

**ANT COLONY OPTIMIZATION IN DYNAMIC
ENVIRONMENTS**

CHEN FEI HUANG



**THISIS SUBMITTED IN FULFILLMENT FOR
THE DEGREE OF MASTER OF SCIENCE IN
SOFTWARE ENGINEERING**

**SCHOOL OF ENGINEERING AND
INFORMATION TECHNOLOGY
UNIVERSITI MALAYSIA SABAH
2010**

UNIVERSITI MALAYSIA SABAH

BORANG PENGESAHAN TESIS

JUDUL : _____

IJAZAH : _____

SAYA : _____ SESI PENGAJIAN : _____

(HURUF BESAR)

Mengaku membenarkan tesis *(LPSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Universiti Malaysia Sabah dengan syarat-syarat kegunaan seperti berikut:-

1. Tesis adalah hak milik Universiti Malaysia Sabah.
2. Perpustakaan Universiti Malaysia Sabah dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (/)

SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di AKTA RAHSIA RASMI 1972)

TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PUSTAKAWAN)

Alamat Tetap: _____

(NAMA PENYELIA)

TARIKH: _____

TARIKH: _____

Catatan:

*Potong yang tidak berkenaan.

*Jika tesis ini SULIT dan TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT dan TERHAD.

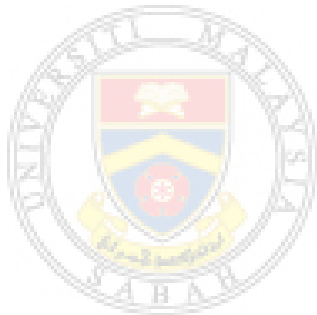
*Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana Secara Penyelidikan atau disertai bagi pengajian secara kerja kursus dan Laporan Projek Sarjana Muda (LPSM).

DECLARATION

I hereby declare that the material in this thesis is my own except for quotations, excerpts, equations, summaries and references, which have been duly acknowledged. Any contribution made to research by others, with whom I have worked at UMS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

04 AUGUST 2010

CHEN FEI HUANG
PS05-008-026



UMS
UNIVERSITI MALAYSIA SABAH

CERTIFICATION

NAME : CHEN FEI HUANG
MATRIC NUMBER : PS05-008-026
TITLE : ANT COLONY OPTIMIZATION IN DYNAMIC ENVIRONMENTS
DEGREE : MASTER OF SCIENCE (SOFTWARE ENGINEERING)
VIVA DATE : 4 JUNE 2010

DECLARED BY

1. SUPERVISOR



Dr. Jason Teo Tze Wi



ACKNOWLEDGEMENTS

I would like to thank all the people who had helped me along the way in completing this thesis. First of all, I would like to take the opportunity to thank Universiti Malaysia Sabah especially to the Dean of the School of Engineering and Information Technology, Associate Professor Dr. Rosalam Sabartly. I would like to express my sincere gratitude to my supervisor, Dr. Jason Teo for his precious guidance, advice and assistance during my research. I would also like to thank the academics and staffs of the School of Engineering and Information Technology that provided a lot of guidance and help.

My acknowledgement also goes to the PGD Scholarship provided by YTL, with further support from Centre for Postgraduate Studies, Universiti Malaysia Sabah.

I would also like to express my gratitude to my family and friends for their devotion, support and patience.

Last but not least, it would not be possible to list down everyone that has helped and lend their assistance in completing this work. I am deeply honored and your dedication is truly remembered.

Chen Fei Huang
4 August 2010



UMMS
UNIVERSITI MALAYSIA SABAH

ABSTRACT

ANT COLONY OPTIMIZATION IN DYNAMIC ENVIRONMENTS

Optimization is the process of finding the maximum or minimum of some objective function. During recent years, a new direction of research has emerged in optimization, with the research focus moving from the conventional static optimization problems to the dynamic optimization problems. Dynamic optimization problems differ from the static ones where the optimal solution is dynamic and changes over time. A similar trend can be seen in the application of Ant Colony Optimization (ACO). ACO is an optimization technique inspired by the ants' foraging behavior which optimizes their routes taken to food sources. This thesis is an investigation into the application of ACO for solving dynamic optimization problems. The first objective of this study is to identify which ant algorithm performs the best under a dynamic environment. In order to achieve this objective, six ant algorithms namely Ant System (AS), Ant Colony System (ACS), Best-Worst Ant System (BWAS), Elitist Ant System (EAS), Max-Min Ant System (MMAS) and Rank-Based Ant System (RBAS) were implemented to solve a dynamic optimization problem in the form of the dynamic Traveling Salesman Problem (TSP). Three different sizes of the dynamic TSP test sets were used: eil51 (small), lin318 (medium) and d1291 (large). Apart from the size of the optimization problem, how the swapping interval affects the dynamic optimization by the ant algorithms is also investigated. Swapping of cities in the dynamic TSP was done in the early, middle and late stages of the optimization process. A series of 30 test runs were conducted on each dynamic TSP instance and also for each swap condition. The second objective of the research is to investigate the suitability of applying local search algorithms to the best performing ant algorithm from the first objective. For this purpose, three local search algorithms namely 2-opt, 2.5-opt and 3-opt were chosen to be coupled with the ant algorithm in order to solve the dynamic TSP. The last objective of this thesis is to optimize the parameter settings of the best performing ant algorithm with local search. From the experimental analysis, it was found that ACS works best for solving the dynamic TSP compared to the other five ant algorithms. When coupled with a local search technique, a significant improvement can be seen for ACS using the 3-opt local search algorithm. Lastly, it was also found that different optimal parameter settings were required for solving the different sizes of the dynamic TSP problems.

ABSTRAK

Pengoptimuman merupakan satu proses pencarian maksimum atau minimum fungsi berobjektif. Sejak kebelakangan ini, suatu halatuju penyelidikan yang baru telah timbul dalam bidang pengoptimuman, di mana fokus penyelidikan telah berubah daripada masalah pengoptimuman statik kepada masalah pengoptimuman dinamik. Perbezaan di antara masalah dinamik dengan masalah statik adalah masalah dinamik mempunyai penyelesaian optimum yang dinamik dan berubah mengikut masa. Corak yang sama dapat dilihat dalam pengaplikasian "**Ant Colony Optimization**" (ACO). ACO adalah teknik pengoptimuman yang diilhamkan oleh sifat pencarian makanan semut yang mengoptimumkan penghalaan jalan-jalan mereka ke punca makanan. Tesis ini menerangkan satu penyelidikan yang dijalankan untuk mengenalpasti algoritma semut yang mempunyai pencapaian terbaik dalam keadaan dinamik. Objektif pertama penyelidikan ini ialah mengenal pasti algoritma semut yang berpencapaian terbaik di dalam keadaan dinamik. Untuk mencapai objektif ini, enam algoritma semut, iaitu "**Ant System**" (AS), "**Ant Colony System**" (ACS), "**Best-Worst Ant System**" (BWAS), "**Elitist Ant System**" (EAS), "**Max-Min Ant System**" (MMAS) dan "**Rank-Based Ant System**" (RBAS) telah dibangunkan untuk menyelesaikan masalah dinamik yang berbentuk masalah dinamik "**Traveling Salesman**" (TSP). Tiga set ujian TSP yang berlainan saiz telah digunakan: eil51 (kecil), lin318 (sederhana) dan d1291 (luas). Selain daripada saiz masalah yang berlainan, kesan masa penukaran bandar-bandar pada masa-masa berlainan ke atas pengoptimuman dinamik oleh algoritma semut juga dikaji. Penukaran bandar-bandar di dalam TSP dinamik telah dilakukan pada tahap awal, pertengahan dan akhir semasa proses pengoptimuman. Satu siri ujian yang merangkumi 30 pelaksanaan bagi setiap masalah TSP dinamik dan juga untuk setiap masa penukaran telah dijalankan. Objektif kedua penyelidikan ialah menyiasat kesesuaian penggunaan algoritma pencarian tempatan apabila diaplikasikan kepada algoritma terbaik hasil dapatan daripada objektif pertama untuk penyelesaian masalah dinamik. Untuk mencapai tujuan ini, tiga algoritma pencarian tempatan iaitu 2-opt, 2.5-opt dan 3-opt telah dipilih untuk diaplikasikan bersama algoritma semut bagi menyelesaikan TSP dinamik. Untuk objektif terakhir, penentuan parameter terbaik bagi algoritma semut bersama dengan algoritma pencarian tempatan dilaksanakan. Analisa hasil kajian menunjukkan bahawa ACS mempunyai prestasi terbaik berbanding lima algoritma yang lain dalam keadaan dinamik. Peningkatan prestasi yang ketara dapat dilihat bagi algoritma semut apabila algoritma pencarian tempatan 3-opt diaplikasikan bersama ACS. Hasil ujian daripada objektif terakhir menunjukkan penentuan parameter terbaik yang berlainan diperlukan untuk menyelesaikan dinamik TSP yang berlainan saiz.

TABLE OF CONTENTS

	Page
FRONT PAGE	i
DECLARATION	ii
CERTIFICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
LIST OF SYMBOLS	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Overview	1
1.2 Ant Colony Optimization	1
1.3 Dynamic Environment	3
1.3.1 Traveling Salesman Problem and Dynamic Traveling Salesman Problem	3
1.4 Motivation Of Research	4
1.5 Objectives Of Research	5
1.6 Scope And Limitations Of The Research	6
1.7 Expected Outcomes Of The Research	6
1.8 Structure Of Thesis	6
1.9 Conclusion	7
CHAPTER 2: LITERATURE REVIEW	9
2.1 Overview	9
2.2 Swarm Intelligence	9
2.3 Ant Colony	9
2.3.1 The artificial ants compared to real ants	10
2.4 Ant Colony Optimization Metaheuristic	11
2.4.1 The basic framework of the ACO algorithm	12
2.5 Ant Colony Optimization Algorithm	15
2.5.1 Ant System (AS)	15
2.5.2 Ant Colony System (ACS)	16
2.5.3 Elitist Ant System (EAS)	18
2.5.4 Max-Min Ant System (MMAS)	18
2.5.5 Rank-based Ant System (RBAS)	19
2.5.6 Best-Worst Ant System (BWAS)	20
2.6 ACO Application To Solve Optimization Problems	21

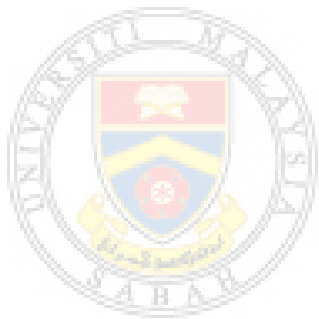
2.7	ACO Application In Dynamic Optimization Problems	22
2.8	Local Search Techniques	24
2.8.1	2-opt	24
2.8.2	2.5-opt	26
2.8.3	3-opt	29
2.9	Critical Summary Of Literature Review	31
2.10	Conclusion	32
CHAPTER 3: METHODOLOGY		33
3.1	Overview	33
3.2	Hardware And Software Specifications	33
3.3	Test Bed	34
3.4	Pheromone Equalization Strategy	35
3.5	Local Search Techniques	36
3.6	Experimental Setup	38
3.7	Parameter Tuning Setup	40
3.8	Performance Metrics	40
3.9	Conclusion	41
CHAPTER 4: ANT COLONY OPTIMIZATION FOR DYNAMIC TSP		42
4.1	Overview	42
4.2	Pheromone Equalization Strategy	42
4.3	Experimental Setup	43
4.4	Results And Discussion	44
4.5	Comparison Of ACS And EAS	49
4.6	Experimental Findings	53
4.7	Conclusion	54
CHAPTER 5: LOCAL SEARCH TECHNIQUES IN ANT COLONY SYSTEM FOR DYNAMIC TSP		55
5.1	Overview	55
5.2	Experimental Setup	55
5.3	Results And Discussion	56
5.4	Conclusion	62
CHAPTER 6: OPTIMIZING THE PARAMETER SETTINGS OF ANT COLONY SYSTEM FOR DYNAMIC TSP		64
6.1	Overview	64
6.2	Experimental Objectives	64
6.3	Parameter Tuning	65
6.4	Results And Discussion	67
6.4.1	Number of ants	67
6.4.2	β value	71
6.4.3	ρ value	79
6.5	ACS With Optimal Parameter Setting Vs ACS With Previous Parameter Setting	85
6.6	Conclusion	89
CHAPTER 7: CONCLUSION		90
7.1	Overview	90
7.2	Summary Of Results	90
7.3	Research Contributions	92
7.4	Future Works	92
7.5	Conclusion	93

REFERENCES

94

APPENDIXES

97



UMS
UNIVERSITI MALAYSIA SABAH

LIST OF FIGURES

		Page
Figure 1.1	The process how ants find the shorter path.	2
Figure 2.1	The ACO metaheuristic in pseudo-code.	13
Figure 2.2	An example of 2-opt move on the original tour.	25
Figure 2.3	The 2-opt algorithm.	26
Figure 2.4	An example of 2.5-opt move on the original tour.	27
Figure 2.5	The 2.5-opt algorithm.	28
Figure 2.6	Examples of 3-opt move on the original tour.	29
Figure 2.7	The 3-opt algorithm.	30
Figure 3.1	A 2-opt move on the original tour.	36
Figure 3.2	A 2.5-opt move on the original tour.	37
Figure 3.3	A 3-opt move on the original tour.	38
Figure 4.1	Average optimal value for 30 test runs on eil51 TSP.	45
Figure 4.2	Average optimal value for 30 test runs on lin318 TSP.	46
Figure 4.3	Average optimal value for 30 test runs on d1291 TSP.	48
Figure 4.4	Average optimal value comparison for ACS and EAS on eil51.	50
Figure 4.5	Average optimal value comparison for ACS and EAS on lin318.	52
Figure 4.6	Average optimal value comparison for ACS and EAS on d1291.	53
Figure 5.1	Average optimal value on experiment conducted on eil51 TSP.	57
Figure 5.2	Average optimal value on experiment conducted on lin318 TSP.	59
Figure 5.3	Average optimal value on experiment conducted on d1291 TSP.	61
Figure 5.4	ACS-3opt running on d1291 TSP.	62
Figure 6.1	Average optimal value on number of ants tuning on eil51 TSP.	68
Figure 6.2	Average optimal value on number of ants tuning on lin318 TSP.	69
Figure 6.3	Average optimal value on number of ants tuning on d1291 TSP.	70
Figure 6.4	Relationship between the optimal number of ants and TSP size.	71
Figure 6.5	Average optimal value on β tuning on eil51 TSP (20 ants setting).	73
Figure 6.6	Average optimal value on β tuning on eil51 TSP (30 ants setting).	73
Figure 6.7	Average optimal value on β tuning on lin318 TSP.	75
Figure 6.8	Average optimal value on β tuning on d1291 TSP.	77
Figure 6.9	Relationship between the best Beta setting and TSP size.	79
Figure 6.10	Average optimal value on ρ value tuning on eil51 TSP.	80
Figure 6.11	Average optimal value on ρ value tuning on lin318 TSP.	81
Figure 6.12	Average optimal value on ρ value tuning on d1291 TSP.	83
Figure 6.13	Relationship between the optimal ρ value and TSP size.	85
Figure 6.14	Average optimal value comparison for eil51 TSP.	86
Figure 6.15	Average optimal value comparison for lin318 TSP.	87
Figure 6.16	Average optimal value comparison for d1291 TSP.	88

LIST OF TABLES

		Page
Table 2.1	Applications of ACO.	21
Table 2.2	Recent applications of ACO on dynamic optimization problems.	24
Table 3.1	The number of cities on TSP instance and spare cities pool.	39
Table 3.2	Total of iteration run for each TSP instance for each test run.	39
Table 3.3	Parameter settings for each TSP instance.	40
Table 4.1	Parameter setting for the six ant algorithms.	43
Table 4.2	Results of 30 test runs on eil51 TSP.	46
Table 4.3	Results of 30 test runs on lin318 TSP.	47
Table 4.4	Results of 30 test runs on d1291 TSP.	48
Table 4.5	Experimental results for eil51 TSP.	50
Table 4.6	Experimental results for lin318 TSP.	51
Table 4.7	Experimental results for d2191 TSP.	52
Table 5.1	Parameter settings for ACS.	56
Table 5.2	Result of 30 runs for eil51 TSP.	56
Table 5.3	Result of 30 runs for lin318 TSP.	58
Table 5.4	Result of 30 runs for d1291 TSP.	60
Table 6.1	Parameter tuning for each TSP instance.	66
Table 6.2	Parameters setting of ACS on previous experiment.	67
Table 6.3	Best optimal value on number of ants tuning on eil51 TSP.	68
Table 6.4	Best optimal value on number of ants tuning on lin318 TSP.	69
Table 6.5	Best optimal value on number of ants tuning on d1291 TSP.	70
Table 6.6	Best optimal value on number of ants tuning on eil51 TSP.	72
Table 6.7	Best optimal value on β tuning on lin318 TSP.	74
Table 6.8	Best optimal value on β tuning on d1291 TSP.	76
Table 6.9	Best optimal value on ρ tuning on d1291 TSP.	82
Table 6.10	ACS parameter settings comparison.	88

LIST OF ABBREVIATIONS

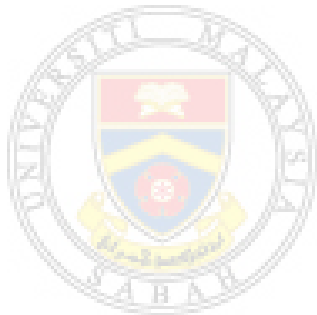
ACO	Ant Colony Optimization
ACS	Ant Colony System
AS	Ant System
BWAS	Best-Worst Ant System
EAS	Elitist Ant System
MMAS	Max-Min Ant System
PSO	Particle Swarm Optimization
QAP	Quadratic Assignment Problem
RBAS	Rank-based Ant System
TSP	Traveling Salesman Problem



UMS
UNIVERSITI MALAYSIA SABAH

LIST OF SYMBOLS

α	Parameter that controls the pheromone information
β	Parameter that controls the heuristic information
d_{ij}	The distance between city i and city j
ρ	Parameter that control the evaporation rate
τ_{ij}	The pheromone information between city i and city j
η_{ij}	The heuristic information between city i and city j
$S_{global-best}$	The global-best ant



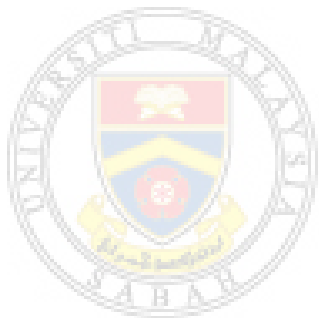
UMS
UNIVERSITI MALAYSIA SABAH

DECLARATION

I hereby declare that the material in this thesis is my own except for quotations, excerpts, equations, summaries and references, which have been duly acknowledged. Any contribution made to research by others, with whom I have worked at UMS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

04 AUGUST 2010

CHEN FEI HUANG
PS05-008-026



UMS
UNIVERSITI MALAYSIA SABAH

CERTIFICATION

NAME : CHEN FEI HUANG
MATRIC NUMBER : PS05-008-026
TITLE : ANT COLONY OPTIMIZATION IN DYNAMIC ENVIRONMENTS
DEGREE : MASTER OF SCIENCE (SOFTWARE ENGINEERING)
VIVA DATE : 4 JUNE 2010

DECLARED BY

1.

SUPERVISOR

Dr. Jason Teo Tze Wi



UMMS
UNIVERSITI MALAYSIA SABAH

ACKNOWLEDGEMENTS

I would like to thank all the people who had helped me along the way in completing this thesis. First of all, I would like to take the opportunity to thank Universiti Malaysia Sabah especially to the Dean of the School of Engineering and Information Technology, Associate Professor Dr. Rosalam Sabartly. I would like to express my sincere gratitude to my supervisor, Dr. Jason Teo for his precious guidance, advice and assistance during my research. I would also like to thank the academics and staffs of the School of Engineering and Information Technology that provided a lot of guidance and help.

My acknowledgement also goes to the PGD Scholarship provided by YTL, with further support from Centre for Postgraduate Studies, Universiti Malaysia Sabah.

I would also like to express my gratitude to my family and friends for their devotion, support and patience.

Last but not least, it would not be possible to list down everyone that has helped and lend their assistance in completing this work. I am deeply honored and your dedication is truly remembered.

Chen Fei Huang
4 June 2010



UMMS
UNIVERSITI MALAYSIA SABAH

ABSTRACT

Optimization is the process of finding the maximum or minimum of some objective function. During recent years, a new direction of research has emerged in optimization, with the research focus moving from the conventional static optimization problems to the dynamic optimization problems. Dynamic optimization problems differ from the static ones where the optimal solution is dynamic and changes over time. A similar trend can be seen in the application of Ant Colony Optimization (ACO). ACO is an optimization technique inspired by the ants' foraging behavior which optimizes their routes taken to food sources. This thesis is an investigation into the application of ACO for solving dynamic optimization problems. The first objective of this study is to identify which ant algorithm performs the best under a dynamic environment. In order to achieve this objective, six ant algorithms namely Ant System (AS), Ant Colony System (ACS), Best-Worst Ant System (BWAS), Elitist Ant System (EAS), Max-Min Ant System (MMAS) and Rank-Based Ant System (RBAS) were implemented to solve a dynamic optimization problem in the form of the dynamic Traveling Salesman Problem (TSP). Three different sizes of the dynamic TSP test sets were used: eil51 (small), lin318 (medium) and d1291 (large). Apart from the size of the optimization problem, how the swapping interval affects the dynamic optimization by the ant algorithms is also investigated. Swapping of cities in the dynamic TSP was done in the early, middle and late stages of the optimization process. A series of 30 test runs were conducted on each dynamic TSP instance and also for each swap condition. The second objective of the research is to investigate the suitability of applying local search algorithms to the best performing ant algorithm from the first objective. For this purpose, three local search algorithms namely 2-opt, 2.5-opt and 3-opt were chosen to be coupled with the ant algorithm in order to solve the dynamic TSP. The last objective of this thesis is to optimize the parameter settings of the best performing ant algorithm with local search. From the experimental analysis, it was found that ACS works best for solving the dynamic TSP compared to the other five ant algorithms. When coupled with a local search technique, a significant improvement can be seen for ACS using the 3-opt local search algorithm. Lastly, it was also found that different optimal parameter settings were required for solving the different sizes of the dynamic TSP problems.

ABSTRAK

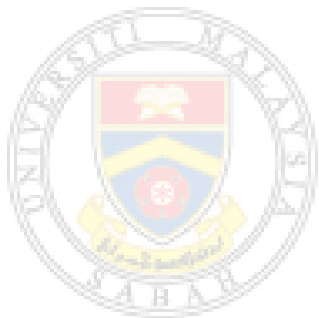
Optimisasi merupakan satu proses pencarian maksimum atau minimum fungsi berobjektif. Sejak kebelakangan ini, suatu halatuju penyelidikan yang baru telah bertimbul dalam bidang optimisasi, di mana focus penyelidikan telah berubah daripada masalah optimisasi 6riter kepada masalah optimisasi dinamik. Perbezaan di antara masalah dinamik dengan masalah 6riter adalah masalah dinamik mempunyai penyelesaian optima yang dinamik dan berubah mengikut masa. Corak yang sama dapat dilihat dalam pengaplikasian "**Ant Colony Optimization**" (ACO). ACO adalah teknik optimisasi yang diilhamkan oleh sifat pencarian makanan semut yang mengoptimasikan penghalaan jalan-jalan mereka ke punca makanan. Tesis ini menerangkan satu penyelidikan yang dijalankan untuk mengenalpasti algoritma semut yang mempunyai pencapaian terbaik dalam keadaan dinamik. Objektif pertama penyelidikan ini ialah mengenal pasti algoritma semut yang berpencapaian terbaik di dalam keadaan dinamik. Untuk mencapai objektif ini, enam algoritma semut, iaitu "**Ant System**" (AS), "**Ant Colony System**" (ACS), "**Best-Worst Ant System**" (BWAS), "**Elitist Ant System**" (EAS), "**Max-Min Ant System**" (MMAS) dan "**Rank-Based Ant System**" (RBAS) telah dibangunkan untuk menyelesaikan masalah dinamik yang berbentuk masalah dinamik "**Traveling Salesman**" (TSP). Tiga set ujian TSP yang berlainan saiz telah digunakan: eil51 (kecil), lin318 (sederhana) dan d1291 (luas). Selain daripada saiz masalah yang berlainan, kesan masa penukaran bandar-bandar pada masa-masa berlainan ke atas optimisasi dinamik oleh algoritma semut juga dikaji. Penukaran bandar-bandar di dalam TSP dinamik telah dilakukan pada tahap awal, pertengahan dan akhir semasa proses optimisasi. Satu siri ujian yang merangkumi 30 perlaksanaan bagi setiap masalah TSP dinamik dan juga untuk setiap 6riteria masa penukaran telah dijalankan. Objektif kedua penyelidikan ialah menyiasat kesesuaian penggunaan algoritma pencarian tempatan apabila diaplikasikan kepada algoritma terbaik daripada objektif pertama untuk penyelesaian masalah dinamik. Untuk mencapai tujuan ini, tiga algoritma pencarian tempatan iaitu 2-opt, 2.5-opt dan 3-opt telah dipilih untuk diaplikasikan bersama algoritma semut bagi menyelesaikan TSP dinamik. Untuk objektif terakhir, penentuan parameter terbaik bagi algoritma semut bersama dengan algoritma pencarian tempatan dilaksanakan. Analisa hasil kajian menunjukkan bahawa ACS mempunyai prestasi terbaik berbanding lima algoritma yang lain dalam keadaan dinamik. Peningkatan prestasi yang ketara dapat dilihat bagi algoritma semut apabila algoritma pencarian tempatan 3-opt diaplikasikan bersama ACS. Hasil ujian daripada objektif terakhir menunjukkan penentuan parameter terbaik yang berlainan diperlukan untuk menyelesaikan dinamik TSP yang berlainan saiz.

TABLE OF CONTENTS

	Page
FRONT PAGE	i
DECLARATION	ii
CERTIFICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
LIST OF SYMBOLS	xiii
CHAPTER 1: INTRODUCTION	1
1.1 OVERVIEW	1
1.2 ANT COLONY OPTIMIZATION	1
1.3 DYNAMIC ENVIRONMENT	3
1.3.1 Traveling Salesman Problem and Dynamic Traveling Salesman Problem	3
1.4 MOTIVATION OF RESEARCH	4
1.5 OBJECTIVES OF RESEARCH	5
1.6 SCOPE AND LIMITATIONS OF THE RESEARCH	6
1.7 EXPECTED OUTCOMES OF THE RESEARCH	6
1.8 STRUCTURE OF THESIS	6
1.9 CONCLUSION	8
CHAPTER 2: LITERATURE REVIEW	9
2.1 OVERVIEW	9
2.2 SWARM INTELLIGENCE	9
2.3 ANT COLONY	9
2.3.1 The artificial ants compared to real ants	10
2.4 ANT COLONY OPTIMIZATION METAHEURISTIC	11
2.4.1 The basic framework of the ACO algorithm	12
2.5 ANT COLONY OPTIMIZATION ALGORITHM	15
2.5.1 Ant System (AS)	15
2.5.2 Ant Colony System (ACS)	16
2.5.3 Elitist Ant System (EAS)	18
2.5.4 Max-Min Ant System (MMAS)	18
2.5.5 Rank-based Ant System (RBAS)	19
2.5.6 Best-Worst Ant System (BWAS)	20
2.6 ACO APPLICATION TO SOLVE OPTIMIZATION PROBLEMS	21

2.7	ACO APPLICATION IN DYNAMIC OPTIMIZATION PROBLEMS	22
2.8	Local Search Techniques	24
	2.8.1 2-opt	24
	2.8.2 2.5-opt	26
	2.8.3 3-opt	29
2.9	CRITICAL SUMMARY OF LITERATURE REVIEW	31
2.10	CONCLUSION	32
CHAPTER 3: METHODOLOGY		33
3.1	OVERVIEW	33
3.2	HARDWARE AND SOFTWARE SPECIFICATIONS	33
3.3	TEST BED	34
3.4	PHEROMONE EQUALIZATION STRATEGY	35
3.5	LOCAL SEARCH TECHNIQUES	36
3.6	EXPERIMENTAL SETUP	38
3.7	PARAMETER TUNING SETUP	40
3.8	PERFORMANCE METRICS	40
3.9	CONCLUSION	41
CHAPTER 4: ANT COLONY OPTIMIZATION FOR DYNAMIC TSP		42
4.1	OVERVIEW	42
4.2	PHEROMONE EQUALIZATION STRATEGY	42
4.3	EXPERIMENTAL SETUP	43
4.4	RESULTS AND DISCUSSION	44
4.5	COMPARISON OF ACS AND EAS	49
4.6	EXPERIMENTAL FINDINGS	53
4.7	CONCLUSION	54
CHAPTER 5: LOCAL SEARCH TECHNIQUES IN ANT COLONY SYSTEM FOR DYNAMIC TSP		55
5.1	OVERVIEW	55
5.2	EXPERIMENTAL SETUP	55
5.3	RESULTS AND DISCUSSION	56
5.4	CONCLUSION	62
CHAPTER 6: OPTIMIZING THE PARAMETER SETTINGS OF ANT COLONY SYSTEM FOR DYNAMIC TSP		64
6.1	OVERVIEW	64
6.2	EXPERIMENTAL OBJECTIVES	64
6.3	PARAMETER TUNING	65
6.4	RESULTS AND DISCUSSION	67
	6.4.1 Number of ants	67
	6.4.2 β value	71
	6.4.3 ρ value	79
6.5	ACS WITH OPTIMAL PARAMETER SETTING VS ACS WITH PREVIOUS PARAMETER SETTING	85
6.6	CONCLUSION	89
CHAPTER 7: CONCLUSION		90
7.1	OVERVIEW	90
7.2	SUMMARY OF RESULTS	90
7.3	RESEARCH CONTRIBUTIONS	92

7.4	FUTURE WORKS	92
7.5	CONCLUSION	93
REFERENCES		94
APPENDIXES		99



UMS
UNIVERSITI MALAYSIA SABAH

LIST OF FIGURES

		Page
Figure 1.1	The process how ants find the shorter path.	2
Figure 2.1	The ACO metaheuristic in pseudo-code.	13
Figure 2.2	An example of 2-opt move on the original tour.	25
Figure 2.3	The 2-opt algorithm.	26
Figure 2.4	An example of 2.5-opt move on the original tour.	27
Figure 2.5	The 2.5-opt algorithm.	28
Figure 2.6	Examples of 3-opt move on the original tour.	29
Figure 2.7	The 3-opt algorithm.	30
Figure 3.1	A 2-opt move on the original tour.	36
Figure 3.2	A 2.5-opt move on the original tour.	37
Figure 3.3	A 3-opt move on the original tour.	38
Figure 4.1	Average optimal value for 30 test runs on eil51 TSP.	45
Figure 4.2	Average optimal value for 30 test runs on lin318 TSP.	46
Figure 4.3	Average optimal value for 30 test runs on d1291 TSP.	48
Figure 4.4	Average optimal value comparison for ACS and EAS on eil51.	50
Figure 4.5	Average optimal value comparison for ACS and EAS on lin318.	52
Figure 4.6	Average optimal value comparison for ACS and EAS on d1291.	53
Figure 5.1	Average optimal value on experiment conducted on eil51 TSP.	57
Figure 5.2	Average optimal value on experiment conducted on lin318 TSP.	59
Figure 5.3	Average optimal value on experiment conducted on d1291 TSP.	61
Figure 5.4	ACS-3opt running on d1291 TSP.	62
Figure 6.1	Average optimal value on number of ants tuning on eil51 TSP.	68
Figure 6.2	Average optimal value on number of ants tuning on lin318 TSP.	69
Figure 6.3	Average optimal value on number of ants tuning on d1291 TSP.	70
Figure 6.4	Relationship between the optimal number of ants and TSP size.	71
Figure 6.5	Average optimal value on β tuning on eil51 TSP (20 ants setting).	73
Figure 6.6	Average optimal value on β tuning on eil51 TSP (30 ants setting).	73
Figure 6.7	Average optimal value on β tuning on lin318 TSP.	75
Figure 6.8	Average optimal value on β tuning on d1291 TSP.	77
Figure 6.9	Relationship between the best Beta setting and TSP size.	79
Figure 6.10	Average optimal value on ρ value tuning on eil51 TSP.	80
Figure 6.11	Average optimal value on ρ value tuning on lin318 TSP.	81
Figure 6.12	Average optimal value on ρ value tuning on d1291 TSP.	83
Figure 6.13	Relationship between the optimal ρ value and TSP size.	85
Figure 6.14	Average optimal value comparison for eil51 TSP.	86
Figure 6.15	Average optimal value comparison for lin318 TSP.	87
Figure 6.16	Average optimal value comparison for d1291 TSP.	88

LIST OF TABLES

		Page
Table 2.1	Applications of ACO.	21
Table 2.2	Recent applications of ACO on dynamic optimization problems.	24
Table 3.1	The number of cities on TSP instance and spare cities pool.	39
Table 3.2	Total of iteration run for each TSP instance for each test run.	39
Table 3.3	Parameter settings for each TSP instance.	40
Table 4.1	Parameter setting for the six ant algorithms.	43
Table 4.2	Results of 30 test runs on eil51 TSP.	46
Table 4.3	Results of 30 test runs on lin318 TSP.	47
Table 4.4	Results of 30 test runs on d1291 TSP.	48
Table 4.5	Experimental results for eil51 TSP.	50
Table 4.6	Experimental results for lin318 TSP.	51
Table 4.7	Experimental results for d1291 TSP.	52
Table 5.1	Parameter settings for ACS.	56
Table 5.2	Result of 30 runs for eil51 TSP.	56
Table 5.3	Result of 30 runs for lin318 TSP.	58
Table 5.4	Result of 30 runs for d1291 TSP.	60
Table 6.1	Parameter tuning for each TSP instance.	66
Table 6.2	Parameters setting of ACS on previous experiment.	67
Table 6.3	Best optimal value on number of ants tuning on eil51 TSP.	68
Table 6.4	Best optimal value on number of ants tuning on lin318 TSP.	69
Table 6.5	Best optimal value on number of ants tuning on d1291 TSP.	70
Table 6.6	Best optimal value on number of ants tuning on eil51 TSP.	72
Table 6.7	Best optimal value on β tuning on lin318 TSP.	74
Table 6.8	Best optimal value on β tuning on d1291 TSP.	76
Table 6.9	Best optimal value on ρ tuning on d1291 TSP.	82
Table 6.10	ACS parameter settings comparison.	88