

UNIVERSITY TEKNOLOGI MARA

**ANT COLONY ALGORITHM FOR
TEXT CLASSIFICATION IN
MULTICORE-MULTITHREAD
ENVIRONMENT**

AHMAD NAZMI BIN FADZAL

Thesis submitted in fulfilment
of the requirements for the degree of
Master of Science

Faculty of Computer and Mathematical Sciences

January 2017

ABSTRACT

In the age of wide digital usage, text classification is one of the significant prominent attribute required in order to automatically arrange emails, articles, and other textual data in an organization. Unclassified data can lead to slower data retrieval thus a reliable method is required to effectively retrieve data efficiently and in systematic manner. Ant Colony Optimization (ACO) is a bio-inspired technique that was introduced to solve Non-Polynomial hard problem of high text data dimension that is similar to Traveling Salesman Problem (TSP) using probabilistic way. Pheromone concept is the main criterion that distinguish ACO to other algorithms. Based on the concept, pheromone saturation is used to combine stackable solution pattern that is discovered while straying to different term node to build a path. ACO classification accuracy is compared to Genetic Algorithm classifier which also a wrapper method. On integration of the technique, ACO is proposed to work in a multicore-multithread environment to gain additional execution time advantage. In multicore-multithread environment, the adjustment aims to make artificial ants communicate across the physical core of processor. As a trade to the investment for more computing power, the execution time reduction is expected to show an improvement without compromising the original classification accuracy. The unthreaded and multicore-multithreaded version of ACO was experimented and compared in term of accuracy and execution time. It was found that the result show a positive improvement.

ACKNOWLEDGEMENT

Firstly, I wish to thank God for giving me the opportunity to embark on my master programme and for completing this long and challenging journey successfully. My gratitude and thanks go to my supervisor Prof Madya Dr. Mazidah Puteh, with co-supervisor, Assoc. Prof. Dr. Adnan Ahmad and Dr. Norlela Samsudin. Thank you for the support, patience and ideas in assisting me with this project.

Finally, this thesis is dedicated to the loving memory of my very dear mother and late father for the vision and determination to educate me. This piece of victory is dedicated to both of you. Alhamdulillah.

TABLE OF CONTENTS

	Page
CONFIRMATION BY PANEL OF EXAMINERS	ii
AUTHOR'S DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER ONE: INTRODUCTION	1
1.1 Background of Study	2
1.2 Problem Statement	3
1.3 Objectives	6
1.4 Scope	6
1.5 Research Methodology	7
1.6 Significance	8
1.7 Summary and Contribution	9
1.8 Term Definition	9
1.9 Thesis Organization	11
CHAPTER TWO: LITERATURE REVIEW	12
2.1 Introduction	12
2.2 Text Classification	12
2.3 Genetic Algorithm	14
2.4 Support Vector Machine	20
2.5 Ant Colony Algorithm	24
2.5.1 Pheromone Reading	29
2.5.2 Evaporation	30
2.5.3 Stagnation	31

2.5.4 Factors in Choosing ACO over GA and other Techniques	32
2.6 Multicore-Multithread Environment	33
2.6.1 Threading Over Parallelism for Multicore-Multithread Representation	35
2.6.2 Staying Idle	36
2.6.3 Code Development of Multitasking	38
2.6.4 Multithreading in Software Level	39
2.6.5 Multiplexing Time Division	40
2.6.6 Schedule of Multiplexing	41
2.6.7 Multithreading in Hardware Level	43
2.6.8 Advantages and Disadvantages Multicore Processor	44
2.6.9 Backup Slower Threads	46
2.7 Summary	49
CHAPTER THREE: RESEARCH METHODOLOGY	50
3.1 Introduction	50
3.2 Phase 1: Knowledge Acquisition	51
3.3 Phase 2: Text Data Collection and Identification	53
3.3.1 Data Pre-processing	54
3.3.2 GA Experiment Setting	58
3.3.3 Development of ACO	60
3.3.3.1 Simulation of Artificial Ant Visiting Term Nodes	63
3.3.3.2 Stopping Criterion	65
3.3.4 ACO Text Classification	66
3.3.4.1 Skip Common Term Calculation with Same Frequency	66
3.3.4.2 Obtaining Classification Rule Every Cycle	68
3.3.4.3 Choosing Next State	69
3.3.4.4 Pheromone Usage	72
3.3.5 Set Up Multicore-Multithread Environment	73
3.3.6 Multicore-Multithread ACO Model	75
3.3.6.1 Overcome Unstable Execution Speed	78
3.3.6.2 Signalling and Stopping Multiple Threads	79
3.4 Phase 3: Evaluation on Multicore-Multithread ACO Model	82
3.5 Summary	84