Lignocellulosic Chemistry, Beijing Forestry University, Beijing 100083, China and State Key Laboratory of Pulp and Paper Engineering, South China University of Technology, Guangzhou 510640, China.
- Benvenga, M. A. C., Librantz, A. F. H and Araújo, S. A., 2012, "Use of the Genetic Algorithm in the Simulation and Optimization of the Hydrolysis Process of Cassava Starch to Obtain the Bioethanol," *International conference on industrial engineering and operations management*, Nove de Julho University, (UNINOVE), Av. Francisco Matarazzo, São Paulo Brazil.
- Chipperfield, A. J. and Fleming, P. J., 1995, "Colloquium on Applied Control Techniques Using MATLAB," Department of Automatic Control and Systems Engineering, University of Sheffield, England
- Dias, M.O.S., Adriano V. E., Silvia A. N., Filho, R.M., Carlos, E.V. and Maciel, M.R.W., 2009, "production of bioethanol and other bio-based materials from sugarcane bagasse: Integration to conventional bioethanol production process," *Chemical Engineering Research and Design*.
- Diptarka Dasgupta, S. K. (2013). Design and optimization of ethanol production. *SpringerOpen Journal*, 10.

- Enitan, M. A. and Adeyemo, J., 2011, "Food processing optimization using evolutionary algorithms," African Journal of Biotechnology. Department of Biotechnology and Food Technology, Durban University of Technology, Durban, South Africa and Department of Civil Engineering and Surveying, Durban University of Technology South Africa.
- Walia, N. K., Sekhon K. K., Chaudhary, D. P., Cameotra, SS, Srivastava, P., 2014, "Optimization of fermentation parameters for Bioconversion of Corn to Ethanol Using Response Surface Methodology," Petroleum & Environmental Biotechnology 5: 178. doi
- Wong, Y. C and Sanggari, V., April, 2014, "Bioethanol Production from Sugarcane Bagasse using Fermentation Process," Oriental journal of chemistry, Faculty of Agro Based Industry, University Malaysia Kelantan, Jeli Campus, Locked Bag 100, 17600 Jeli, Kelantan, Malaysia.
- Vargas, B. G. and Pereira, N., 2010, "Sugarcane bagasse as feedstock for second generation ethanol production," Part I: Diluted acid pre-treatment optimization," Electronic Journal of Biotechnology, Pontificia Universidad Católica de Valparaíso, Chile
- Verônica Ferreira. (2010). Simultaneous saccharification and fermentation process of different. *Electronic Journal of Biotechnology ISSN: 0717-3458*.

APPENDICES

Appendix (i): Initial range of [10;30] against three fitness limits

Variables values					Function Value	fitness limits
Runs	x1	x2	x3	x4	Ethanol(g/l)	(g/l)
1	29.58	4.56	24.03	0.16	14.28	
2	29.83	4.59	24.09	0.44	12.29	50
3	28.66	4.67	24.3	0.23	24.04	
4	28.54	4.66	24.14	0.17	23.29	
1	30	5.07	24.2	0.21	27.07	
2	29.1		24.26	0.17	19.64	55
3	28.64	4.62	24.77	0.41	25.4	
4	29.16	4.51	24.95	0.32	21.52	
1	29.85	4.5	24.78	0.27	16.29	
2	29.3	4.62	24.25	0.44	17.53	60
3	29.73	4.52	24.46	0.39	14.24	
4	28.84	4.51	24.4	0.3	18.95	

Appendix (ii): Initial range of [10;20]

Variables values					Function Value	fitness limits
Runs	x1	x2	x3	x4	Ethanol(g/l)	(g/l)
1	39.35	4.55	27	0.35	-33.15	
2	36.77	5.04	24.79	0.18	-12.52	50
3	39.84	4.9	24.82	0.19	-43.21	
4	39.84	5.07	24.52	0.39	-43.42	
1	39.26	4.61	25.06	0.29	-48.42	
2	39.96	4.61	26.33	0.4	-44.09	55
3	38.2	4.81	24.73	0.21	-33.23	
4	36.47	4.63	24.12	0.39	-33.33	
1	39.84	4.51	25.13	0.25	-55.62	
2	39.26	4.73	24.9	0.23	-44	60
3	37.3	4.56	24.64	0.27	-26.92	
4	39.3	4.56	24.54	0.44	-57.99	

Appendix (iii): Initial range of [15;30]

Runs	Variables values				Function Value	fitness limits
	x1	x2	x3	x4	Ethanol(g/l)	(g/l)
1	29.8	4.55	24.97	0.27	19.45	
2	28.85	4.5	24	0.23	16.12	50
3	28.88	4.76	24.22	0.44	17.93	
4	30.28	4.86	24.76	0.27	23.7	
1	29.3	4.53	24.3	0.41	15.81	
2	29.82	4.74	24.13	0.23	18.29	55
3	28.52	4.6	24.37	0.24	23.39	
4	28.56	4.61	24.24	0.27	22.16	
1	29.78	4.6	24.63	0.4	17.7	
2	29.83	4.59	24.34	0.23	15.81	60
3	30.14	4.61	24.88	0.2	18.98	
4	29.98	4.51	24.84	0.28	16.3	

Appendix (iv): Initial range of [15;20]

Runs	Variables values 2				Function Value	fitness limits
	x1	x2	x3	x4	Ethanol(g/l)	(g/l)
1	38.9	4.66	24.8	0.28	-45.32	
2	38.34	4.7	25.14	0.26	-35.35	50
3	39.79	4.75	26.39	0.44	-37.42	
4	34.73	4.51	24.04	0.4	-23.74	
1	39.55	4.79	26.53	0.31	-30.17	
2	39.55	4.51	24.26	0.37	-63.75	55
3	38.59	4.73	24.37	0.31	-44.25	
4	38.68	4.59	25.94	0.28	-35.07	
1	39.91	4.54	25.88	0.21	-47.12	
2	38.16	4.98	24.24	0.32	-33.23	60
3	39.76	4.6	24.2	0.24	-60.46	
4	39.64	4.53	24.89	0.36	-57.38	

Appendix (v): Initial range of [15;21]

Runs	Variables values				Function Value	fitness limits
	x1	x2	x3	x4	Ethanol(g/l)	(g/l)
1	20.89	4.5	24	0.36	47.12	
2	20.88	4.58	24.12	0.23	49.39	50
3	20	4.5	24	0.15	48.59	
4	20	4.5	24	0.15	48.59	
1	20	4.5	24	0.15	49.97	
2	20.14	4.5	24	0.15	50.2	55
3	20.7	4.5	24	0.15	48.8	
4	21	4.5	24	0.45	46.46	
1	20	4.5	24	0.38	50.09	
2	21	4.5	24	0.26	47.05	60
3	20.93	4.5	24	0.45	48.19	
4	21	4.5	24	0.15	46.46	

Appendix (vi): Genealogy plot

