

BRANCH COVERAGE TEST CASE GENERATION USING GENETIC ALGORITHM
AND HARMONY SEARCH

HOSEIN ABEDINPOURSHOTORBAN

A dissertation submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

JANUARY 2015

I declare that this dissertation entitled "*Branch Coverage Test Case Generation using Genetic Algorithm and Harmony Search*" is the result of my own research except as cited in the references. The dissertation has not been accepted for any degree and is not concurrently submitted in candidature for any other degree.

Signature :

Name : HOSEIN ABEDINPOURSHOTORBAN

Date : JANUARY 26, 2015

I strongly dedicated this dissertation to my beloved parents for their supports,
encouragement and love.

ACKNOWLEDGEMENT

I would like to thank my supervisor **Assoc. Prof. Dr. Dayang Norhayati Abang Jawawi**, who has been an invaluable friend and mentor. Her gift for conceptualization, her enduring encouragement, and her practical advice have been an inestimable source of support for me during this research.

Besides, my endless gratitude goes to my parents for their strong moral and emotional support. I would also like to thank my dear friend Nur Fatimah As'Sahra who shared this journey with me, and always supported me.

ABSTRACT

Due to the vital role of software in the modern world, there is a great demand for reliability, and it can be achieved through the process of testing. White-Box testing is one of the testing methods that aims to uncover errors of coding by investigating the internal structure of the software. Moreover, generation of test cases for White-Box testing of software can be done manually or automatically. However, due to possible mistakes and expenses of manual test case generation, trend is toward making this activity automatic. So far, proposed techniques for automatic test case generation are mostly based on Genetic Algorithm (GA). However, existing GA techniques are quite slow and unable to achieve full coverage when it comes to test case generation for complex software with a wide range of inputs. Thus, in this research an improved fitness function is proposed based on Control Dependence Graph (CDG) and branch distance that can improve the speed and coverage of test cases generation by the means of evolutionary algorithms like GA. Also, a GA-based branch coverage test case generation technique is proposed in this research that takes advantage of our proposed fitness function, and comparison results based on two benchmark case studies show that our proposed technique outperforms the original CDG technique in speed and coverage of test case generation. In addition, we evaluated our proposed fitness function with harmony search algorithm (HS), which is a more recent optimization algorithm compared to GA, and find out that HS outperforms GA in speed of test case generation for branch coverage of software code.

ABSTRAK

Oleh kerana perisian memainkan peranan yang penting dalam dunia moden, terdapat permintaan yang besar terhadap kemampuannya, dan ia boleh dicapai melalui proses ujian. Ujian Kotak Putih merupakan salah satu kaedah ujian yang bertujuan untuk mendedahkan kesilapan pengekodan dengan menyiasat struktur dalaman perisian. Selain itu, generasi kes-kes ujian untuk ujian Kotak Putih terhadap perisian boleh dilakukan secara manual ataupun secara automatik. Namun, disebabkan oleh kesilapan dan penggunaan ujian manual generasi kes, haluan ke arah menjadikan aktiviti ini automatik. Setakat ini, teknik yang dicadangkan untuk kes ujian generasi automatik kebanyakannya berdasarkan Algoritma Genetik (GA). Namun, teknik GA sedia ada agak perlahan dan tidak dapat mencapai liputan sepenuhnya apabila ia digunakan untuk menguji generasi kes untuk perisian kompleks dengan pelbagai input. Oleh itu, dalam kajian ini fungsi kecergasan yang lebih baik adalah dicadangkan berdasarkan Kawalan Kebergantungan Graf (CDG) dan jarak cawangan yang boleh meningkatkan kelajuan dan liputan generasi kes-kes ujian dengan cara-cara evolusi algoritma seperti GA. Tambahan lagi, satu cabang ujian liputan teknik penjanaan kes berdasarkan GA dicadangkan dalam kajian ini yang mengambil kesempatan daripada fungsi kecergasan yang kami dicadangkan, dan keputusan perbandingan berdasarkan dua kajian kes penanda aras menunjukkan bahawa teknik yang kami cadangkan melebihi prestasi teknik CDG asal dalam kelajuan dan liputan ujian generasi kes. Selain itu, kami menilai fungsi kecergasan kami dengan algoritma pencarian harmoni (HS), yang merupakan algoritma pengoptimuman yang lebih terkini berbanding GA, dan kami mengetahui bahawa HS melebihi prestasi GA dalam kelajuan ujian generasi kes untuk liputan cawangan kod perisian.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xiv
1	INTRODUCTION	
1.1	Overview	1
1.2	Problem Background	4
1.3	Problem Statement	7
1.4	Research Aim	8
1.5	Research Objectives	8
1.6	Scope of the Study	8
1.7	Significance of Study	9
1.8	Dissertation Organization	9
2	LITERATURE REVIEW	
2.1	Introduction	11
2.2	Software Testing Definition	11
2.3	Overview of Software Testing	13

2.4	Basic Terminology for Software Testing	15
2.5	Undecidability of Software Testing	15
2.6	Software Testing Levels	16
2.7	Types of Testing	19
2.7.1	Static Techniques	19
2.7.2	Dynamics Techniques	19
2.8	White-Box Testing of Software Unit	20
2.9	Control-Flow Graph (CFG)	21
2.9.1	Overview of Code Coverage	22
2.9.2	Code Coverage Criteria (Metrics)	23
2.10	Relations between Software Testing Definitions	24
2.11	Test Case Generation	25
2.11.1	Random Test Data Generation	26
2.11.2	GA-Based Test Case Generation	27
2.12	Harmony Search Algorithm	29
2.13	Related Works	31
2.14	Tracey et al (1998) Technique for Branch Distance Calculation	34
2.15	Pargas et al (1999) GA Technique for Software Testing	35
2.16	Criteria for Measuring Speed of GA-Based Test Case Generation Technique	37
2.17	Characteristic of Effective Fitness Function	37
2.18	Summary	38
3	RESEARCH METHODOLOGY	
3.1	Introduction	39
3.2	Research Process Flowchart	39
3.3	Research Process	41
3.3.1	Phase One	42
3.3.2	Phase Two	42
3.3.3	Phase Three	43
3.3.4	Phase Fours	44
3.4	Benchmark Programs	45
3.5	Methodology Framework	46

3.6 Summary	47
4 PROPOSED TECHNIQUE FOR TEST CASE GENERATION	
4.1 Introduction	48
4.2 Coding and CFG of Benchmark Programs	49
4.3 Pargas Technique(Original)	51
4.3.1 Control Dependence Graph	51
4.3.2 Example of Pargas Technique Fitness Evaluation and Shortcoming	53
4.4 The Proposed Technique	54
4.4.1 Enhanced Fitness Function	54
4.4.1.1 Example of Fitness Evaluation by Enhanced Fitness Function	58
4.4.2 Proposed Algorithm	58
4.4.2.1 Part 1: Initialization	60
4.4.2.2 Part 2: Test Case Generation	60
4.4.2.3 Part 3: Example of Test Case Generation by Proposed Algorithm	60
4.5 Characteristics of Proposed Technique	62
5 COMPARISON OF PROPOSED TECHNIQUE AND ORIGINAL TECHNIQUE	
5.1 Introduction	63
5.2 Experimental Setup	63
5.3 Comparison of Proposed Technique and Pargas et al (1999) Technique	64
5.3.1 Summary of Comparison between Proposed Technique and Pargas et al (1999) Technique	69
5.4 Evaluation of Proposed Fitness Function	70
5.4.1 Comparison Summary of GA and HS for Branch Coverage Test Case Generation	72
5.5 Summary	73

6	CONCLUSION AND FUTURE WORK	
6.1	Introduction	74
6.2	Summary	74
6.3	Research Contribution	75
6.4	Future Work	76
	REFERENCES	77-81