

**IMMUNITY ANT COLONY SYSTEM (IACS) FOR SOLVING TRAVELING
SALESMAN PROBLEM**

LEE SUNG YAO

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS*

JUDUL: IMMUNITY ANT COLONY SYSTEM (IACS) FOR SOLVING TRAVELING SALESMAN PROBLEM

SESI PENGAJIAN: 2010 / 2011

Saya LEE SUNG YAO
(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi 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 dalam AKTA RAHSIA RASMI 1972)

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

 / TIDAK TERHAD


(TANDATANGAN PENULIS)

Alamat tetap: A-2516, Lrg Sri Kuantan 18,
Jln Beserah, 25300 Kuantan,
Pahang, Malaysia.

Tarikh: 13 JULY 2011


(TANDATANGAN PENYELIA)

DR. CHOO YUN HUOY
Nama Penyelia

Tarikh: 13/7/2011

CATATAN: *Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
**Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

**IMMUNITY ANT COLONY SYSTEM (IACS) FOR SOLVING TRAVELING
SALESMAN PROBLEM**

LEE SUNG YAO

This report is submitted in partial fulfillment of the requirements for the Bachelor of
Computer Science (Artificial Intelligence)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2011

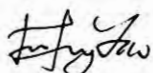
DECLARATION

I hereby declare that this project report entitled

IMMUNITY ANT COLONY SYSTEM FOR SOLVING TRAVELING SALESMAN PROBLEM

is written by me and is my own effort and that no part has been plagiarized without citations.


STUDENT

: 
_____ (Lee Sung Yao)

Date

: 13/7/2011

SUPERVISOR

: 
_____ (Dr. Choo Yun Huoy)

Date

: 13/7/2011

DEDICATION

This is dedicated to my beloved parents, Mr. Lee Faw Yong and Mrs. Leong Ai Mei,
for their expression of love and full support,

and to my supervisor, Dr. Choo Yun Huoy, for making it all worthwhile...

ACKNOWLEDGEMENTS

I would like to extend my gratitude to all those who have contributed directly and indirectly in completing this project. Firstly, I would like to give a special thanks to my Project Supervisor, Dr. Choo Yun Huoy for giving me assistance, guidance and encouragement to complete this project successfully. I also desire to thank Dr. Azah Kamilah, who has given me comments and advices to make the project more perfect.

Next, I wish to thank my beloved father, Lee Faw Yong, who has inspired me for getting the project title and motivated me to complete the project. Then, I would like to thank my university, Faculty of Information and Communication Technology lecturers who have taught me. With all the knowledge they have taught me, I manage to complete the project in time.

I would also like to thank my classmates for making this study a wonderful experience. Besides that, I would like to thank my friends who have accessed to my system and given me valuable and sincere comments. Finally, I want to express my sincere thanks to my parents who have given me full support with the encouragement, inspiration and patience which they provided at every step during this course of studies.

ABSTRACT

Ant Colony System (ACS) is a well-known method for solving Traveling Salesman Problem (TSP), which is the problem faced by a salesman who wants to find a shortest possible trip through a given set of cities, starting from his home town and visiting each city once before finally returning home. However, one underlying weakness of ACS is it has no protection against the generation of bad solution. Hence, a new method called Immunity Ant Colony System (IACS) was proposed to overcome this problem. The new method has an “immunity operator” which acts just like the immune system in human body. It can check for bad components in the solutions and then repair them. An application was developed to ease the implementation and testing of the new algorithms, by using Microsoft Visual Studio 2010 as the developing tool and Visual C# as the programming language. Datasets used for testing the algorithm were obtained from TSPLIB website. Three datasets were chosen for the experiment, and they are of different sizes. One of the parameters, the heuristic coefficient, was set at different values to test the effect on result. The new algorithm was then compared to ACS, and the result shows that IACS outperforms ACS in all the tested datasets. That is because every long link in the solution is “repaired” before they are brought to the next iteration, and thus producing higher quality of solution with less computation time.

TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	TABLE OF CONTENT	vi
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
CHAPTER I	INTRODUCTION	1
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Project Objective	2
	1.4 Project Scope	3
	1.5 Project Significant	3
	1.6 Expected Output	4
	1.7 Conclusion	4
CHAPTER II	LITERATURE REVIEW	5
	2.1 Introduction	5
	2.2 Traveling Salesman Problem (TSP)	5
	2.2.1 Types of Traveling Salesman Problem	6
	2.3 Techniques for solving TSP	10

2.3.1	Ant Colony System	10
2.3.2	Ant System	10
2.3.3	Elitist Ant System	11
2.3.4	Rank-Based Ant System	12
2.3.5	MAX-MIN Ant System	12
2.3.6	Genetic Algorithm (GA)	13
2.3.7	Evolutionary Programming	13
2.3.8	Simulated Annealing (SA)	14
CHAPTER III	METHODOLOGY	16
3.1	Project Methodology	16
3.2	Review	17
3.3	Propose	17
3.4	Design and Implementation of Algorithm	18
3.5	Design of Application (Simple Simulator)	19
3.6	Dataset	20
3.7	Comparison	21
3.8	Conclusion	22
CHAPTER IV	IMMUNITY ANT COLONY SYSTEM	23
4.1	Ant Colony System (ACS)	23
4.2.1	Representing Ants	25
4.2.2	Distances between Cities	26
4.2.3	Pheromone Trails	27
4.2.4	Solution Construction	27
4.2.5	ACS Algorithm	28
4.2	Immunity Operator	30
4.3	Immunity Ant Colony System	32
4.4	Conclusion	34
CHAPTER V	EXPERIMENTAL RESULT AND ANALYSIS	36
5.1	Result and Analysis for Dataset eil76	36
5.2	Result and Analysis for Dataset pr152	39
5.3	Result and Analysis for Dataset kroA200	42

	5.4 Conclusion	45
CHAPTER VI	CONCLUSION	46
	6.1 Weaknesses and Strengths	46
	6.2 Propositions for Improvement	47
	6.3 Contribution	48
	6.4 Conclusion	48
	REFERENCES	50
	APPENDICES	54

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Chronological development of ACO algorithm	11
5.1	Optimum distances for eil76	39
5.2	Optimum distances for pr152	42
5.3	Optimum distances for kroA200	45

LIST OF FIGURES

DIAGRAM	TITLE	PAGE
2.1	Triangle inequality	7
2.2	Manhattan distance and Euclidean distance	8
2.3	Chebyshev distance	9
3.1	The flow of the project	18
4.1	Behavior of ants	25
4.2	Demonstration for the principle of ACS	26
4.3	ACS algorithm	30
4.4	Process of shortening the long link	32
4.5	Code for immunity operator	33
4.6	Framework of IACS	34
4.7	Core algorithm of IACS	35
5.1	Screen shot of the simulator after running eil76	38
5.2	Comparison between ACS and IACS at 200 th iteration and beta=5 for eil76	40
5.3	Screen shot of the simulator after running pr152	41
5.4	Comparison between ACS and IACS at 200 th iteration and beta=5 for pr152	43
5.5	Screen shot of the simulator after running kroA200	44
5.6	Comparison between ACS and IACS at 200 th iteration and beta=5 for kroA200	46

LIST OF ABBREVIATIONS

TSP	-	Traveling Salesman Problem
ACS	-	Ant Colony System
ACO	-	Ant Colony Optimization
IACO	-	Immunity Ant Colony Optimization
IACS	-	Immunity Ant Colony System
AS	-	Ant System
EAS	-	Elitist Ant System
MMAS	-	MAX-MIN Ant System
GA	-	Genetic Algorithm
EA	-	Evolutionary Programming
SA	-	Simulated Annealing

CHAPTER I

INTRODUCTION

1.1 Project Background

The traveling salesman problem (TSP) is the problem faced by a salesman who wants to find a shortest possible trip through a given set of cities, starting from his home town and visiting each city once before finally returning home. A complete weighted graph $G = (N, A)$ can be used to represent the TSP, with N being the set of nodes representing the cities and A being the set of arcs fully connecting the nodes. Each arc is assigned a weight d_{ij} which represents the distance between cities i and j (Marco Dorigo, Thomas Stutzle, 2006).

In other words, TSP is the problem of finding a minimum length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed walk (a tour) visiting each node of G exactly once. The shortest possible trip through a given set of cities can be found using optimization techniques, and one of the most popular techniques among them is ACS.

Ant Colony System (ACS) is a popular optimization method used in Travelling Salesman Problem (TSP), and it is proven that this method can produce better solution than other optimization methods (Marco Dorigo, Luca Maria Gambardella, 1997). However, one underlying weakness is spotted.

1.2 Problem Statement

In ACS, there is no protection against the generation of bad solution. Since the mechanism of tour generation is totally based on randomized selection of nodes for starting city and a random value chosen for the pseudorandom proportional rule, there are chances that the ants choose a rather long path when constructing the tour. These long journeys may or may not be shortened in the subsequent iterations, and even the algorithm is capable of shortening the journey, more iterations are a must and this means slower convergent speed (Yingzhou Bi, Lixin Ding, Jianbo Lu, 2009).

It is well known that human body has immune system to serve as a protection against the attack of diseases, so it will be better if this protector is used when seeking optimum solution in problems like TSP. Hence, in order to improve the constructed solutions in TSP, a new method must be proposed to analyze the invalid components of the solutions and then repair them. This kind of protection cannot avoid the algorithm from generating a bad solution, but it can know by checking when a bad solution is generated, and it will repair it. So this is the reason why this kind of immunity should be added into the algorithm.

Research based on ACO and IACO has been done before, and the result shows that IACO outperforms ACO (Yingzhou Bi, Lixin Ding, Jianbo Lu, 2009). This is the first research done on ant based algorithm that uses the same concept based on immune system in human body. But we still do not know whether the immunity operator in IACO is applicable in other ant colony method other than ACO.

1.3 Project Objective

The objectives of this project are:

- (a) To propose an embedded immunity operator in Ant Colony System (ACS) algorithm

- (b) To design the new IACS algorithm
- (c) To develop and implement IACS algorithm in a simple simulator
- (d) To compare and evaluate the performance of IACS against ACS algorithm

1.4 Project Scope

The scope of this project is Travelling Salesman Problem. The modules in this project are the algorithms, ACS and IACS. These algorithms as well as the simulator will be implemented in Microsoft Visual C# language. The simulator will have a function that enables users to load different set of TSP data, which must be a .txt file. The selected dataset in this research are eil76.txt, pr152.txt, and kroA200.txt. There are four main parameters in the system: relative influence of pheromone trails (α), relative influence of heuristic information (β), number of ants and number of iterations. Value of α and number of ants is set to 1 and 10 during compilation, while value of β and number of iterations can be changed by users during run time.

1.5 Project Significant

Since this project is based on ant based algorithm for solving Travelling Salesman Problem, researchers who are focusing in this area will be benefited. Optimization problem other than TSP can also utilize the proposed algorithm in this project for seeking an optimum solution. Researchers of ant based algorithm may get inspiration on inventing an even better ant algorithm, or a totally new optimization technique. A more important part is that, other than a new modified algorithm, a simple simulator will also be developed. Although this simulator will be a very simple one, it is enough to simulate the TSP and run with two different algorithms. The interface of the simulator is not an important issue in this project. However, it will be made as user friendly as possible. A great advantage of this application is that

it can be improved from time to time. More algorithms and functions can be added if one knows how to utilize the Microsoft's .NET framework.

1.6 Expected Output

The expected output of this project can be separated into three main areas. The first will be the newly created algorithm, which is an advanced version of ACS. This proposed algorithm is expected to perform better than the current ACS. The second will be a simple application, which can simulate the TSP and test the two algorithms discussed earlier. The application must come with an interface that allows users to make alteration on some parameters. The third will be the experimental result obtained after running the algorithm using the application. The result will be compared and analyzed.

1.7 Conclusion

As a conclusion, this project aims to improve the state-of-the-art algorithm for solving TSP. A new modified algorithm will be proposed to run the TSP datasets. Besides that, a simple application will be developed to ease the implementation and testing of the algorithm. Result will be obtained by running the algorithms using this application. In the next chapter, the review on problem domain and various techniques will be done.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

In this chapter, the literature part of this research will be presented and discussed, including the in depth evaluation of previous research. This includes the study on the travelling salesman problem and the optimization method for finding the optimum solution. The optimization methods will be discussed and compared.

2.2 Traveling Salesman Problem (TSP)

The traveling salesman problem (TSP) is the problem faced by a salesman who, starting from his home town, wants to find a shortest possible trip through a given set of customer cities, visiting each city once before finally returning home. The TSP can be represented by a complete weighted graph $G = (N, A)$ with N being the set of nodes representing the cities, A being the set of arcs fully connecting the nodes. Each arc is assigned a weight d_{ij} which represents the distance between cities i and j . The TSP is the problem of finding a minimum length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed walk (a tour) visiting each node of G exactly once (Laporte, 1992).

A solution to an instance of the TSP can be represented as a permutation of the city indices. This permutation is cyclic, that is, the absolute position of a city in a tour is not important at all but only the relative order is important. In other words,

there are n permutations that map to the same solution (Bernardete Ribeiro et al., 2005).

2.2.1 Types of Traveling Salesman Problem

Before we proceed to various kinds of techniques that can be used for solving TSP, we will first discuss about the types of TSP. I was asked a question during my presentation for this project - is there any problems other than TSP can be solved using these algorithms? The answer is definitely yes, and even TSP itself has a few variants that can represent other similar problems (Marco Dorigo, Thomas Stutzle, 2006). Different types of TSP have different conditions or rules, and each of them can be used to represent certain kinds of problems. So, the understanding of different types of TSP is important if anyone intend to solve other problems using the algorithms in this project.

Generally, TSP can be divided into two main streams, the asymmetric TSP and symmetric TSP. The main factor that differentiates between these two TSP is that one is direction dependent and another is not. In other words, the solution for the problem is a one way tour if it is direction dependent. All the paths that connect the cities in the problem can only be traversed in one direction. In this condition, there is at least one edge for which $d_{ij} \neq d_{ji}$, which means the distance from city i to city j is different from the distance from city j to city i , as the distance between the pair of nodes i, j is dependent on the direction of traversing the edge. This is called an asymmetric TSP (Michael Hahsler, Kurt Hornik, 2009). It can be modeled as a directed weighted graph such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's length. The asymmetric TSP can model problems such as one-way streets and air-fares that depend on the direction of travel.

On the other hand, a symmetric TSP allows the paths to be traversed in two directions. Therefore, $d_{ij} = d_{ji}$ holds for all the edges in the solution, meaning that the

distance between cities i and j is the same in each direction, thus each tour has the same length in both directions. It can be modeled as an undirected weighted graph.

TSP can also be grouped into metric TSP or non-metric TSP. In the metric TSP the intercity distances satisfy the triangle inequality. The triangle inequality ensures that a direct path between two vertices in a triangle is at least as short as any indirect path. This is called the triangle inequality because no one side of a triangle can be longer than the sum of the other two (Cowen, 2009). This can be understood as the absence of shortcuts, in the sense that the direct connection from A to B is never longer than the detour via intermediate C. Figure 2.1 below shows the examples of triangle inequality.

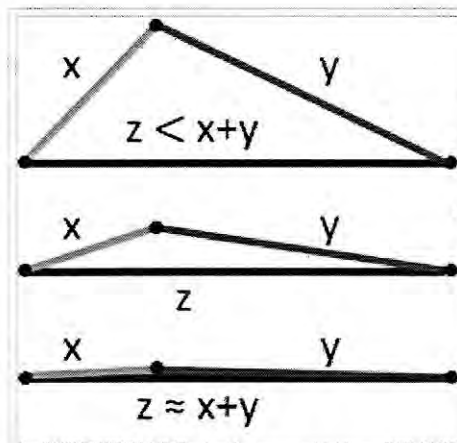


Figure 2.1 Triangle inequalities

Picture above shows three examples of triangle inequality for triangles with sides of lengths x , y , and z . The top example shows the case when there is a clear inequality and the bottom example shows the case when the third side, z , is nearly equal to the sum of the other two sides $x + y$.

Metric TSP can be further divided into three categories: Euclidean TSP, rectilinear TSP, and maximum metric TSP. In Euclidean TSP, the weight of edges

corresponds to ordinary (Euclidean) distances between their endpoints (cities) in the plane (Zachariasen, 2006). The Euclidean distance is the “ordinary” distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. The Pythagorean formula states that the distance between a pair of points can be calculated as the square root of the sum of a^2 and b^2 , where a equals to the difference of x-coordinates of the points and b equals to the difference of y-coordinates of the points. In rectilinear TSP, the distance between two cities is the sum of the differences of their x-coordinates and y-coordinates. Rectilinear distance is also known as taxicab distance, city block distance, or Manhattan distance. The latter names allude to the grid layout of most streets on the island of Manhattan, a densely-populated metropolitan city in United States, which causes the shortest path a car could take between two points in the borough to have length equal to the rectilinear distance. Euclidean distance is not applicable in this case, because if it is to be used in this situation, a vehicle must first crash through buildings before reaches the destination (Weisstein, 2011). See figure 2.2 for more detail.

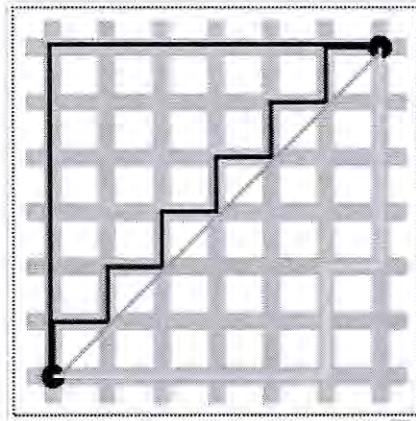


Figure 2.2 Manhattan distance and Euclidean distance

The picture above best describes the difference between Euclidean distance and rectilinear distance. In the picture, rectilinear distance is represented by red, blue, and yellow lines. All these lines have the same length (12) for the same route, which is the distance between the two black dots in this case. The green line represents the Euclidean distance, and is the unique shortest path among all.

For the maximum metric TSP, the distance between two cities is the maximum of the differences of their x-coordinates and y-coordinates. The distance is also known as Chebyshev distance or chessboard distance. In the game of chess the minimum number of moves needed by a king to go from one square on a chessboard to another equals the Chebyshev distance. The Chebyshev distance between two spaces on a chess board gives the minimum number of moves a king requires to move between them (Folio, 2008). Figure 2.3 below shows the Chebyshev distances of each square from the square f6.


	a	b	c	d	e	f	g	h
8	5	4	3	2	2	2	2	5
7	5	4	3	2	1	1	1	2
6	5	4	3	2	1		1	2
5	5	4	3	2	1	1	1	2
4	5	4	3	2	2	2	2	4
3	5	4	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4
1	5	5	5	5	5	5	5	5
	a	b	c	d	e	f	g	h

Figure 2.3 Chebyshev distance

The Chebyshev distance is sometimes used in warehouse logistics.

On the other hand, there are commonly occurring examples on non-metric TSP, in which the intercity distances do not satisfy triangle inequality. For example, there is no reason to expect the input would satisfy the triangle inequality if the weights of edges represent something like the cost of airline tickets. It is often the case that taking a direct route is more expensive than an indirect route (Cowen, 2009). Other examples include problems with one-way streets, traffic collisions, and transportation cost for cities with different departure and arrival fees.

2.3 Techniques for solving TSP

This part will cover the techniques that considered suitable for solving TSP.

2.3.1 Ant Colony System

ACS (Ant Colony System) was introduced by Dorigo and Gambardella in 1997. This distributed algorithm was first applied to the traveling salesman problem (TSP), based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems (Marco Dorigo, Luca Maria Gambardella, 1997). ACS is different from the previous Ant System (AS), which is the progenitor of ACS and was proposed by Dorigo in his Ph.D. dissertation in 1992. There are three main aspects that distinguish ACS from AS: (i) the state transition rule provides a direct way to balance between exploration of new edges and exploitation of a priority and accumulated knowledge about the problem, (ii) the global updating rule is applied only to edges which belong to the best ant tour, and (iii) while ants construct a solution a local pheromone updating rule is applied (Z. Ismail, S. L. Loh, 2009).

2.3.2 Ant System

The “ancestor” of ACS, as mentioned earlier, Ant System (AS) is also the first ever ACO algorithm. It was introduced by Dorigo in 1991 using the TSP as an example application. AS was able to achieve encouraging results in the beginning, however, it was found to be inferior to state-of-the-art algorithms for the TSP latter (Blum, 2005). From then on, the importance of AS lies mainly in the inspiration it provided for a number of extensions that have significant improvement in performance. In fact, most of these extensions or direct successor of AS keep the same solution construction procedure as well as the same pheromone evaporation procedure like that of AS. These extensions include Elitist AS, rank-based AS, Max-min AS, and Ant Colony System. The main difference between AS and its extensions

are how the pheromone trails are managed and the way the pheromone update is performed. Table 2.1 shows the chronological development of ACO algorithm.

Table 2.1 Chronological development of ACO algorithm (Dorigo, 2009)

Algorithm	Authors	Year
Ant System (AS)	Dorigo, Maniezzo, Colomi	1991
Elitist AS	Dorigo	1992
Ant Colony System (ACS)	Dorigo, Gambardella	1996
MAX-MIN Ant System	Stutzle, Hoos	1996
Rank-based Ant System	Bullnheimer, Hartl, Strauss	1997

Initially, Dorigo had proposed three different versions of AS, namely ant-density, ant-quantity, and ant-cycle. In the ant-density and ant-quantity versions, the pheromone was updated immediately after the ants move from one city to an adjacent city, while in ant-cycle version the pheromone was updated only after all the ants had constructed the tours (Krohn, 2002). However, due to the inferior performance of ant-density and ant-quantity, these two variants were abandoned. So, one actually refers to ant-cycle when referring to AS nowadays. The notable disadvantage of AS is that its performance decreases dramatically as the size of the test-instance increases (Marco Dorigo, Thomas Stutzle, 2006).

2.3.3 Elitist Ant System

Elitist Ant System (EAS) was introduced by Dorigo in 1992, and was the first improvement made on the initial AS. In Elitist model, strong additional reinforcement is provided to the arcs belonging to the best tour found since the start of the algorithm. This additional feedback to the best-so-far tour can be viewed as additional pheromone deposited by an additional ant called best-so-far-ant (Ayan Acharya, Deepyaman Maiti, Arita Banerjee, Amit Konar, 2008). The disadvantage of EAS is discussed in section 2.3.5.