AN EXPERIMENTAL STUDY OF NEIGHBOURHOOD BASED META-HEURISTIC ALGORITHMS FOR TEST CASE GENERATION SATISFYING THE MODIFIED CONDITION / DECISION COVERAGE CRITERION

ARIFUL HAQUE

MASTERS OF SCIENCE (SOFTWARE ENGINEERING)

UNIVERSITI MALAYSIA PAHANG
SUPERVISOR’S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Masters of Science (Software Engineering).

_______________________________
(Supervisor’s Signature)
Full Name : Prof. Dr. Kamal Zuhairi Bin Zamli
Position : Professor
Date : May 2018
STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_________________________________________________
(Student’s Signature)

Full Name : Ariful Haque
ID Number : MCS13002
Date : May 2018
AN EXPERIMENTAL STUDY OF NEIGHBOURHOOD BASED META-HEURISTIC ALGORITHMS FOR TEST CASE GENERATOR SATISFYING THE MODIFIED CONDITION / DECISION COVERAGE CRITERION

ARIFUL HAQUE

Thesis submitted in fulfillment of the requirements for the award of the degree of
Master of Science (Software Engineering)

Faculty of Computer Systems and Software Engineering
UNIVERSITI MALAYSIA PAHANG

May 2018
ACKNOWLEDGEMENTS

First of all, I would like to show my sincere gratitude and thank to my supervisor Prof. Dr. Kamal Z. Zamli. His endless enthusiasm, constantly advice, support and patient have encouraged me in completing this research work of master degree.

I would also like to thank my wife Salma Sultana Tajia to keep inspiring and encouraging me to continue the research and complete the thesis. Thanks to my parents who not only supported me, also funded my study in initial level. Without this support, it would not be easy for me to start the Masters course.

Next, my appreciations are expressed towards the administrative staffs in the Faculty of Computer Systems and Software Engineering (FSKKP), Institute of Postgraduate Studies (IPS) and International Office (IO) for their passionate to handle all forms of official tasks during my master study. The appreciations also go to all members of Bangladeshi community, local and foreign friends in UMP.

Finally, this research work of Master study is funded and supported by UMP Postgraduate Research Grant Scheme (PRGS) - “Development of AI Based Test Data Generation Strategy Satisfying MC/DC Criterion”, Graduate Research Scheme (GRS) - “Development of AI Based Test Data Generation Strategy Satisfying MC/DC Criterion” from IPS, UMP. I would to thank UMP for funding the research and helping me to concentrate on the study.
TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS  ii

ABSTRAK  iii

ABSTRACT  iv

TABLE OF CONTENT  v

LIST OF TABLES  ix

LIST OF FIGURES  x

LIST OF ABBREVIATIONS  xii

CHAPTER 1 INTRODUCTION  1

1.1 Background and Motivation  1

   1.1.1 Importance of Software Testing  2

1.2 Problem Statement  4

   1.2.1 Problem with Existing Works and Motivation  5

1.3 Research Aim and Objective  6

1.4 Scope of the Research  6

1.5 Organization of the Thesis  7

CHAPTER 2 LITERATURE REVIEW  8

2.1 Introduction  8

2.2 MC/DC Criterion  8

   2.2.1 Definitions  8

   2.2.2 Structural Coverage and MC/DC  9
2.2.3 MC/DC Test Cases 12

2.3 Overview of Software Testing 15
   2.3.1 Black Box Testing 16
   2.3.2 White Box Testing 16

2.4 Search Based Software Engineering 17

2.5 Search Based Software Testing 18
   2.5.1 Solution 21
   2.5.2 Fitness Function 22

2.6 Structural Testing 23

2.7 Use of Metaheuristics Algorithms to Automate Test Case Generation 25
   2.7.1 Metaheuristics Algorithms 25
   2.7.2 Evolutionary Testing 27
   2.7.3 Local Search Techniques 29

2.8 Masking Problem and Modified Condition / Decision Coverage (MC/DC) Criterion 42

2.9 Summary 50

CHAPTER 3 METHODOLOGY 52

3.1 Introduction 52
   3.1.1 Justification of Neighbourhood-based metaheuristics algorithm 52

3.2 Algorithm Implementation Design 52
   3.2.1 Control Flow Graph 53
   3.2.2 Decision coverage and MC/DC coverage 55

3.3 MC/DC Pair Generation using Neighbourhood-based Algorithm 56
   3.3.1 Stopping Criteria 57
   3.3.2 Hill Climbing 57
3.3.3 Great Deluge Algorithm 58
3.3.4 Simulated Annealing 60
3.3.5 Late Acceptance Hill Climbing 61
3.4 Algorithm Calibration 64
   3.4.1 Comparative Studies 64
3.5 Subject Expressions 65
3.6 Approach Steps 65
3.7 Summary 66

CHAPTER 4 RESULTS AND DISCUSSION 68
4.1 Introduction 68
4.2 Result Presentation 68
4.3 Discussion 69
   4.3.1 Group A: Expressions with 3 conditions 70
   4.3.2 Group B: Expressions with 4 conditions 74
   4.3.3 Group C: Expressions with 8 conditions 78
4.4 Observation 83
   4.4.1 Presence of No Free Lunch Theorem: 83
   4.4.2 Exploration and Exploitation 84
   4.4.3 Easy to use 84
4.5 Summary 84

CHAPTER 5 CONCLUSION 86
5.1 Introduction 86
5.2 Concluding Remarks 86
5.3 Contributions 87
LIST OF TABLES

Table 2.1  Truth table of decision at line 21 of the Calculate method  15
Table 2.2  A Domestic Case Studies Probability to Overcome Software Failures  47
Table 3.1  Algorithm Configuration Values  64
Table 3.2  List of Boolean expressions for experiment  65
Table 4.1  Result of minimum, maximum number of test cases generated  69
Table 4.2  Result of minimum, maximum and average run time from experiments  69
LIST OF FIGURES

Figure 2.1  Representations of Elementary Logical Gates 13
Figure 2.2  Calculate Method 14
Figure 2.3  Block diagram of the genetic algorithm 29
Figure 2.5  Local Search Schema 31
Figure 2.6  Local optima problem in local search 32
Figure 2.7  Simulated Annealing Algorithm. Initial Temperature to cooling down process. 36
Figure 2.8  Schema for Simulated Annealing 37
Figure 2.9  Schema for Great Deluge Algorithm 39
Figure 2.10  Schema for the LAHC algorithm 42
Figure 2.11  Simple function with AND condition 43
Figure 2.12  Simple function with OR condition and control flow graph 44
Figure 2.13  Equivalent If statement for AND operator of Figure 2.11 45
Figure 2.14  Equivalent If statement for OR operator of Figure 2.12 46
Figure 3.1  Calculation function 54
Figure 3.2  CFG of calculation method 55
Figure 3.3  Pseudo Code of Hill Climbing Algorithm 58
Figure 3.4  Great Deluge Algorithm Pseudo Code 60
Figure 3.5  Pseudo code of Simulated Annealing Algorithm 61
Figure 3.6  Late Acceptance Hill Climbing Pseudo Code 63
Figure 3.7  Approach Steps in the Automation of Testing 66
Figure 4.1  Number of test cases generated for the Expression 1 (Group A) 70
Figure 4.2  Number of test cases generated for the Expression 2 (Group A) 71
Figure 4.3  Number of test cases generated for the Expression 3, Group A. 71
Figure 4.4  Time Performance of different algorithms for Expression 1 (Group A) 72
Figure 4.5  Time Performance of different algorithms for Expression 2 (Group A) 73
Figure 4.6  Time Performance different algorithms for Expression 3 (Group A) 73
Figure 4.7  Number of test cases generated for the Expression 4 (Group B) 75
Figure 4.8  Number of test cases generated for the Expression 5 (Group B) 75
Figure 4.9  Number of test cases generated for the Expression 6 (Group B) 76
Figure 4.10  Result of Execution time of Expression 4 (Group B) 77
<table>
<thead>
<tr>
<th>Figure 4.11</th>
<th>Result of execution time for Expression 5 (Group B)</th>
<th>77</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4.12</td>
<td>Time Performance of Algorithms for Expression 6 in Group B</td>
<td>78</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Number of test cases generated for the expression 7 in Group C.</td>
<td>79</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Number of test cases generated for the expression 8 in Group C.</td>
<td>80</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Number of test cases generated for the expression 9 in Group C.</td>
<td>80</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Time Performance of Algorithms for the expression 7 in Group C</td>
<td>82</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Time Performance of Algorithms for the expression 7 in Group C</td>
<td>82</td>
</tr>
<tr>
<td>Figure 4.18</td>
<td>Time Performance of Algorithms for the expression 9 (Group C)</td>
<td>83</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ACO</td>
<td>Ant Colony Optimization</td>
<td></td>
</tr>
<tr>
<td>AMP</td>
<td>Adaptive Memory Programming</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>Condition Coverage</td>
<td></td>
</tr>
<tr>
<td>CFG</td>
<td>Control Flow Graph</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Decision Coverage</td>
<td></td>
</tr>
<tr>
<td>ET</td>
<td>Evolutionary Testing</td>
<td></td>
</tr>
<tr>
<td>FAA</td>
<td>Federal Aviation Administration</td>
<td></td>
</tr>
<tr>
<td>GDA</td>
<td>Great Deluge Algorithm</td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical user Interface</td>
<td></td>
</tr>
<tr>
<td>HC</td>
<td>Hill Climbing</td>
<td></td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
<td></td>
</tr>
<tr>
<td>LAHC</td>
<td>Late Acceptance Hill Climbing</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>Linear Programming</td>
<td></td>
</tr>
<tr>
<td>MC/DC</td>
<td>Modified Condition Decision Coverage</td>
<td></td>
</tr>
<tr>
<td>MCC</td>
<td>Multiple Condition Coverage</td>
<td></td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
<td></td>
</tr>
</tbody>
</table>