

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

การอิมพลีเมนต์อัลกอริทึมจำลองอาณานิคม

IMPLEMENTATION OF ANT COLONY OPTIMIZATION ALGORITHM



H006651

โดย

อนน ชำนาญเวช

ANON CHAMNANWET

อาจารย์ที่ปรึกษา

ผศ.ดร.พรฤดี เนติโสภาค

เลขหมู่.....  
เลขทะเบียน..... 6651  
วัน,เดือน,ปี..... 11 ต.ค. 2555

b.....
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาการศึกษาศาสตร  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 2 ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IMPLEMENTATION OF ANT COLONY OPTIMIZATION ALGORITHM

ANON CHAMNANWET



A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

OF THE COURSE

INDEPENDENT STUDY

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2011**

**FACULTY OF THE INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ใบรับรองการศึกษาอิสระ (Independent Study)

เรื่อง

การอิมพลีเมนต์อัลกอริทึมจำลองอาณานิคมมด


## IMPLEMENTATION OF ANT COLONY OPTIMIZATION ALGORITHM


นายอนล ชำนาญเวช

รหัสประจำตัว 51066532

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด  
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของการ  
ศึกษาวิชาการศึกษาอิสระ หลักสูตรวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)  
ภาคเรียนที่ 2 ปีการศึกษา 2553

  
.....อาจารย์ที่ปรึกษา  
(ผศ.ดร. พรฤดี เนติโสภาค)

  
.....กรรมการสอบ  
(รศ.ดร. วรพงษ์ กรีสู่ระเดช)

  
.....กรรมการสอบ  
(รศ.ดร. อาริต ธรรมโน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ระบบจำลองการทำงานอัลกอริทึมของการจำลองแบบอาณาจักรมด
นักศึกษา	นายอนล ชำนาญเวช
รหัสนักศึกษา	51066532
ปริญญา	วิทยาศาสตร์มหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2553
อาจารย์ที่ปรึกษา	ผศ.ดร.พรฤดี เนติโสภาคกุล

### บทคัดย่อ

Ant colony optimization (ACO) เป็นการทำงานรูปแบบหนึ่งของปัญญาประดิษฐ์ (AI) ซึ่งเป็นวิธีการหาเส้นทางที่เหมาะสมที่สุด (สั้นที่สุด) ของแต่ละปัญหาเช่น ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem - TSP) โดยใช้วิธีเลียนแบบการออกหาอาหารของมดซึ่งเป็นวิธีการทำงานแบบกลุ่ม โดยมดแต่ละตัวจะช่วยกันออกหาเส้นทางและสุดท้ายจะได้เส้นทางจากรังไปยังแหล่งอาหารที่สั้นและมีประสิทธิภาพ

ระบบจำลองการทำงานอัลกอริทึมของการจำลองแบบอาณาจักรมด เป็นระบบที่จัดทำขึ้นเพื่อแสดงให้เห็นถึงการทำงานของอัลกอริทึม 3 แบบที่แตกต่างกันออกไป โดยนำปัญหาการเดินทางของพนักงานขาย (TSP) มาเป็นตัวอย่างในการทำงานของระบบซึ่งมีทั้งการแสดงผลการทำงานในรูปแบบกราฟฟิก และผลลัพธ์ที่ได้จากการทำงานในแต่ละครั้ง เพื่อให้เข้าใจการทำงานของอัลกอริทึมได้ดียิ่งขึ้น นอกเหนือจากส่วนของระบบแล้ว ยังมีการเปรียบเทียบการทำงานของอัลกอริทึม โดยจำลองการทำงานในรูปแบบที่ต่างกัน พร้อมทั้งแสดงข้อมูลการเปรียบเทียบการทำงานให้เห็นข้อแตกต่างและข้อจำกัดในการทำงานของอัลกอริทึมนั้น

<b>Title</b>	Implementation of Ant Colony Optimization Algorithm
<b>Student</b>	Mr. Anon Chamnanwet
<b>Student ID.</b>	51066532
<b>Degree</b>	Master of Science
<b>Program</b>	Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2010
<b>Advisor</b>	Asst. Prof. Dr. Ponrudee Netisopakul

## ABSTRACT

Ant colony optimization (ACO) is one of the Artificial Intelligence's (AI) working model, which is the model of finding the most proper and shortest way for problem solving. For example, Traveling Salesman Problem (TSP). This ACO model imitated the way ants seeking for their foods which usually go by group. Each of them will find the shortest and the best way from their home to the consuming source.

The implementation of Ant Colony Optimization Algorithm is implemented to compare efficiency of the three different working processes using Traveling Salesman Problem as examples. The examples and the results are presented in graphic for a better understanding. In addition, a comparison of algorithms is to show the differences and limitations of these working processes.

## กิตติกรรมประกาศ

โครงการในวิชาการศึกษาระดับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร.พรฤดี เนติโสภากุล ซึ่งเป็นอาจารย์ผู้ควบคุมโครงการ และคณาจารย์คณะเทคโนโลยีสารสนเทศ ที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้าเป็นอย่างสูง

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำโครงการฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์อันพึงมาจาก โครงการฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

อนล ชำนาญเวช



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VI
สารบัญรูป .....	VII
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา .....	1
1.3 ขอบเขตของการศึกษา .....	1
1.4 ขั้นตอนของการศึกษา .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 Ant Colony Optimization .....	3
2.2 Traveling Salesman Problem .....	4
2.3 Ant Colony Optimization Metaheuristic .....	5
2.4 Ant Colony Optimization Algorithm .....	7
บทที่ 3 วิเคราะห์และออกแบบระบบ	
3.1 การวิเคราะห์ระบบ.....	10
3.2 การออกแบบระบบ .....	12
3.3 Class Responsibility Collaborator (CRC) .....	18
3.4 Class Diagram .....	19
3.5 Sequence Diagram.....	21

# สารบัญ (ต่อ)

	หน้า
บทที่ 4 การวิเคราะห์และออกแบบระบบ	
4.1 การกำหนดตัวแปรที่เกี่ยวข้อง.....	26
4.2 ผลลัพธ์จากการทำงานของระบบ.....	27
4.3 หน้าจอสำหรับการใช้งาน.....	28
4.4 ตัวอย่างการใช้งาน.....	29
4.5 การเปรียบเทียบการทำงานของอัลกอริทึม.....	32
บทที่ 5 บทสรุปและข้อเสนอแนะ	
5.1 สรุปผลการพัฒนาระบบ.....	42
5.2 ข้อจำกัดในการเปรียบเทียบการทำงานของอัลกอริทึม.....	42
บรรณานุกรม.....	44
ประวัติผู้เขียน.....	45

# สารบัญตาราง

ตารางที่	หน้า
4.1 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 1.....	33
4.2 ผลลัพธ์ของ Min-Max Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 1 .....	34
4.3 ผลลัพธ์ของ Ant Colony System สำหรับการจัดวางโหนดแบบที่ 1.....	34
4.4 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 2.....	35
4.5 ผลลัพธ์ของ Min-Max Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 2 .....	36
4.6 ผลลัพธ์ของ Ant Colony System สำหรับการจัดวางโหนดแบบที่ 2.....	36
4.7 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 3.....	37
4.8 ผลลัพธ์ของ Min-Max Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 3 .....	38
4.9 ผลลัพธ์ของ Ant Colony System สำหรับการจัดวางโหนดแบบที่ 3.....	39
4.10 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวางโหนดแบบที่ 1.....	39
4.11 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวางโหนดแบบที่ 2.....	40
4.12 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวางโหนดแบบที่ 3.....	41

# สารบัญรูป

รูปที่	หน้า
2.1 The Double Bridge Experiment.....	4
2.2 การเติบโตของ function แบบ exponential time.....	5
2.3 ตัวอย่างการใช้ Brute Force Algorithm ในการแก้ปัญหา TSP.....	5
2.4 ACO Algorithm (Pseudo Code) .....	6
3.1 ตัวอย่างความสัมพันธ์ระหว่างโหนด.....	10
3.2 โครงสร้างข้อมูลของระบบ .....	12
3.3 ชูโดโค้ดโดยรวมของระบบ .....	13
3.4 ชูโดโค้ดของการสร้างวิธีการแก้ไขปัญหา.....	13
3.5 ชูโดโค้ดของการตัดสินใจในการเดินทางไปยังโหนดถัดไป.....	14
3.6 ชูโดโค้ดของการเลือกโหนดที่ดีที่สุดสำหรับการเดินทาง .....	14
3.7 ชูโดโค้ดของการปรับปรุงค่าฟีโรโมน.....	15
3.8 ชูโดโค้ดของการลดค่าฟีโรโมน .....	16
3.9 ชูโดโค้ดของการเพิ่มค่าฟีโรโมน .....	16
3.10 ชูโดโค้ดของการปรับปรุงค่าความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไป .....	17
3.11 ชูโดโค้ดของการปรับปรุงค่าฟีโรโมนของ Ant Colony System .....	17
3.12 Class Responsibility Collaborator (CRC) ของระบบ .....	18
3.13 Class diagram ของระบบ .....	19
3.14 Sequence ของการทำงาน Initialize data .....	21
3.15 Sequence ของการทำงาน Construction solution.....	22
3.16 Sequence ของการทำงาน Update pheromone.....	24
4.1 ผลลัพธ์จากการทำงานของระบบ .....	28
4.2 หน้าจอสำหรับการใช้งาน .....	28
4.3 แสดงการสร้างโหนดและกำหนดค่าเริ่มต้น .....	29
4.4 ระบบเริ่มทำงาน .....	30
4.5 ระบบหยุดทำงาน .....	30
4.6 แสดงหน้าจอเมื่อกดปุ่ม Clear Statistics .....	31
4.7 แสดงหน้าจอเมื่อกดปุ่ม Clear Screen .....	31
4.8 รูปแบบการจัดวางโหนดแบบง่าย.....	33

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 รูปแบบการจัดวางโหนดแบบปานกลาง.....	35
4.10 รูปแบบการจัดวางโหนดแบบยาก.....	37



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การศึกษาเกี่ยวกับเทคโนโลยีสารสนเทศในปัจจุบันมีมากมายหลายสาขา มีทั้งที่เป็นเนื้อหาทั่วไปที่สามารถศึกษาทำความเข้าใจได้ง่ายหรือไม่ยากมากจนเกินไป และแบบที่ต้องใช้ความเชี่ยวชาญเป็นพิเศษในการศึกษา “ปัญญาประดิษฐ์ (Artificial Intelligence - AI)” ก็เป็นแขนงหนึ่งที่ต้องใช้ความเชี่ยวชาญเฉพาะทาง เพื่อที่สามารถเข้าใจการทำงานหรือนำไปประยุกต์ใช้ให้เกิดประโยชน์ได้ ด้วยเหตุนี้จึงทำให้ไม่ค่อยมีคนศึกษาอย่างจริงจังมากนัก

Ant Colony Optimization เป็นหนึ่งในรูปแบบของปัญญาประดิษฐ์ มีรูปแบบในการแก้ปัญหาหรืออัลกอริทึมหลายรูปแบบ โดยแต่ละรูปแบบก็มีวิธีการที่แตกต่างกันออกไปและมีข้อดีข้อเสียกันด้วยกันทั้งสิ้น ในการศึกษาที่ต้องการที่จะนำเสนอการทำงาน ACO ในการแก้ไขปัญหาค่าการเดินทางไปยังเมืองต่างๆ ของพนักงานขาย (Traveling Salesman Problem – TSP) เพื่อให้เห็นวิธีการทำงานที่ชัดเจน ผ่านทางโปรแกรม และนอกจากนั้นยังมีการเปรียบเทียบการทำงานของอัลกอริทึมทั้ง 3 รูปแบบเพื่อให้เห็นข้อดีและข้อเสียของแต่ละแบบด้วย

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อพัฒนาโปรแกรมโดยใช้หลักการของ Ant Colony Optimization
- 1.2.2 เพื่อให้เห็นภาพการทำงานของ Ant Colony Optimization ที่ชัดเจนและง่ายต่อการเข้าใจ
- 1.2.3 เพื่อเปรียบเทียบการทำงานของอัลกอริทึมดังนี้
  - 1.2.3.1. Ant System
  - 1.2.3.2. MAX-MIN Ant System
  - 1.2.3.3. Ant Colony System

### 1.3 ขอบเขตของการศึกษา

ในการเปรียบเทียบการทำงานของอัลกอริทึม จำเป็นต้องมีขอบเขตในการทำงานดังนี้

- 1.3.1 พัฒนาระบบเพื่อจำลองการทำงานของอัลกอริทึม โดยใช้ภาษา Java
- 1.3.2 ออกแบบการทดลองเพื่อเปรียบเทียบอัลกอริทึม โดยใช้ปัจจัยดังต่อไปนี้
  - 1.3.2.1. จำนวนมดที่ใช้ในการทดลอง 3 แบบ ต่อหนึ่งอัลกอริทึม
  - 1.3.2.2. จำนวนของ Node และรูปแบบการจัดวาง Node 3 แบบ ต่อหนึ่งอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

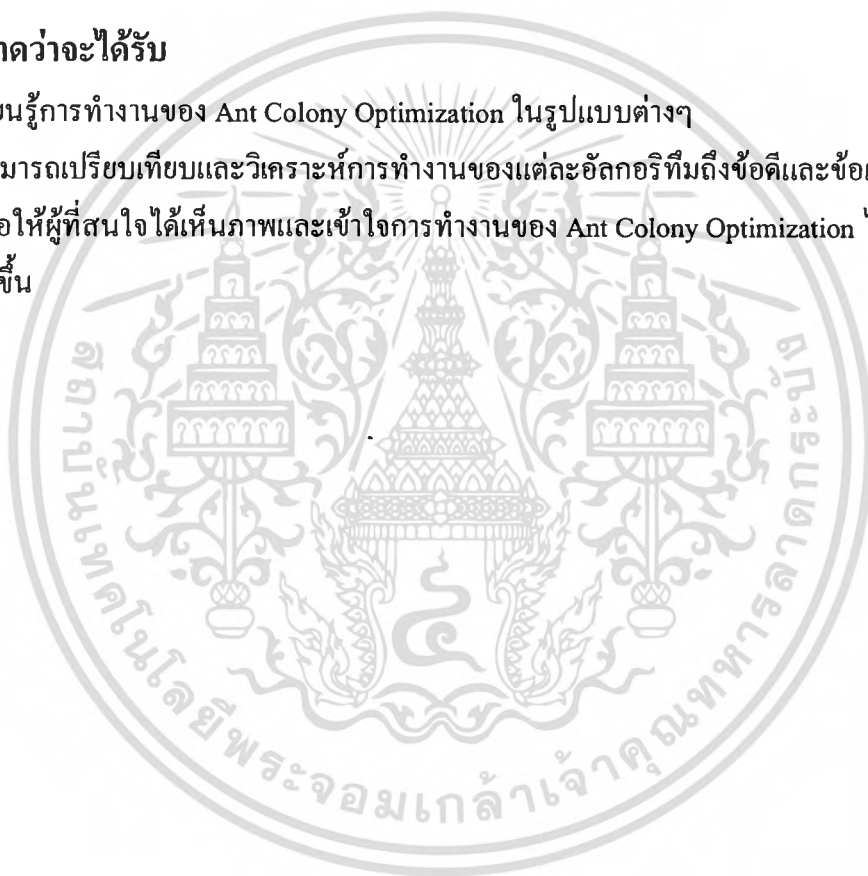
### 1.3.2.3. จำนวนอัลกอริทึมที่ใช้ 3 อัลกอริทึม

## 1.4 ขั้นตอนของการศึกษา

- 1.4.1. ศึกษาการวิธีการทำงานของ Ant Colony Optimization
- 1.4.2. ออกแบบระบบโดยอาศัยหลักการของ UML
- 1.4.3. พัฒนาระบบและทดสอบการทำงานของระบบ
- 1.4.4. เก็บข้อมูลและเปรียบเทียบการทำงานของแต่ละอัลกอริทึม

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1. เรียนรู้การทำงานของ Ant Colony Optimization ในรูปแบบต่างๆ
- 1.5.2. สามารถเปรียบเทียบและวิเคราะห์การทำงานของแต่ละอัลกอริทึมถึงข้อดีและข้อเสียได้
- 1.5.3. เพื่อให้ผู้ที่สนใจได้เห็นภาพและเข้าใจการทำงานของ Ant Colony Optimization ได้ชัดเจนยิ่งขึ้น



## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

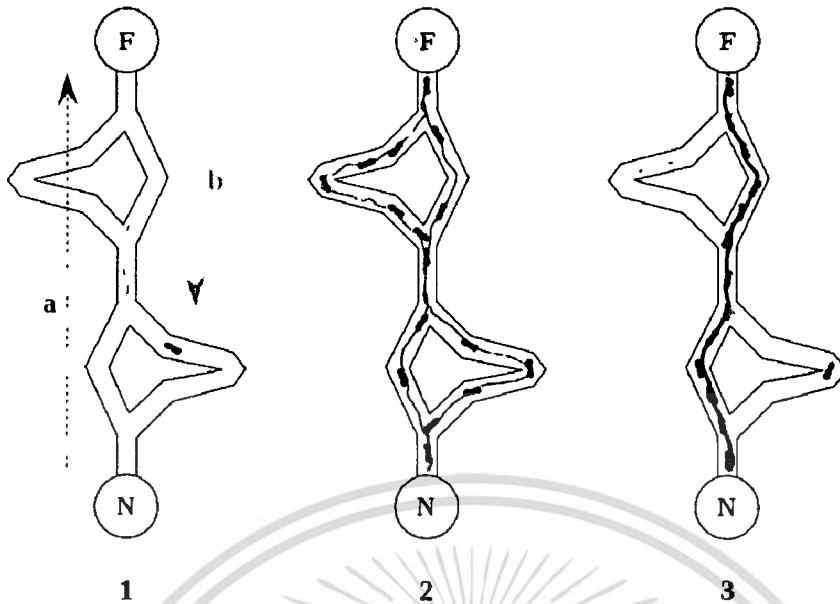
### 2.1 Ant Colony Optimization

ขั้นตอนวิธี ACO ที่ได้นำเสนอครั้งแรกโดย Marco Dorigo ในปี 1992 เรียกว่า Ant System (AS) ถูกสร้างขึ้นมาเพื่อเลียนแบบพฤติกรรมของ Real Ants (มดจริง) ซึ่งในช่วงแรกของการออกหาอาหารของมดนั้น มดในตอนแรกของการเริ่มออกหาอาหาร การเดินทางของมดแต่ละตัวจะมีลักษณะกระจายกันออกไป เส้นทางที่มดแต่ละตัวใช้จะถูกปล่อยสารฟีโรโมน (Pheromone) ไว้ด้วยมดตัวอื่นจะสามารถรับรู้ได้ถึงบริเวณที่มีสารฟีโรโมนที่มดตัวก่อนหน้าได้ปล่อยเอาไว้ บริเวณหรือเส้นทางใดที่มีมดเดินผ่านมาก ปริมาณของสารฟีโรโมนที่มดปล่อยออกมาก็จะมีมากตามไปด้วย ในทำนองเดียวกันในบริเวณที่มีมดเดินผ่านน้อย ปริมาณสารฟีโรโมนก็จะมีน้อยตาม จนท้ายที่สุดสารฟีโรโมนก็จะหายไปจากบริเวณหรือเส้นทางนั้นๆ เนื่องจากไม่มีมดตัวอื่นๆ มาปล่อยสารฟีโรโมนนั่นเอง ด้วยเหตุนี้ปริมาณสารฟีโรโมนจึงเป็นส่วนสำคัญที่ส่งผลกระทบต่อการตัดสินใจในการเดินทางหรือเลือกเส้นทางที่จะใช้ของมดตัวตัวอื่นๆ มดที่ตามหลังมานั้น จะพยายามเดินตามทางที่มีสารฟีโรโมนปล่อยอยู่ จนท้ายที่สุดแล้วจะเหลือเส้นทางที่เหมาะสมที่สุดจากวังของมดไปยังแหล่งอาหารเพียงเส้นทางเดียว

#### 2.1.1 Double Bridge Experiment

Double Bridge Experiment เป็นการทดลองที่แสดงให้เห็นถึงการการเดินทางของมดและความสัมพันธ์ของปริมาณสารฟีโรโมนกับการตัดสินใจเลือกทางเดินของมดได้อย่างชัดเจนวิธีหนึ่ง การทดลองจะเป็นการให้มดเดินจากรัง (Nest) ไปยังแหล่งอาหาร (Food) บนเส้นทางที่กำหนดไว้ เป็นทางเดินซึ่งเมื่อออกจากรังแล้ว จะมีทางแยก 2 ทางและทางแยกทั้ง 2 ทางจะมาบรรจบกันก่อนจะถึงแหล่งอาหาร ทางเดินหรือสะพานที่ใช้ในการทดลองนั้นจะมีอยู่ 2 แบบ โดยแบบที่ 1 จะเป็นทางเดินหรือสะพานที่มีความยาวของเส้นทางที่เท่ากันคือ ระยะทางของทางแยกที่แยกออกมาทั้งสองทางจะมีระยะทาง (ความยาว) เท่ากัน ส่วนแบบที่ 2 ระยะทางของทางแยกที่แยกออกมา จะมีทางใดทางหนึ่งที่ยาวกว่าอีกทางหนึ่ง

เมื่อมดเริ่มเดินทางออกจากรัง มดจะทำการสุ่มเส้นทางที่ใช้ไปถึงอาหาร การเดินทางของมดตัวนั้นจะมีการปล่อยสารฟีโรโมนออกมาเรื่อยๆ ไปจนถึงจุดที่พบอาหาร การเดินทางของมดตัวถัดไป จะเริ่มมีการเรียนรู้การเดินทางจากมดตัวก่อนหน้า โดยมดจะสังเกตปริมาณของฟีโรโมนบนเส้นทาง เส้นทางใดมีปริมาณฟีโรโมนที่มาก เส้นทางนั้นก็มีโอกาสสูงที่มดจะเลือกผ่านเส้นทางนั้น ส่วนเส้นทางใดที่มีปริมาณฟีโรโมนน้อย โอกาสที่มดจะเลือกเส้นทางนั้น ก็จะน้อยตาม



รูปที่ 2.1 The Double Bridge Experiment

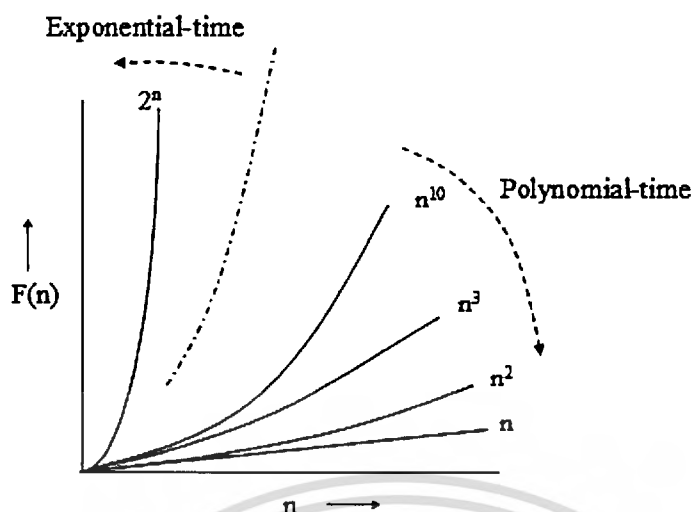
จากรูปที่ 2.1 เป็นการแสดงการทดลองบนสะพานคู่ที่มีความยาวไม่เท่ากัน (การทดลองนี้ไม่มีสะพานคู่ที่มีความยาวเท่ากันมาแสดงด้วย) จะเห็นได้ (1) มดจะเดินทางแบบสุ่มเลือกเส้นทางจากรังไปยังแหล่งอาหารและเริ่มมีการปล่อยสารฟีโรโมน (2) เมื่อมดมีจำนวนมากขึ้น ปริมาณสารฟีโรโมนก็จะมีอยู่ทั่วทุกเส้นทางที่มดเดินผ่าน ในเส้นทางที่มีระยะทางยาวกว่านั้นอัตราการระเหยของสารฟีโรโมนจะมีมากกว่าเส้นทางที่มีระยะทางสั้นกว่า เนื่องจากมดต้องใช้เวลาในการเดินทางนานกว่าและมดตัวถัดไปที่ใช้เส้นทางเดียวกันอาจปล่อยสารฟีโรโมนไม่ทันก่อนที่ปริมาณสารฟีโรโมนที่มีอยู่แล้วจะหมดลง (3) ท้ายที่สุดจะเหลือเพียงเส้นทางเดียวที่มดทั้งหมดใช้เดินทางจากรังไปยังแหล่งอาหาร เส้นทางนี้จะเป็นเส้นทางที่สั้นและใช้เวลาที่น้อยที่สุด

## 2.2 Traveling Salesman Problem

การแก้ปัญหา Traveling Salesman Problem (TSP) ซึ่งอยู่ในกลุ่ม NP-hard คือเป็นปัญหาที่ยากในการหาคำตอบ โดย ปัญหาของ TSP ก็คือการหาลำดับ (permutation) ของการเดินทางไปยังเมืองต่าง ๆ ทุกเมือง โดยไปเยี่ยมชมแต่ละเมืองได้เพียงครั้งเดียว แล้วเดินทางกลับมายังเมืองที่เริ่มต้น

ความยากของปัญหานี้อยู่ที่ เมื่อขนาดของจำนวนเมืองเพิ่มมากขึ้น ลำดับการเดินทางไปยังเมืองต่าง ๆ ที่เป็นไปได้ก็จะมีจำนวนมากขึ้น ในลักษณะการเติบโตของ function แบบ exponential time (รูปที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การเติบโตของ function แบบ exponential time

เพียงเราเพิ่มจำนวนของเมืองเข้าไปแค่ไม่กี่เมือง ในกรณีที่เราใช้ Algorithms ธรรมดา ๆ ยกตัวอย่าง เช่น Brute Force Algorithm คือพิจารณาทุก ๆ ผลลัพธ์ที่เป็นไปได้ (candidate solution) เราอาจจะต้องใช้ระยะเวลายาวนานจนบางที่ไม่สามารถรอคำตอบได้ ดังรูปที่ 3 เมื่อมีการเพิ่มจำนวนเมือง จำนวนครั้งที่จะต้องใช้ในการเดินทางให้ครบทุกเมืองก็จะเพิ่มขึ้นเป็นทวีคูณ

City number	Number of Steps
1	1
2	1
3	6
4	24
5	120
6	720
7	5,040
8	40,320
9	362,880
10	3,628,800
11	39,916,800
12	479,001,600
13	6,227,020,800
...	...
50	$3.041 * 10^{64}$

รูปที่ 2.3 ตัวอย่างการใช้ Brute Force Algorithm ในการแก้ปัญหา TSP

### 2.3 Ant Colony Optimization Metaheuristic

Artificial ant ถูกสร้างขึ้นเพื่อใช้แก้ปัญหา combinatorial optimization โดยมันจะเดินทางเชื่อมกันจนเกิดเป็นกราฟ ตัวแปรในการตัดสินใจ  $X^i = v_j^i$  หรือ solution component (ให้เซตของความน่าจะเป็นที่ เกิดขึ้นจาก solution component คือ  $C$  กราฟ  $G_C(V, E)$  จะถูกสร้างขึ้นจากความสัมพันธ์ของ ส่วนประกอบใน  $C$  โดย  $V$  คือ เซตของจุดสูงสุด  $E$  คือ เซตของขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการในการแก้ไขปัญหของ ACO ก็คือมจะเดินทางจากจุดสูงสุดไปยังจุดสูงสุด (vertex) ภายใต้ขอบเขต (edges) จนเกิดเป็นกราฟ ปริมาณฟีโรโมนจะถูกเพิ่มขึ้นเมื่อมเดินทางผ่านจุดสูงสุดหรือขอบเขตของกราฟ ค่าฟีโรโมนที่ถูกสะสม ( $\Delta\tau$ ) จะบ่งบอกถึงคุณภาพของวิธีการแก้ปัญหที่หาได้ มตัวถัดไปก็จะใช้ค่าฟีโรโมนเป็นข้อมูลในการเดินทางเพื่อค้นหาพื้นที่ต่อไป ซึ่งคล้ายกับวิธีการที่มจริง (real ant) ใช้ในการเดินทางจากยังไปยังอาหาร

```

Set parameters, initialize pheromone trails
SCHEDULE_ACTIVITIES
  ConstructAntSolutions
  DaemonActions {optional}
  UpdatePheromones
END SCHEDULE_ACTIVITIES

```

รูปที่ 2.4 ACO Algorithm (Pseudo Code)

การแก้ปัญหของ ACO จะเริ่มต้นด้วยการกำหนดค่าต่างๆ (initialize) แล้วจะเริ่มวนลูป ซึ่งภายในลูปจะประกอบด้วย algorithm ย่อยภายในอีกสามตัวคือ ConstructAntSolutions, DaemonActions (optional) และ UpdatePheromone ซึ่งมีรายละเอียดดังนี้

### 2.3.1 ConstructAntSolutions

เซตของ artificial ant จำนวน  $m$  ตัวจะสร้างคำตอบที่เป็นไปได้จากส่วนของคำตอบ  $C = \{c_{ij}\}$  โดยที่  $i = 1, \dots, n$  และ  $j = 1, \dots, |D_i|$  การสร้างคำตอบนั้นจะเริ่มจากไม่มีคำตอบ (empty solution)  $S^P$  หลังจากนั้น ในแต่ละขั้นตอนของการสร้างคำตอบ (partial solution)  $S^P$  จะถูกเพิ่มด้วยส่วนประกอบของคำตอบจากส่วนที่ใกล้เคียงกันโดยที่  $N(S^P) \subseteq C$  และจะนำความสัมพันธ์มาสร้างเป็นกราฟ โดยตัวเลือกของส่วนของคำตอบจาก  $N(S^P)$  จะเสร็จสมบูรณ์ในรอบการทำงานถัดไป

### 2.3.2 DaemonActions

เมื่อมีการปรับค่าฟีโรโมนหลังจากการสร้างคำตอบใดคำตอบหนึ่ง อาจมีปัญหบางอย่างที่กระทำเกิดขึ้น เรียกว่า *daemon action* ที่ต้องใช้วิธีการแก้ปัญหโดยเฉพาะ ปัญหเหล่านี้ไม่สามารถแก้ไขได้ด้วยมเพียงตัวเดียว อาจจะต้องใช้วิธี local search ในการแก้ปัญหหลังจากที่มีการปรับค่าฟีโรโมนแล้ว

### 2.3.3 UpdatePheromone

จุดมุ่งหมายของการอัปเดตฟีโรโมน คือการเพิ่มค่าของ ฟีโรโมนที่ได้จากการหาเส้นทางที่ดีที่สุด และลดค่าฟีโรโมนที่ได้จาเส้นทางที่ไม่ดีนัก โดย

1. ลดค่าของฟีโรโมนในเส้นทางที่ไม่ดี (ไม่มีมดเดินผ่านหรือมีน้อย) เทียบได้กับการระเหยของฟีโรโมนในเส้นทางที่มดจริงใช้เดิน
2. เพิ่มระดับ ฟีโรโมนให้กับเส้นทางที่ดี (เส้นทางที่มีมดใช้เดินทางมาก) โดยเลือกเลือกใช้วิธีการที่เหมาะสม

## 2.4 Ant Colony Optimization Algorithm

ACO algorithms ที่ถูกนำไปใช้มีมากมายหลายแบบ แต่ทั้งหมดล้วนมีพื้นฐานเดียวกัน การทำงานของ ACO คือ เริ่มจากมดแต่ละตัวจะเดินทางจากจุดหนึ่งไปยังจุดหนึ่งหรือจากจุดๆ หนึ่งไปยังจุดสูงสุดภายใต้ขอบเขตที่กำหนดไปเรื่อยๆ จนเกิดเป็นกราฟ ปริมาณของฟีโรโมนจะถูกเพิ่มขึ้นและถูกสะสมเมื่อมดเดินทางผ่านจุดเดิม ค่าของฟีโรโมนที่ถูกสะสมจะบ่งบอกถึงคุณภาพของเส้นทางที่มดใช้ในการเดินทาง จากนั้นมดตัวต่อๆ ไปก็ใช้ค่าฟีโรโมนเป็นข้อมูลในการตัดสินใจเดินทาง (ภายในขอบเขตที่กำหนด) ต่อไป ACO algorithms น่าสนใจและใช้กันอย่างแพร่หลายมีดังนี้

### 2.4.1 Ant System

Ant system (AS) เป็น ACO ตัวแรกที่ถูกนำเสนอ โดยมีลักษณะการทำงานคือ ค่าฟีโรโมนจะถูกปรับปรุงโดยมดทุกตัวที่เดินทางครบทุกเส้นทาง ส่วนของคำตอบ (solution component)  $C_{ij}$  คือขอบเขตของกราฟและ  $\tau_{ij}$  ใช้สำหรับปรับปรุงฟีโรโมน  $i$  คือฟีโรโมนความสัมพันธ์ระหว่างขอบเขตของเมืองที่ติดต่อกันและ  $j$  คือฟีโรโมนที่ถูกปรับปรุงครั้งสุดท้าย

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (2.1)$$

โดยที่

$\rho \in (0, 1]$  คือ อัตราการระเหยของฟีโรโมน

$m$  คือ จำนวนมดทั้งหมด

$\Delta \tau_{ij}^k$  คือ ปริมาณฟีโรโมนที่ขอบเขตของเมือง  $(i, j)$  ที่มด  $k$  ปลดปล่อยออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จริงแล้ว ระบบมดเดิมได้มีอยู่ 3 แบบด้วยกันคือ Ant - density, Ant - quantity และ Ant - cycle ระบบ Ant - density และระบบ Ant - quantity นั้นจะมีการอัปเดตสารฟีโรโมนทันทีขณะที่เดินทางจากโหนดหรือเมือง (Node or city) ไปยังเมือง ขณะที่ระบบ Ant - cycle นั้นจะอัปเดตสารฟีโรโมนหลังจากที่มดเดินทางครบทุกเมืองแล้ว โดยที่ปริมาณของสารฟีโรโมนที่จะอัปเดตนั้นขึ้นอยู่กับอัตราส่วนระหว่างค่าคง ที่ต่อระยะทางหรือคุณภาพของผลเฉลยที่ได้ ทำยที่สุดแล้วระบบ Ant - density และระบบ Ant - quantity ก็ไม่ได้รับการปรับปรุงและพัฒนาต่อไปอีก เนื่องจากมีประสิทธิภาพในการหาผลเฉลยหรือเส้นทางที่น้อยมากเมื่อเทียบกับ ระบบ Ant - cycle ดังนั้นในปัจจุบัน เมื่อก้าวถึงระบบมด ก็คือระบบ Ant - cycle นั้นเอง

#### 2.4.2 Ant Colony System

จุดที่แตกต่างจาก ant system จุดแรกคือรูปแบบการตัดสินใจของมดในระหว่างการประมวลผล มดใน ant colony system จะใช้กฎ pseudorandom proportional คือการที่มดเดินทางจากเมือง  $i$  ไปเมือง  $j$  จะขึ้นอยู่กับค่าตัวแปรแบบสุ่มประ โยชน์ที่น่าสนใจของข้อมูลฟีโรโมนใน ACS คือ การทำ local pheromone update ซึ่งจะถูกระทำโดยมดทุกตัวหลังจากการทำงานในแต่ละขั้นตอน (การสร้างคำตอบ) มดแต่ละตัวใช้ค่าสุดท้ายของขอบเขตที่เดินทางเท่านั้น

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau + \rho \cdot \Delta \tau_{ij}^{best} \quad (2.2)$$

โดยที่

$\Delta \tau_{ij}^{best} = 1/L_{best}$  ถ้ามดที่ดีที่สุดใช้ขอบเขตเส้นทาง  $(i, j)$

$\Delta \tau_{ij}^{best} = 0$  นอกเหนือจากกรณีแรก

$L_{best}$  สามารถปรับปรุงได้หากพบการเดินทางที่ดีที่สุดระหว่างขอบเขตในแต่ละรอบ

Ant Colony System มีหลักการการทำงานอยู่ 3 หลักการใหญ่ๆ คือ

1. ACS จะ พัฒนาในส่วนของการจำเส้นทางในการเดินของมด โดยจะทำให้มดมีประสบการณ์ในการจำเส้นทางมากขึ้นและจะมีผลต่อการตัดสินใจในการเลือกเส้นทางมากขึ้นด้วย
2. การระเหยของฟีโรโมนและการวางฟีโรโมนจะทำในส่วนที่เป็นเส้นทางที่ดีที่สุดเท่านั้น
3. ในแต่ละเส้นทางที่มดเดินผ่านไปนั้น มดจะเอาฟีโรโมนออก เพื่อที่จะทำให้เกิดการเพิ่มเส้นทางหรือโอกาสในการเลือกเส้นทางอื่น

### 2.4.3 MAX-MIN Ant System

เป็นแนวความคิดที่ปรับปรุงมาจาก Ant System ซึ่งมีจุดต่างจาก Ant System คือ

1. มดตัวที่ดีที่สุดเท่านั้นจึงจะได้ปรับปรุงค่าฟีโรโมน
2. การกำหนดช่วงของฟีโรโมนให้อยู่ในช่วงที่สมการกำหนด เพื่อที่เราจะได้จำกัดขอบเขตของเส้นทางที่ดีที่สุดเพียงหนึ่งช่วงเท่านั้น ทำให้หาเส้นทางที่ดีที่สุดได้อย่างรวดเร็ว
3. ค่าฟีโรโมนเริ่มต้นจะมีค่าตัวแปรการระเหยของปริมาณฟีโรโมนไว้ในตอนแรกเลย
4. ถ้าปริมาณฟีโรโมนเริ่มต้น เริ่มมีค่าคงที่หรือไม่มีการเพิ่มขึ้นแล้วก็จะสร้างจำนวนรอบที่แน่นอนสำหรับการคำนวณครั้งต่อไป

วิธีการปรับปรุงค่าฟีโรโมนของ MAX-MIN Ant System

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau + \rho \cdot \Delta \tau_{ij}^{best} \quad (2.3)$$

โดยที่

$$\Delta \tau_{ij}^{best} = 1/L_{best} \text{ ถ้ามดที่ดีที่สุดใช้ขอบเขตเส้นทาง } (i, j)$$

$$\Delta \tau_{ij}^{best} = 0 \text{ นอกเหนือจากกรณีแรก}$$

ค่าของฟีโรโมนจะถูกตรวจสอบให้อยู่ระหว่าง  $\tau_{min}$  และ  $\tau_{max}$  หลังจากที่มีมดได้ปรับปรุงค่าฟีโรโมนทั้งหมด โดยที่  $\tau_{ij}$  จะเท่ากับ  $\tau_{min}$  ก็ต่อเมื่อ  $\tau_{ij} < \tau_{min}$  และ  $\tau_{ij}$  จะเท่ากับ  $\tau_{max}$  ก็ต่อเมื่อ  $\tau_{ij} > \tau_{max}$

## บทที่ 3

# วิเคราะห์และออกแบบระบบ

การพัฒนาระบบนั้นสิ่งที่จะต้องทำอย่างหนึ่งคือการวิเคราะห์และออกแบบระบบ ซึ่งจะช่วยให้ทราบถึงรายละเอียดที่ระบบต้องการ สามารถพัฒนาระบบได้อย่างมีประสิทธิภาพ ในบทนี้จะกล่าวถึงการวิเคราะห์และออกแบบระบบของ Ant Colony Optimization (ACO) เพื่อใช้ในการแก้ปัญหา Traveling Salesman Problem (TSP) โดยอาศัยพื้นฐานจากข้อมูลเดิมที่มีอยู่แล้ว มาพัฒนาเพื่อให้เหมาะสมต่อการพัฒนาในภาษา Java ในส่วนของการออกแบบระบบจะเป็นการอธิบายโดยใช้แบบจำลอง UML ซึ่งเหมาะกับการพัฒนาระบบเชิงวัตถุ (Object-Oriented) นอกเหนือจาก UML แล้วใน Algorithm บางส่วนที่มีความซับซ้อน จะมีการใช้ Flow-Chart อธิบายการทำงาน

### 3.1 การวิเคราะห์ระบบ

#### ความต้องการของระบบ

##### Distance

ปัญหาของ Traveling Salesman Problem - TSP หรือปัญหาการเดินทางของพนักงานขายคือ การที่พนักงานขายต้องเดินทางไปยังเมืองต่างๆ ให้ครบทุกเมือง โดยมีข้อกำหนดว่าต้องไม่เดินทางผ่านเมืองที่เคยผ่านมาแล้ว และในท้ายที่สุดนั้นจะกลับมายังเมืองที่เริ่มต้นเดินทาง ในการเก็บข้อมูลของระยะทางจะเก็บข้อมูลของระยะทางระหว่างเมืองหนึ่งๆ ไปยังทุกเมืองเช่น กำหนดให้มีเมืองที่ต้องเดินทาง 3 เมืองคือ  $i$ ,  $j$  และ  $k$  ระยะทางระหว่างเมืองที่ต้องเก็บมีดังนี้

$i$	->	$j$
$i$	->	$k$
$j$	->	$i$
$j$	->	$k$
$k$	->	$i$
$k$	->	$j$

รูปที่ 3.1 ตัวอย่างความสัมพันธ์ระหว่างโหนด

ซึ่งเมื่อพิจารณาแล้วจะพบว่าระยะทางระหว่าง  $i \rightarrow j$  มีค่าเท่ากับ  $j \rightarrow i$  หรือ  $j \rightarrow k$  มีค่าเท่ากับ  $k \rightarrow j$  ดังนั้นระยะทางที่ควรเก็บจึงเหลือเพียง  $i \rightarrow j$ ,  $i \rightarrow k$ ,  $j \rightarrow k$  ซึ่งจำนวนระยะทางระหว่างเมืองที่เก็บจะแปลงได้เป็นดังสมการ  $n(n-1)/2$  โดยที่  $n$  คือจำนวนเมืองทั้งหมดของปัญหา

### Nearest-Neighbor List

นอกจากระยะทางระหว่างเมืองแล้ว จำเป็นที่จะต้องเก็บรายการของเมืองที่มีระยะทางที่ใกล้กับเมืองนั้นๆ เพื่อใช้ในการตัดสินใจเดินทางไปยังเมืองถัดไป ระยะทางที่เก็บเป็นระยะทางระหว่างเมืองทั้งหมด โดยเรียงลำดับจากเมืองที่ใกล้ที่สุดไปยังเมืองที่อยู่ไกลที่สุด

### Pheromone Trails

เป็นค่าที่ใช้สำหรับประกอบการตัดสินใจเดินทาง (ของมด) และยังเป็นค่าที่มีการเพิ่มขึ้นหากมีการเดินทางผ่านเมืองเป็นจำนวนมากหรือลดลงหากไม่มีการเดินทางผ่านเมืองในแต่ละรอบ การเก็บค่าฟีโรโมนใช้วิธีการเก็บเช่นเดียวกับการเก็บค่าระยะทางระหว่างเมืองคือ ค่าฟีโรโมนของระยะทางระหว่างเมือง  $i$  กับเมือง  $j$  จะมีค่าเท่ากับค่าฟีโรโมนของระยะทางระหว่างเมือง  $j$  กับเมือง  $i$

### Heuristic Information

เป็นข้อมูลที่ได้จากการคำนวณของระบบเพื่อใช้ในการสุ่มเลือกหาความน่าจะเป็นของเมืองที่จะต้องเดินทางเป็นเมืองถัดไป เช่นเดียวกับค่า Distance และ Pheromone ข้อมูล Heuristic Information ระหว่างเมือง  $i \rightarrow j$  จะมีค่าเท่ากับ  $j \rightarrow i$

### Ant

ทำหน้าที่เดินทางไปยังเมืองทางต่างๆ กล่าวคือทำหน้าที่แทนพนักงานขายในปัญหา Traveling Salesman Problem นั่นเอง โดยหน้าที่ของมันจะเป็นการเดินทางไปยังเมืองต่างๆ เก็บค่าระยะทางระหว่างเมือง จนท้ายที่สุดจะกลับมาถึงเมืองเริ่มต้น จากนั้นจะคำนวณระยะทางทั้งหมดที่ใช้เดินทางเพื่อนำไปช่วยในการปรับปรุงค่าฟีโรโมนของระยะทางระหว่างเมืองต่อไป

## 3.2 การออกแบบระบบ

### 3.2.1 Data Structures

Ant Colony Optimization Algorithm ได้ถูกคิดค้นและพัฒนาต่อยอดไปมากมาย แต่ก็ยังอาศัยพื้นฐานเดิมที่ได้ถูกออกแบบในลักษณะของ Data Structure ตามรูปด้านล่าง

```

% Problem data
integer    dist [n] [n]
integer    nn_list [n] [nn]
double     pheromone [n] [n]
double     choice_info [n] [n]

% Ants
structure single_ant
begin
    double    tour_length
    integer    tour [n+1]
    integer    visited [n]
end
single_ant ant [n]
  
```

รูปที่ 3.2 โครงสร้างข้อมูลของระบบ

โดย

$n$  คือ จำนวนของเมืองหรือโหนด (city or node) สำหรับปัญหา

ข้อมูลทั่วไปสำหรับปัญหามีดังนี้

$dist [n] [n]$  คือ ความยาวระหว่างโหนด

$nn\_list [n] [nn]$  คือ list ของโหนดที่อยู่ใกล้โหนดนั้นๆ ที่สุด

$pheromone [n] [n]$  คือ ค่าฟีโรโมน

$choice\_info [n] [n]$  คือ ค่ารวมของ pheromone และ heuristic information

ข้อมูลของมด (สำหรับแก้ไขปัญหา) มีดังนี้

$single\_ant$  เป็น type ที่ถูกสร้างขึ้นใหม่เพื่อใช้แก้ไขปัญหา โดยมีรายละเอียดดังนี้

$tour\_length$  คือ ความยาวรวมของเส้นทางของเมืองหรือโหนดทั้งหมดที่ได้เดินทางผ่าน

$tour [n+1]$  คือ ลำดับของเมืองหรือโหนดทั้งหมดที่ได้เดินทางผ่าน

$visited [n]$  คือ เมืองหรือโหนดทั้งหมดที่ได้เดินทางผ่าน

$ant [n]$  คือ มดที่ถูกสร้างขึ้นมาเพื่อแก้ไขปัญหา TSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 Pseudo code

คือ รหัสจำลองที่ใช้เป็นตัวแทนหรือเพื่ออธิบายการทำงานของอัลกอริทึม โดยมีด้อยคำหรือประโยค คำสั่งที่เขียนอยู่ในรูปแบบของภาษาอังกฤษที่ไม่ขึ้น กับภาษาคอมพิวเตอร์ใดภาษาหนึ่ง สำหรับ Ant Colony Optimization Algorithm ที่ใช้สำหรับแก้ปัญหา Traveling Salesman Problem มีขั้นตอนการทำงานซึ่งอธิบาย ด้วย Pseudo code ดังนี้

#### 3.2.2.1 ซูโดโค้ดโดยรวมของระบบ

```

procedure ACOforTSP
  InitializeData
  while (not terminate) do
    ConstructSolutions
    LocalSearch (Optional)
    UpdateStatistics
    UpdatePheromoneTrails
  end while
end procedure

```

รูปที่ 3.3 ซูโดโค้ดโดยรวมของระบบ

#### 3.2.2.2 ซูโดโค้ดของการสร้างวิธีการแก้ไขปัญหา

```

procedure ConstructSolutions
  for k = 1 to m do
    for i = 1 to n do
      ant[k].visited[i] <- false
    end for
  end for
  step <- 1
  for k = 1 to m do
    r <- random{1, ..., n}
    ant[k].tour[step] <- r
    ant[k].visited[r] <- true
  end for
  while (step < n) do
    step <- step + 1
    for k = 1 to m do
      NeighborListASDescisionRule (k, step)
    end for
  end while
  for k = 1 to m do
    ant[k].tour[n + 1] <- ant[k].tour[1]
    ant[k].tour_length <- ComputeTourLength(k)
  end for
end procedure

```

รูปที่ 3.4 ซูโดโค้ดของการสร้างวิธีการแก้ไขปัญหา

ในชุดโคโด้นี้จะแสดงการทำงานเป็นลำดับขั้นโดยเริ่มจากการลบข้อมูลของการเดินทางของมดทุกตัวและทำการสุ่มค่าเพื่อกำหนดโหนดเริ่มต้นในการเดินทาง จากนั้นจะทำการตัดสินใจเลือกเส้นทางเพื่อให้มดเดินทางไปยังโหนดถัดไปโดยใช้ฟังก์ชัน ASDecisionRule สุดท้ายมดแต่ละตัวจะทำการคำนวณหาระยะทางทั้งหมดที่มดใช้เดินทางจากโหนดแรกไปยังโหนดสุดท้าย

### 3.2.2.3 ชุดโคโด้ของการตัดสินใจในการเดินทางไปยังโหนดถัดไป

```

Procedure NeighborListASDecisionRule(k, i)
  input k
  input i
  c <- ant[k].tour[i - 1]
  sum_probabilities <- 0.0
  for j = 1 to m do
    if ant[k].visited[nn_list[c][j]] then
      selection_probability[j] <- 0.0
    else
      selection_probability[j] <- choice_info[c][nn_list[c][j]]
      sum_probabilities = sum_probabilities + selection_probability[j]
    end if
  end for
  if (sum_probabilities = 0.0) then
    ChooseBestNext(k,i)
  else
    r <- random[0, sum_probabilities]
    j <- 1
    p <- selection_probability[j]
    while (p < r) do
      j <- j + 1
      p <- p + selection_probability[j]
    end while
    ant[k].tour[i] <- nn_list[c][j]
    ant[k].visited[nn_list[c][j]] <- true
  end if
end procedure

```

รูปที่ 3.5 ชุดโคโด้ของการตัดสินใจในการเดินทางไปยังโหนดถัดไป

การตัดสินใจในการเลือกเดินทางไปยังโหนดถัดไปนั้น มดจะอาศัยข้อมูลการเดินทางของตัวเองและใช้ข้อมูลของ Heuristic Information ( $choice\_info[n][n]$ ) ในการตัดสินใจ โดยหากโหนดใดได้เดินทางผ่านมาแล้ว จะไม่เดินทางผ่านอีก ค่าความน่าจะเป็นที่ได้สำหรับโหนดนั้นจะเป็น 0 ส่วนโหนดใดที่ยังไม่ได้เดินทางผ่าน จะถูกกำหนดค่าความน่าจะเป็นของการเลือกเส้นทางโดยใช้ค่า  $choice\_info$  ของแต่ละโหนดนั้น และทำการเพิ่มค่าของความน่าจะเป็นในการเดินทางรวม ( $sum\_probabilities$ ) หากค่า  $sum\_probabilities$  มีค่าเท่ากับ 0 แสดงว่าทุกโหนดได้ทุกมดเดินทางผ่านมาแล้ว การเลือกเส้นทางจะเปลี่ยนไปใช้ฟังก์ชัน ChooseBestNext แต่ถ้าค่า  $sum\_probabilities$  ไม่เท่ากับ 0 จะทำการสุ่มข้อมูลจากโหนดที่มีค่า  $choice\_info$  ไม่เท่ากับ 0 โดยอาศัยค่า  $sum\_probabilities$  เป็นตัวช่วยในการสุ่มข้อมูล เมื่อได้ข้อมูลที่ถูกสุ่มออกมา มดจะทำการกำหนดค่าโหนดที่ต้องเดินทางโหนดถัดไปตามค่าที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2.4 ซูโดโค้ดของการเลือกโหนดที่ดีที่สุดสำหรับการเดินทาง

```

Procedure ChooseBestNext(k, i)
  input k
  input i
  v <- 0.00
  c <- ant[k].tour[i - 1]
  for j = 1 to n do
    if not ant[k].visited[j] then
      if choice_info[c][j] > v then
        nc <- j
        v <- choice_info[c][j]
      end if
    end if
  end for
  ant[k].tour[i] <- nc
  ant[k].visited[nc] <- ture
end procedure

```

รูปที่ 3.6 ซูโดโค้ดของการเลือกโหนดที่ดีที่สุดสำหรับการเดินทาง

การเลือกโหนดที่ดีที่สุดนั้น จะทำการวนลูปตามจำนวนโหนดทุกโหนดที่อยู่ติดกับโหนดที่มอดอาศัยอยู่ เพื่อทำการหาโหนดที่มีค่า Heuristic Information (choice\_info) มากที่สุด และเดินทางไปยังโหนดนั้นต่อไป

### 3.2.2.5 ซูโดโค้ดของการปรับปรุงค่าฟีโรโมน

แบ่งการทำงานออกเป็น 2 ส่วนคือ

3.2.2.5.1 การลดค่าฟีโรโมน(การระเหยของฟีโรโมน)

3.2.2.5.1 การเพิ่มค่าฟีโรโมน

```

procedure ASPheromoneUpdate
  Evaporate
  for k = 1 to m do
    DepositPheromone(k)
  end for
  ComputeChoiceInformation
end procedure

```

รูปที่ 3.7 ซูโดโค้ดของการปรับปรุงค่าฟีโรโมน

### 3.2.2.5.1 ชุดโค้ดของการลดค่าฟีโรโมน

```

procedure Evaporate
  for i = 1 to n do
    for j = i to n do
      pheromone[i][j] <- (1 - p) * pheromone[i][j]
      pheromone[j][i] <- pheromone[i][j]
    end for
  end for
end procedure

```

รูปที่ 3.8 ชุดโค้ดของการลดค่าฟีโรโมน

ระบบจะวนลูปเพื่อทำการคำนวณเพื่อหาค่าที่ลดลงของทุกฟีโรโมนและกำหนดค่าที่ได้ให้กับตัวแปร ที่มีการกำหนดค่าสองครั้งเนื่องจาก  $\text{pheromone}[j][i]$  นั้นถือว่าเป็นข้อมูลเดียวกันกับ  $\text{pheromone}[i][j]$  นั่นเอง

### 3.2.2.5.2 ชุดโค้ดของการเพิ่มค่าฟีโรโมน

```

procedure DepositPheromone(k)
  input k
  t <- 1 / ant[k].tour_length
  for i = 1 to n do
    j <- ant[k].tour[i]
    l <- ant[k].tour[i + 1]
    pheromone[j][l] <- pheromone[j][l] + t
    pheromone[l][j] <- pheromone[j][l]
  end for
end procedure

```

รูปที่ 3.9 ชุดโค้ดของการเพิ่มค่าฟีโรโมน

เช่นเดียวกับการลดค่าฟีโรโมน ระบบจะวนลูปเพื่อทำการคำนวณเพื่อหาค่าที่เพิ่มของทุกฟีโรโมนโดยใช้ค่าระยะทางทั้งหมดที่มดเดินทางจากโหนดแรกไปยังโหนดสุดท้ายของมด  $k$  มาช่วยคำนวณ และกำหนดค่าที่ได้ให้กับตัวแปร ที่มีการกำหนดค่าสองครั้งเนื่องจาก  $\text{pheromone}[j][i]$  นั้นถือว่าเป็นข้อมูลเดียวกันกับ  $\text{pheromone}[i][j]$

## 3.2.2.5.3 ชุดโค้ดของการปรับปรุงค่าความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไป

```

procedure ComputeChoiceInformation
  for i = 1 to n do
    for j = i to n do
      choice_info[i][j] <- pheromone[i][j] * exp(1 / dis[i][j], B)
      choice_info[j][i] <- choice_info[i][j]
    end for
  end for
end procedure

```

รูปที่ 3.10 ชุดโค้ดของการปรับปรุงค่าความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไป

เป็นการคำนวณค่าความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไปของทุกโหนด โดยจะทำการวนลูปเพื่อคำนวณค่าระหว่างโหนด [i] และโหนด [j] โดยใช้ค่าของฟีโรโมนและระยะทางระหว่างโหนดมาช่วยคำนวณ จากนั้นนำค่าที่ได้มากำหนดค่าให้กับตัวแปร ที่มีการกำหนดค่าสองครั้งเนื่องจาก choice\_info[j][i] นั้นถือว่าเป็นข้อมูลเดียวกันกับ choice\_info[i][j] เหมือนกับการคำนวณค่าการเพิ่มและการลดของฟีโรโมน

## 3.2.2.5.4 ชุดโค้ดของการปรับปรุงค่าฟีโรโมนของ Ant Colony System

```

procedure ACSLocalPheromoneUpdate(k, i)
  input k
  input i
  h <- ant[k].tour[i - 1]
  j <- ant[k].tour[i]
  pheromone[h][j] <- (1 - E) * (pheromone[h][j] + Et0)
  pheromone[j][h] <- pheromone[h][j]
  choice_info[h][j] <- pheromone[h][j] * exp(1 / dis[h][j], B)
  choice_info[j][h] <- choice_info[h][j]
end procedure

```

รูปที่ 3.11 ชุดโค้ดของการปรับปรุงค่าฟีโรโมนของ Ant Colony System

การปรับปรุงค่าฟีโรโมนและความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไปของ Ant Colony System จะแตกต่างจาก Ant System และ Min-Max Ant System คือ ในแต่ละครั้งที่มุดเดินทางไปยังโหนดถัดไป ระบบจะทำการคำนวณค่าของฟีโรโมนและความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไปใหม่ หลังจากนั้นจะกำหนดค่าใหม่ทันทีโดยไม่ต้องรอให้มุดเดินทางจนครบทุกโหนดเหมือนอัลกอริทึมอื่น

เนื่องจากการพัฒนาระบบได้อาศัยภาษา Java ในการพัฒนา โดยภาษา Java นั้นเป็นภาษาเชิงวัตถุ เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น จึงได้มีการออกแบบใหม่จากการใช้ Data Structure มาเป็น class และอาศัยพื้นฐานของ UML Model ดังจะกล่าวในส่วนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 Class Responsibility Collaborator (CRC)

สร้างขึ้นเพื่อให้เห็นภาพของ Class ได้ชัดเจนยิ่งขึ้นว่ามีหน้าที่อย่างไรและทำหน้าที่ร่วมกับ Class ใดบ้าง ก่อนที่จะนำไปสร้างเป็น Class จริงของระบบ จาก Data Structure ข้างต้น สามารถสร้างใหม่ในรูปแบบของ CRC ดังนี้

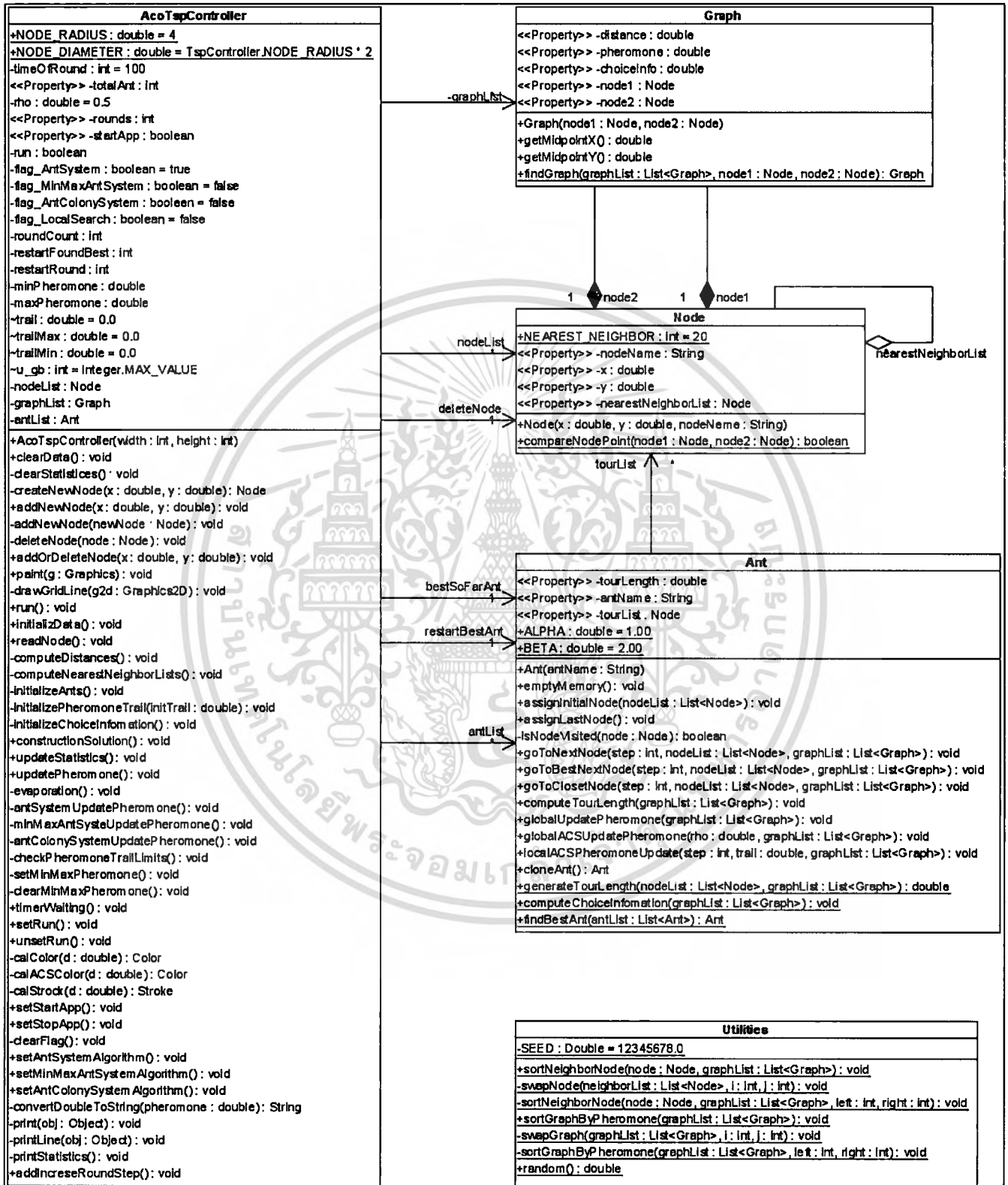
<b>Description:</b> เก็บข้อมูลที่เกี่ยวกับระหว่างโหนด		<b>Description:</b> เก็บข้อมูลของโหนดที่ถูกรัง	
<b>Attributes:</b>		<b>Attributes:</b>	
<b>Name</b>	<b>Description</b>	<b>Name</b>	<b>Description</b>
Start node	โหนดเริ่มต้น	Point X	ระขบนแกน X
End node	โหนดสิ้นสุด	Point Y	ระขบนแกน Y
Distance value	ระขทางระหว่างโหนด	Nearest-Neighbor list	รายการของโหนดที่อยู่ติดกัน
Pheromone value	ค่าฟีโรโมน		
Choice info value	ค่าความน่าจะเป็นสำหรับเลือกเส้นทาง		
<b>Description:</b> มดที่ใช้ในการทำงานของระบบ		<b>Description:</b> ความคุมการทำงานของระบบ	
<b>Attributes:</b>		<b>Attributes:</b>	
<b>Name</b>	<b>Description</b>	<b>Name</b>	<b>Description</b>
Tour list	รายการของโหนดที่เคยเดินทางผ่าน	Node list	โหนดทั้งหมดในระบบ
Tour length	ระขทางทั้งหมด	Graph list	กราฟที่เกิดขึ้นในระบบ
		Ant list	มดทั้งหมดในระบบ
		Parameter	ตัวแปรในระบบ
<b>Responsibilities:</b>		<b>Responsibilities:</b>	
<b>Name</b>	<b>Collaborator</b>	<b>Name</b>	<b>Collaborator</b>
Choose and go to next node	Node, Graph	Run algorithm	Node, Graph, Ant
Update pheromone	Graph	Draw graphic	
		Update statistics	

รูปที่ 3.12 Class Responsibility Collaborator (CRC) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 Class Diagram

เป็นแบบจำลองที่แสดงถึง Object ที่เกิดขึ้น พร้อมทั้งแสดงความสัมพันธ์ของแต่ละ Object



รูปที่ 3.13 Class diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รายละเอียดของ Class diagram

สำหรับ ACO application จะถูกแบ่งออกเป็น 2 package โดยมี package application สำหรับ class ที่เกี่ยวข้องกับส่วนแสดงผลทั้งหมด และ package acotsp สำหรับคลาสที่เกี่ยวข้องกับการทำงานของ Ant Colony Optimization ที่ใช้ในการแก้ปัญหาการเดินทางของพนักงานขาย (TSP) โดยจะกล่าวถึงรายละเอียดเฉพาะ - ของคลาสใน package acotsp เท่านั้น

### *Class AcoTspController*

เป็นคลาสหลักของส่วนประมวลผลข้อมูล ซึ่งคลาส AcoTspController นี้จะควบคุมการทำงานเบื้องหลังทั้งหมดเช่น การสั่งการให้ class อื่นทำงานหรือคำนวณค่า การส่งข้อมูลไปยังส่วนแสดงผล เป็นต้น

### *Class Ant*

เป็นคลาสของมดที่ถูกสร้างขึ้น (มดประดิษฐ์) เพื่อใช้การทำงานของ ACO Algorithm เช่นการเดินทางไปยังโหนดต่างๆ การปรับปรุงค่าฟีโรโมน สำหรับอัลกอริทึมที่ใช้ในการปรับปรุงค่าฟีโรโมนก็จะอยู่ในคลาสนี้

### *Class Node*

เป็นคลาสที่ใช้สำหรับเก็บข้อมูลของ node ที่ผู้ใช้งานสร้าง โดยจะเก็บตำแหน่งของ node (x, y) และรายการของโหนดที่อยู่ใกล้กับตัวเอง (Nearest-Neighbor)

### *Class Graph*

เป็นคลาสที่ใช้สำหรับเก็บค่าความสัมพันธ์ระหว่างโหนด 2 โหนดที่ติดกัน โดยจะเก็บค่าของฟีโรโมนระหว่าง node, ระยะทางระหว่าง node และค่าความน่าจะเป็นสำหรับการตัดสินใจเลือกเดินทางของมด

### *Class Utilities*

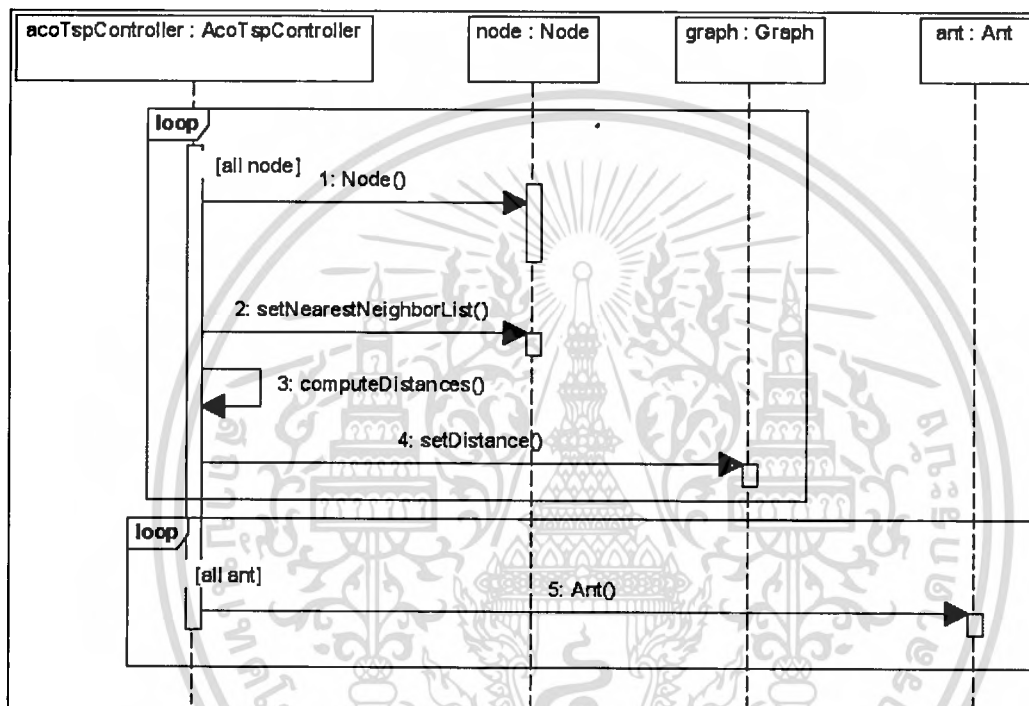
เป็นคลาสที่ใช้ช่วยประมวลผลการทำงานบางอย่างเช่น การเรียงลำดับข้อมูล

### 3.5 Sequence diagram

Sequence diagram เป็นการแสดงปฏิสัมพันธ์ (Interaction) ระหว่าง Object ตามลำดับของเหตุการณ์ที่เกิดขึ้น Object จะติดต่อกันผ่านทาง message โดยแสดงรายละเอียดได้ดังนี้

#### 3.5.1 Initialize data

ทำการกำหนดค่าเริ่มต้นต่างๆ ที่จำเป็นสำหรับการทำงานของระบบ

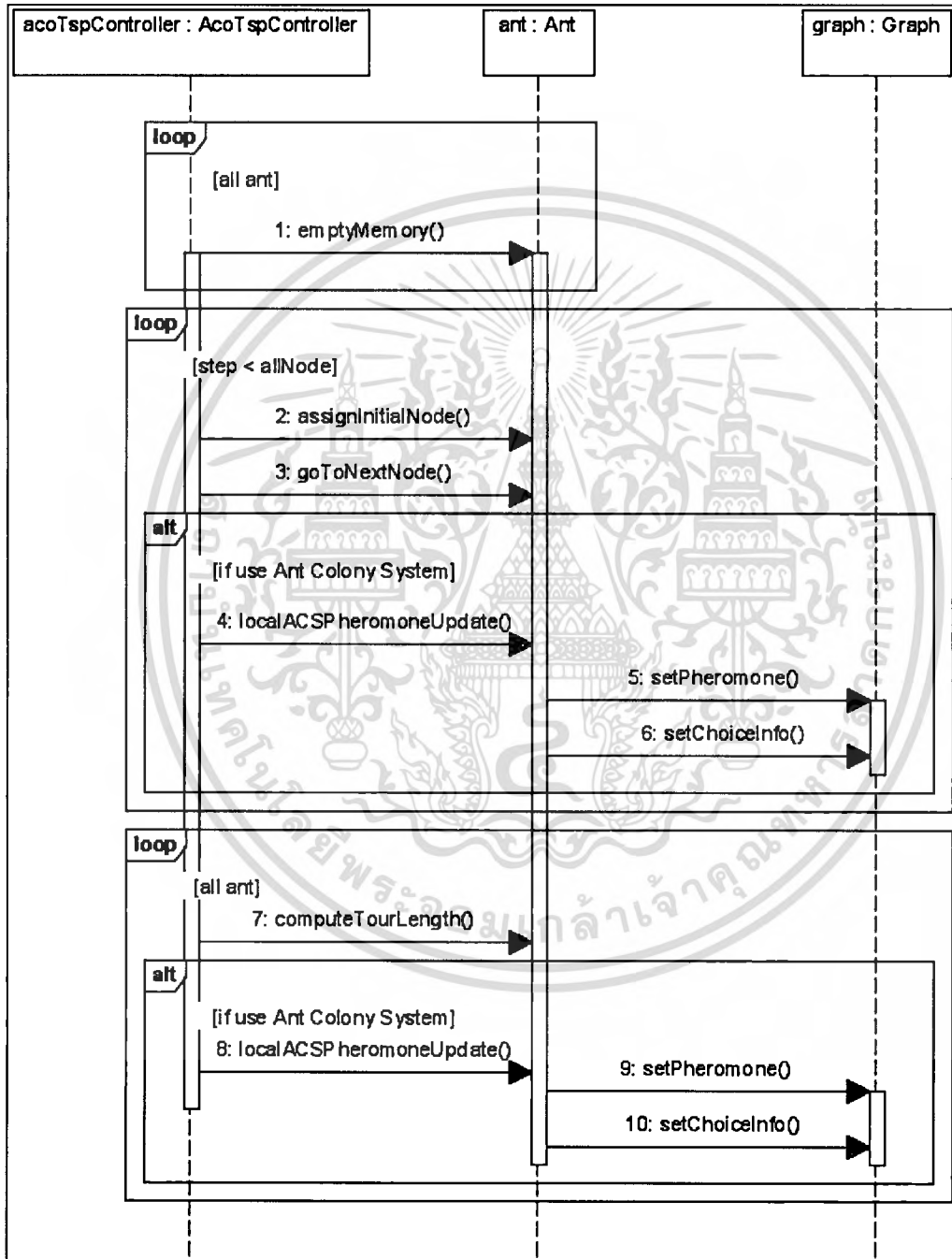


รูปที่ 3.14 Sequence ของการทำงาน Initialize data

1. คลาส AcoTspController ทำการวนลูปครั้งแรก เพื่อสร้าง โหนด (Node) ที่ถูกกำหนดทั้งหมด
2. หลังจากสร้างโหนดแล้วจะทำการกำหนดค่าของ Nearest-Neighbor หรือ โหนดที่ใกล้กับตัวเอง เรียงลำดับจากน้อยไปหามากจรครบทุกโหนด
3. คำนวณค่าระยะทางระหว่างโหนด
4. กำหนดค่าระยะทางระหว่างโหนดลงบนกราฟ (Graph object) โดยในเบื้องต้นกราฟจะเก็บค่าของ โหนดเริ่มต้น-โหนดสิ้นสุดและ ระยะทางระหว่างโหนด
5. คลาส AcoTspController วนลูปครั้งที่สองเพื่อสร้างมด (Ant) ตามจำนวนที่กำหนดเพื่อใช้ในการ แก้ไขปัญหา

### 3.5.2 Construction solution

สร้างวิธีการแก้ไขปัญหโดยการให้มดเดินทางไปยังโหนดทุกโหนดที่มีในระบบ หากใช้งานอัลกอริทึมของ Ant Colony System จะมีการปรับปรุงค่าฟีโรโมน (Pheromone) และความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไป (Choice Info) ในขั้นตอนนี้ด้วย



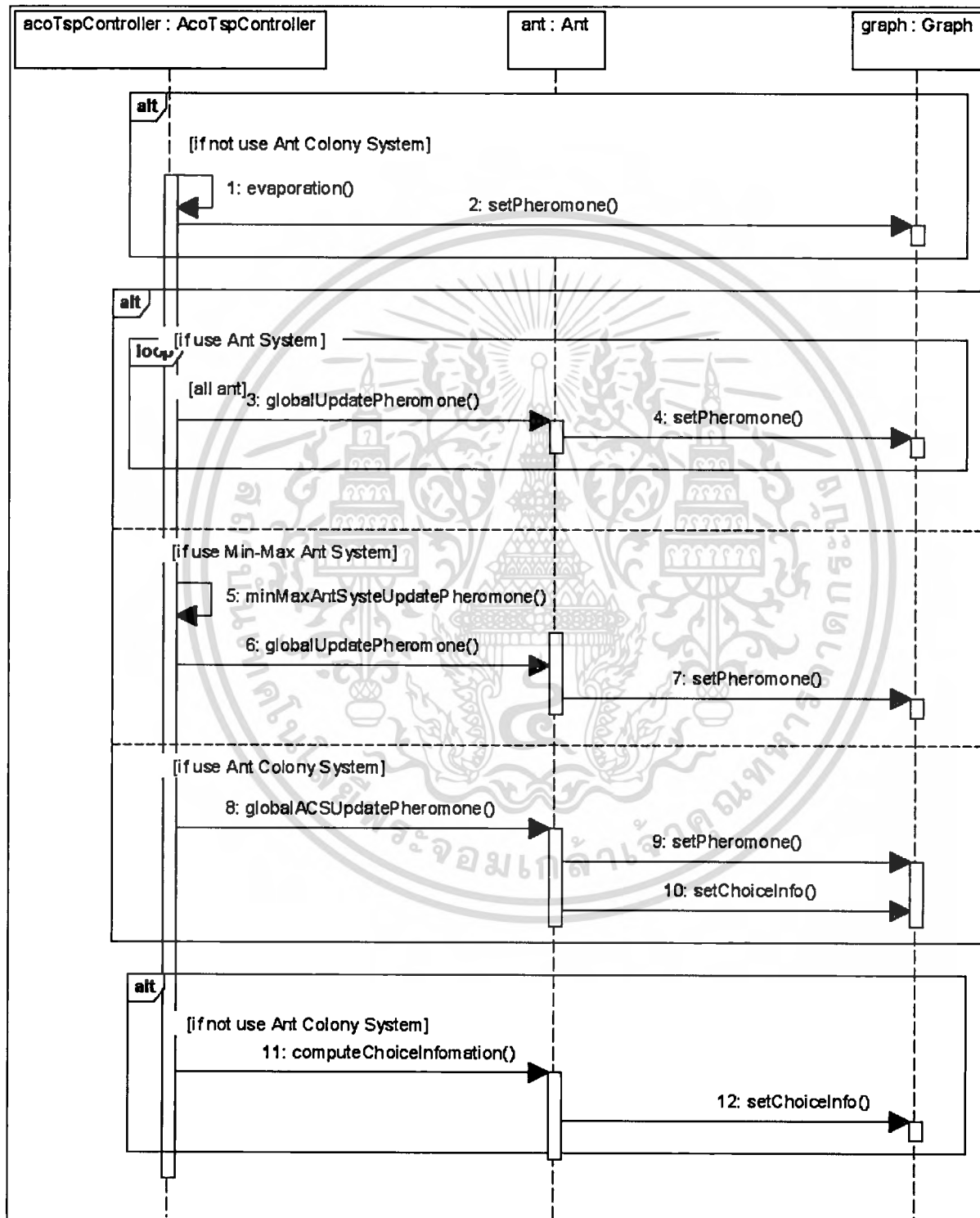
รูปที่ 3.15 Sequence ของการทำงาน Construction solution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คลาส AcoTspController ทำการวนลูปครั้งแรกเพื่อลบข้อมูลการเดินทางของมดทุกตัว
2. วนลูปครั้งที่สองตามจำนวนจุดโหนดทั้งหมด และกำหนดโหนดแรกที่มีคต้องเริ่มต้นการเดินทางโดยการสุ่มข้อมูล
3. ในการวนลูปครั้งที่สอง คลาส AcoTspController สั่งการให้มดเดินทางไปยังโหนดถัดไป
4. หากใช้อัลกอริทึมของ Ant Colony System จะมีเพิ่มการปรับปรุงค่าฟีโรโมนเข้าไปในช่วงของการวนลูปครั้งที่สองด้วย
5. กำหนดค่าของฟีโรโมนที่ได้ปรับปรุงแล้วให้กับกราฟของระยะทางระหว่างโหนด
6. กำหนดค่าของความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไปให้กับกราฟของระยะทางระหว่างโหนด
7. วนลูปครั้งที่สามตามจำนวนมดทั้งหมดเพื่อให้มดทำการคำนวณระยะทางที่ได้เดินทางผ่านโหนดแรกไปจนถึงโหนดสุดท้าย
8. หากใช้อัลกอริทึมของ Ant Colony System จะมีเพิ่มการปรับปรุงค่าฟีโรโมนเข้าไปในช่วงของการวนลูปครั้งที่สองด้วย
9. กำหนดค่าของฟีโรโมนที่ได้ปรับปรุงแล้วให้กับกราฟของระยะทางระหว่างโหนด
10. กำหนดค่าของความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไปให้กับกราฟของระยะทางระหว่างโหนด

### 3.5.3 Update pheromone

ปรับปรุงค่าฟีโรโมน (Pheromone) และค่าความน่าจะเป็นในการเลือกเส้นทางไปยังโหนดถัดไป (Choice Info) และกำหนดค่าที่ปรับปรุงให้กับกราฟของโหนดสองโหนด



รูปที่ 3.16 Sequence ของการทำงาน Update pheromone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ในกรณีที่ไม่ได้ใช้อัลกอริทึมของ Ant Colony System จะมีการลดค่าของฟีโรโมน (การระเหยของฟีโรโมน) ก่อนการปรับปรุงค่าฟีโรโมน
2. กำหนดค่าของฟีโรโมนที่ถูกลดลงให้กับกราฟ
3. หากใช้อัลกอริทึมของ Ant System วนลูปตามจำนวนมดทั้งหมด ทุกตัวจะทำการปรับปรุงค่าฟีโรโมนโดยรวมทั้งหมดในระบบ
4. กำหนดค่าฟีโรโมนที่ถูกปรับปรุงโดย Ant System ให้กับกราฟ
5. หากใช้อัลกอริทึมของ Min-Max Ant System จะทำการเลือกหามดที่เหมาะสมเพียงตัวเดียวเพื่อทำการปรับปรุงค่าฟีโรโมน
6. มดที่ถูกเลือกปรับปรุงค่าฟีโรโมนโดยรวมทั้งหมดในระบบ
7. กำหนดค่าฟีโรโมนที่ถูกปรับปรุงโดย Min-Max Ant System ให้กับกราฟ
8. หากใช้อัลกอริทึมของ Ant Colony System มดที่ใช้ระยะทางในการเดินทางสั้นจะทำการปรับปรุงค่าฟีโรโมนโดยรวมทั้งหมดในระบบ
9. กำหนดค่าฟีโรโมนที่ถูกปรับปรุงโดย Ant Colony System ให้กับกราฟ
10. กำหนดค่าของความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไปให้กับกราฟ
11. หากไม่ได้ใช้อัลกอริทึมของ Ant Colony System ระบบจะทำการคำนวณค่าความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไป
12. กำหนดค่าของความน่าจะเป็นในการเลือกเดินทางไปยังโหนดถัดไปให้กับกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# การทำงานของระบบ และการเปรียบเทียบอัลกอริทึม

การพัฒนา ระบบ Ant Colony Optimization for Traveling Salesman Problem ได้พัฒนาโดยใช้ภาษาจาวา (Java) เนื่องจากเป็นภาษาแบบเปิดไม่ต้องเสียต้นทุนในการซื้อลิขสิทธิ์เพื่อนำมาใช้พัฒนา ทั้งยังได้รับความนิยมค่อนข้างมาก นอกจากนั้นแล้วยังเป็นภาษาเชิงวัตถุ (Object-Oriented) ที่ให้ความสำคัญกับ วัตถุ ซึ่งสามารถนำมาประกอบกันและนำมาทำงานรวมกันได้ โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผลและส่งข่าวสารที่ได้ไปให้ วัตถุ อื่นๆที่เกี่ยวข้องเพื่อให้ทำงานต่อไป ทำให้การพัฒนาสามารถแบ่งการทำงานต่างๆ ของระบบออกมาได้อย่างชัดเจน

โดยรายละเอียดและสภาพแวดล้อมของพัฒนาและการทำงานของระบบมีดังนี้

### ฮาร์ดแวร์

- CPU Intel Core2 Duo 2.0 GHz
- Hard Disk 160 GB
- RAM 2 GB

### ซอฟต์แวร์

- Microsoft Windows 7 64-bit
- Netbeans IDE เวอร์ชัน 6.9.1
- Java SDK เวอร์ชัน 1.6

## 4.1 การกำหนดตัวแปรที่เกี่ยวข้อง

สำหรับการพัฒนาระบบ จะมีการกำหนดตัวแปร (Parameter) ที่เกี่ยวข้องกับการทำงานของระบบ โดยแบ่งเป็น 2 ประเภทคือ 1.) ตัวแปรคงที่ของระบบ 2.) ตัวแปรที่ถูกกำหนดขึ้นเพื่อใช้งาน

### 4.1.1 ตัวแปรคงที่ของระบบ

เพื่อให้การทำงานของระบบเกิดประสิทธิภาพมากยิ่งขึ้น จึงมีการกำหนดตัวแปรในการทำงานของอัลกอริทึมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\rho = 0.5$

เปลี่ยนตัวแปรที่ใช้สำหรับการคำนวณเพื่อลดค่าของ Pheromone ในแต่ละรอบของการทำงาน

$\alpha = 1.0$

เป็นค่าที่กำหนดความสำคัญของฟีโรโมน ใช้การคำนวณเพื่อเพิ่มค่าของ Pheromone ในระบบ

$\beta = 2.0$

เป็นค่าที่กำหนดความสำคัญของการตัดสินใจเลือก โหนดถัด ใช้ในการคำนวณเพื่อเพิ่มค่า Choice Info ในระบบ

#### 4.1.2 ตัวแปรที่ถูกกำหนดขึ้นเพื่อใช้งาน

เพื่อให้ระบบสามารถรองรับการทำงานหลายรูปแบบเช่น การเปลี่ยนอัลกอริทึม หรือการกำหนด โหนดของปัญหา จึงมีการกำหนดตัวที่สามารถปรับเปลี่ยนได้ดังนี้

*Algorithm*

เพื่อให้สามารถเลือกการทำงานของอัลกอริทึม ได้ มีให้เลือก 3 อัลกอริทึมคือ

- Ant System Algorithm
- Min-Max Ant System Algorithm
- Ant Colony System Algorithm

*Node*

เพื่อให้สามารถกำหนดจำนวนของ โหนดและพื้นที่ที่แสดง โหนด

*Number of ant*

เพื่อให้สามารถกำหนดจำนวนของมดที่ทำงานในระบบ

*Rounds*

เพื่อให้สามารถกำหนดจำนวนรอบของการทำงานในระบบ

## 4.2 ผลลัพธ์จากการทำงานของระบบ

สำหรับผลลัพธ์ที่ได้จากการเมื่อระบบเสร็จสิ้นแล้ว ส่วนแรกคือการแสดงผลทางหน้าจอการทำงาน ของระบบ จะแสดงจำนวน โหนดและกราฟที่เกิดขึ้นระหว่าง โหนด โดยกราฟที่แสดงนั้นจะเป็นกราฟที่มีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณฟีโรโมนในจำนวนมาก กล่าวคือเป็นเส้นทางที่มดเดินทางผ่านเป็นจำนวนมากนั่นเอง ซึ่งก็หมายความว่าเส้นทางนั้นเป็นเส้นทางที่สั้นและเหมาะสมสำหรับการเดินทาง

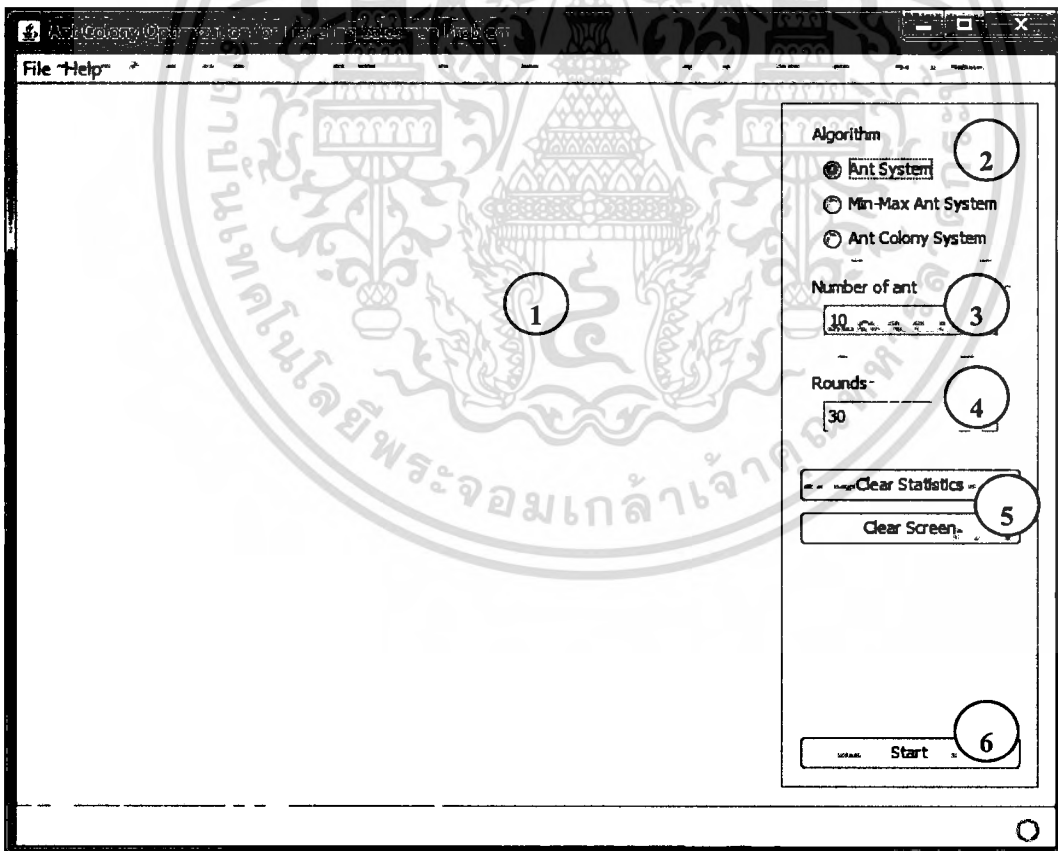
ผลลัพธ์ส่วนที่สองคือสถิติที่เกิดขึ้นจากการทำงาน โดยจะแสดงค่าของระยะทางที่สั้นที่สุดของการทำงานในรอบนั้นๆ รายละเอียดของผลลัพธ์ส่วนที่สองเป็นดังภาพที่ 4.1

```
[Algorithm]
Total ant:
Best round:
Distance:
Time use:
Best tour:
```

รูปที่ 4.1 ผลลัพธ์จากการทำงานของระบบ

### 4.3 หน้าจอสำหรับใช้งาน

เป็นส่วนที่ใช้แสดงผลการทำงานของระบบ สามารถเลือกตั้งค่าต่างๆ เพื่อการทำงานได้



รูปที่ 4.2 หน้าจอสำหรับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รายละเอียดของหน้าจอ

1. พื้นที่สำหรับสร้างโหนดและแสดงผลการทำงานของระบบ
2. อัลกอริทึมที่สามารถเลือกใช้ในการทำงานได้
3. กำหนดจำนวนมดที่ใช้ในการทำงาน
4. กำหนดจำนวนรอบของการทำงาน
5. ปุ่มสำหรับล้างข้อมูลที่ได้จากการทำงาน และปุ่มสำหรับล้างหน้าจอการทำงาน
6. ปุ่มสำหรับเริ่มต้นและสิ้นสุดการทำงาน

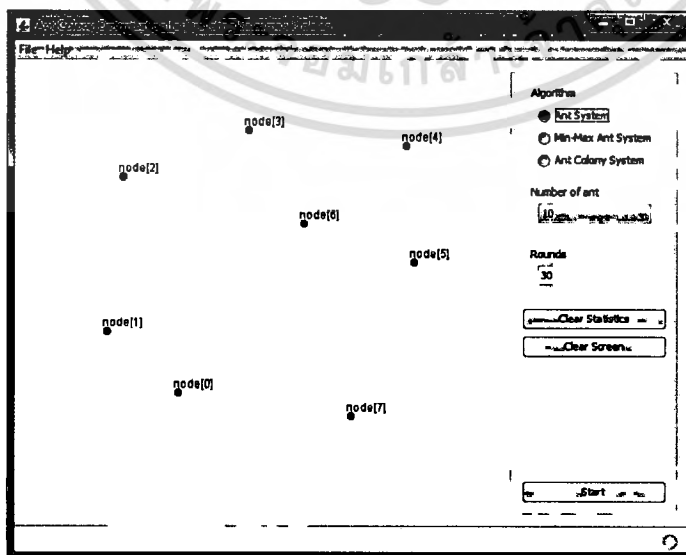
ในการใช้งานระบบนั้น จำเป็นต้องกำหนดโหนดสำหรับการทำงานของระบบก่อน โดยการคลิกที่พื้นที่ในส่วนที่ 1.) เพื่อสร้างโหนด ระบบจะทำการวาดโหนดตรงจุดที่มีการคลิก จากนั้นเลือกอัลกอริทึมที่ต้องการ กำหนดจำนวนมดที่ทำงานในระบบ กำหนดจำนวนรอบสำหรับการทำงาน เมื่อกดปุ่ม Start ระบบจะเริ่มทำงานตามจำนวนรอบที่กำหนด ในแต่ละรอบจะทำการวาดกราฟระยะทางระหว่างโหนด ซึ่งกราฟแสดงจำนวนของค่าฟีโรโมนที่มีอยู่ และสีของกราฟที่แตกต่างกันแสดงถึงระดับของฟีโรโมนโดย

- หากกราฟมีสีเข้มแสดงว่ามีปริมาณของสารฟีโรโมนอยู่จำนวนมาก
- หากกราฟมีสีอ่อนแสดงว่ามีปริมาณของสารฟีโรโมนอยู่จำนวนน้อย

### 4.4 ตัวอย่างการใช้งาน

แสดงตัวอย่างการใช้งานระบบ โดยคร่าวดังนี้

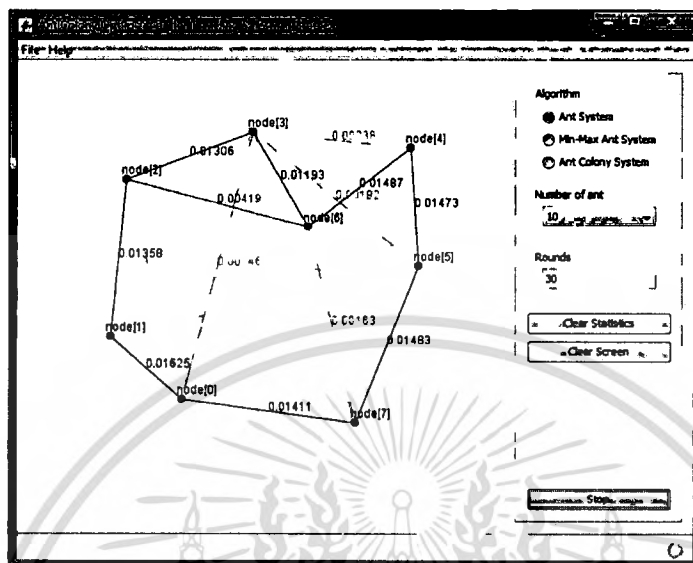
1. สร้างโหนดและกำหนดค่าเริ่มต้นของระบบตามรูปที่ 4.3



รูปที่ 4.3 แสดงการสร้างโหนดและกำหนดค่าเริ่มต้น

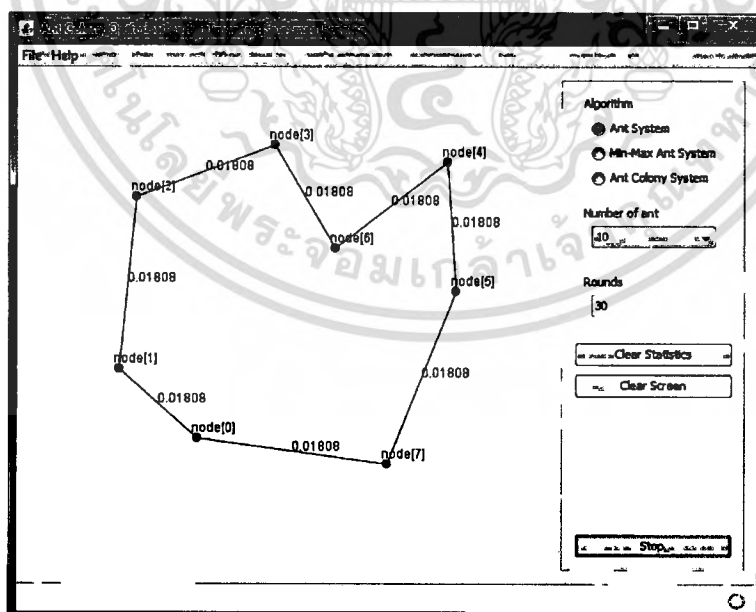
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กดปุ่ม Start เพื่อเริ่มทำงานจะเห็นได้ว่าระบบเริ่มวาดกราฟและกราฟแต่ละกราฟก็แสดงผลแตกต่างกันออกไปดังรูปที่ 4.4



รูปที่ 4.4 ระบบเริ่มทำงาน

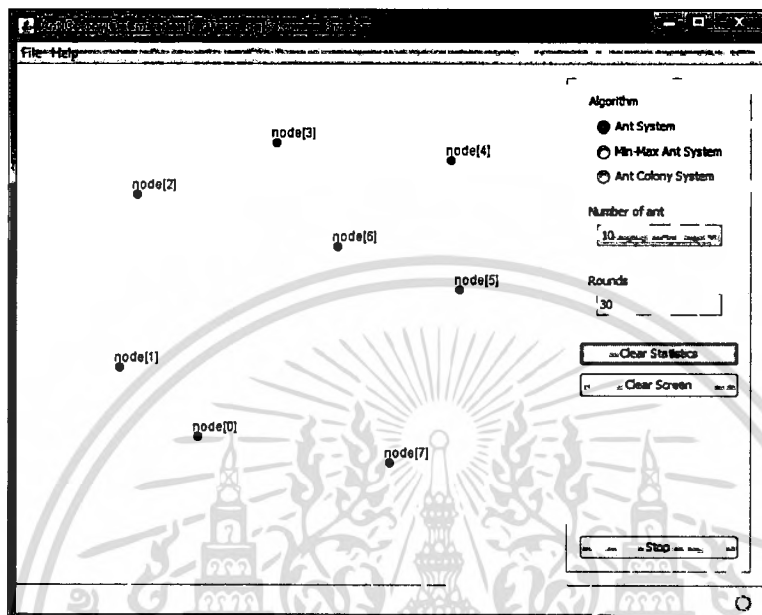
3. เมื่อระบบทำงานครบตามจำนวนรอบจะหยุดการทำงานและแสดงข้อมูลสุดท้ายของการทำงานตามรูปที่ 4.5



รูปที่ 4.5 ระบบหยุดทำงาน

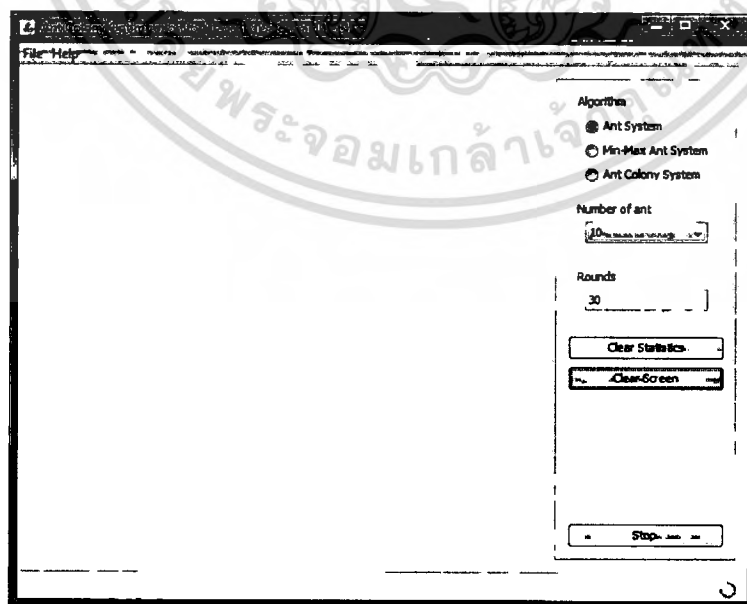
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หากต้องการใช้ข้อมูลโหนดเดิมที่สร้างไว้ ให้กดที่ปุ่ม Clear Statistics ระบบจะทำการล้างข้อมูลที่เกิดขึ้นจากการทำงานทั้งหมด แต่จะยังคงเหลือโหนดที่เคยสร้างไว้ดังเดิม



รูปที่ 4.6 แสดงหน้าจอเมื่อกดปุ่ม Clear Statistics

5. หากต้องการล้างข้อมูลทั้งหมดรวมทั้งโหนดที่ได้สร้างไว้ให้กดปุ่ม Clear Screen ระบบจะทำการล้างข้อมูลที่เกิดขึ้นจากการทำงานทั้งหมด รวมถึงโหนดที่ได้สร้างไว้ก็จะถูกลบออกไปด้วย



รูปที่ 4.7 แสดงหน้าจอเมื่อกดปุ่ม Clear Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.5 การเปรียบเทียบการทำงานของอัลกอริทึม

### 4.5.1 การออกแบบการทำลอง

การเปรียบเทียบการทำงานของอัลกอริทึมนั้นกระทำโดยนำผลลัพธ์ที่ได้จากการทำงานของอัลกอริทึมมาเปรียบเทียบข้อมูล การออกแบบการทำลองและขอบเขตของการเปรียบเทียบข้อมูลดังนี้

- อัลกอริทึมที่ใช้ในการเปรียบเทียบทั้งหมด 3 อัลกอริทึมดังนี้
  - Ant System Algorithm
  - Min-Max Ant System Algorithm
  - Ant Colony System Algorithm
- จำนวนของมดที่ใช้ในการทำงานแต่ละรอบ 3 แบบ ดังนี้
  - มด 10 ตัว
  - มด 30 ตัว
  - มด 50 ตัว
- รูปแบบการจัดวางโหนดในการเดินทาง ดังนี้
  - รูปแบบง่าย
  - รูปแบบปานกลาง
  - รูปแบบยาก

### 4.5.2 การทดลองเปรียบเทียบการทำงานของอัลกอริทึม

สำหรับขั้นตอนการทดลองเปรียบเทียบการทำงานเป็นดังนี้

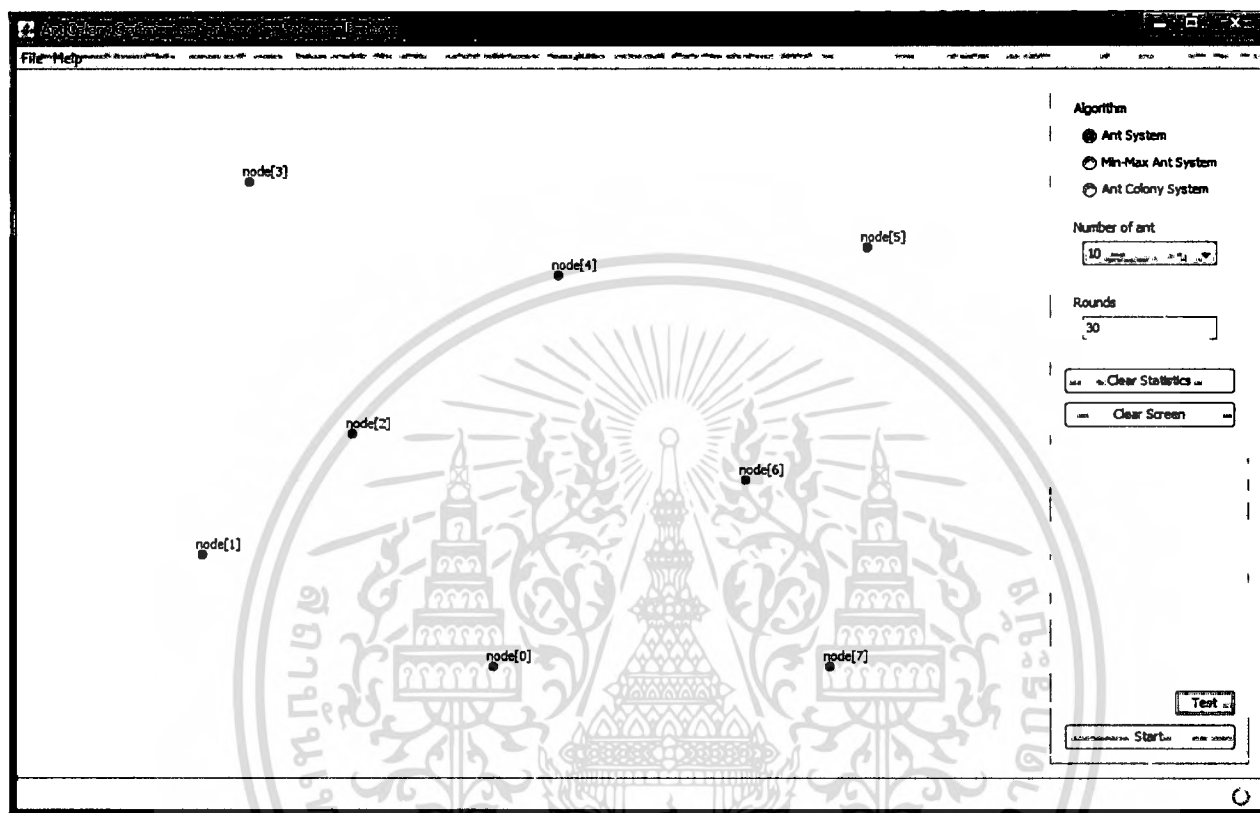
1. สร้างรูปแบบการจัดวางโหนดสำหรับการทดลองโดยเริ่มจากรูปแบบง่าย
2. กำหนดอัลกอริทึมและจำนวนมดสำหรับการทำงาน
3. ทดลองการทำงานของแต่ละอัลกอริทึม โดยเริ่มจากรูปแบบการจัดวางโหนดที่ง่ายที่สุด ในแต่ละอัลกอริทึมจะใช้มดสำหรับการทำงาน 3 จำนวนคือ 10 ตัว, 20 ตัว และ 30 ตัว ตามลำดับ
4. จัดเก็บผลลัพธ์ที่ได้จากการทดลอง
5. ทดลองการทำงานกับโหนดที่สร้างขึ้นในแบบปานกลางและแบบยากโดยใช้วิธีการทดลองตามข้อ 1 ถึงข้อ 4
6. สรุปและวิเคราะห์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.5.3 ผลการทดลอง

ผลการทดลองแบ่งตามรูปแบบการจัดวางโหนดดังนี้

#### 4.5.3.1 รูปแบบการจัดวางโหนดแบบที่ 1



รูปที่ 4.8 รูปแบบการจัดวางโหนดแบบง่าย

ตารางที่ 4.1 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวาง โหนดแบบที่ 1

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่คิดที่สุด	โหนดที่ผ่าน	การเดินทาง
10	4	1893.91606	0.003	42	[4], [5], [6], [7], [0], [1], [2], [3], [4]
30	3	1893.91606	0.006	92	[4], [5], [6], [7], [0], [1], [2], [3], [4]
50	2	1893.91606	0.009	102	[0], [7], [6], [5], [4], [3], [2], [1], [0]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลลัพธ์ของ Min-Max Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 1

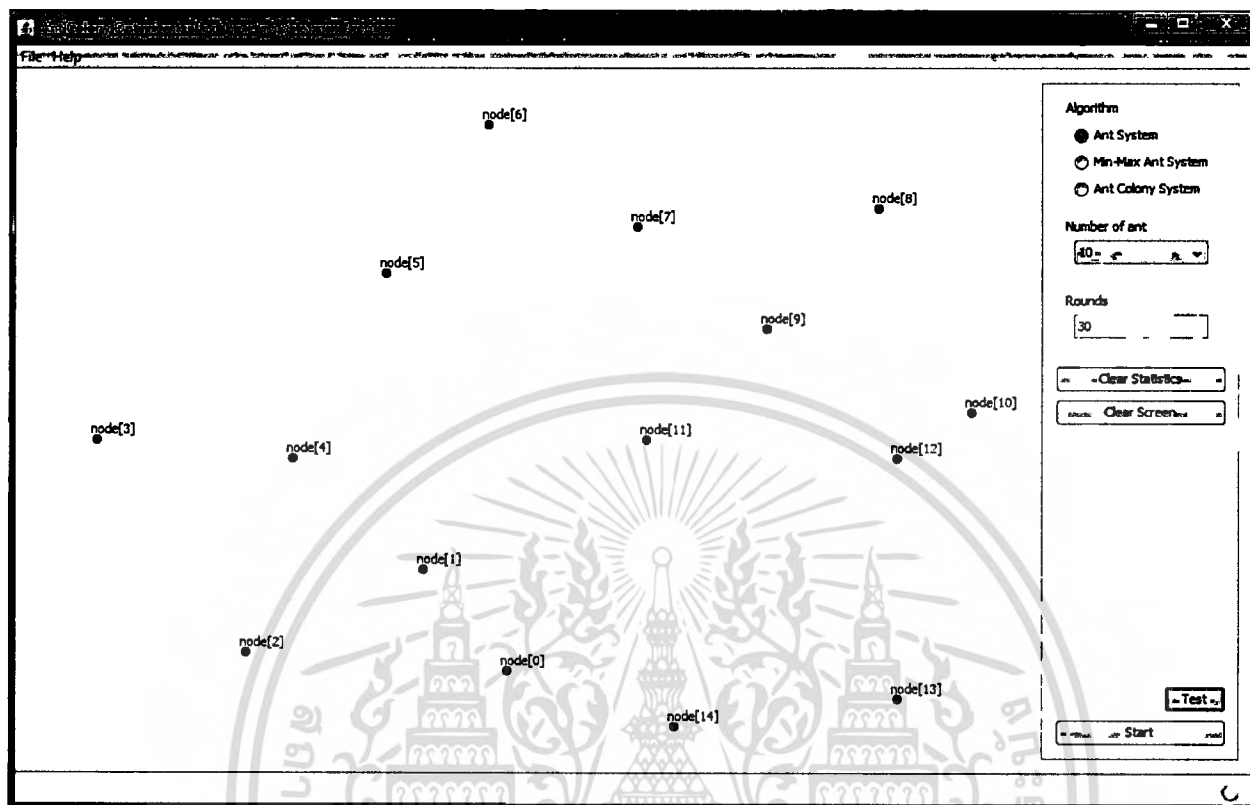
จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	2	1893.91606	0.002	22	[5], [4], [3], [2], [1], [0], [7], [6], [5]
30	3	1893.91606	0.007	92	[3], [2], [1], [0], [7], [6], [5], [4], [3]
50	2	1893.91606	0.008	102	[3], [2], [1], [0], [7], [6], [5], [4], [3]

ตารางที่ 4.3 ผลลัพธ์ของ Ant Colony System สำหรับการจัดวางโหนดแบบที่ 1

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	3	1893.91606	0.003	32	[0], [7], [6], [5], [4], [3], [2], [1], [0]
30	1	1893.91606	0.003	32	[4], [5], [6], [7], [0], [1], [2], [3], [4]
50	2	1893.91606	0.009	102	[4], [5], [6], [7], [0], [1], [2], [3], [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.5.3.2 รูปแบบการจัดวาง โหนดแบบที่ 2



รูปที่ 4.9 รูปแบบการจัดวาง โหนดแบบปานกลาง

ตารางที่ 4.4 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวาง โหนดแบบที่ 2

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่คิดที่สุด	โหนดที่ผ่าน	การเดินทาง
10	12	2453.671	0.075	122	[11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6], [7], [11]
30	13	2453.671	0.208	392	[5], [6], [7], [11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5]
50	10	2453.671	0.254	502	[6], [7], [11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ผลลัพธ์ของ Min-Max Ant System Algorithm สำหรับการจัดวางโหนดแบบที่ 2

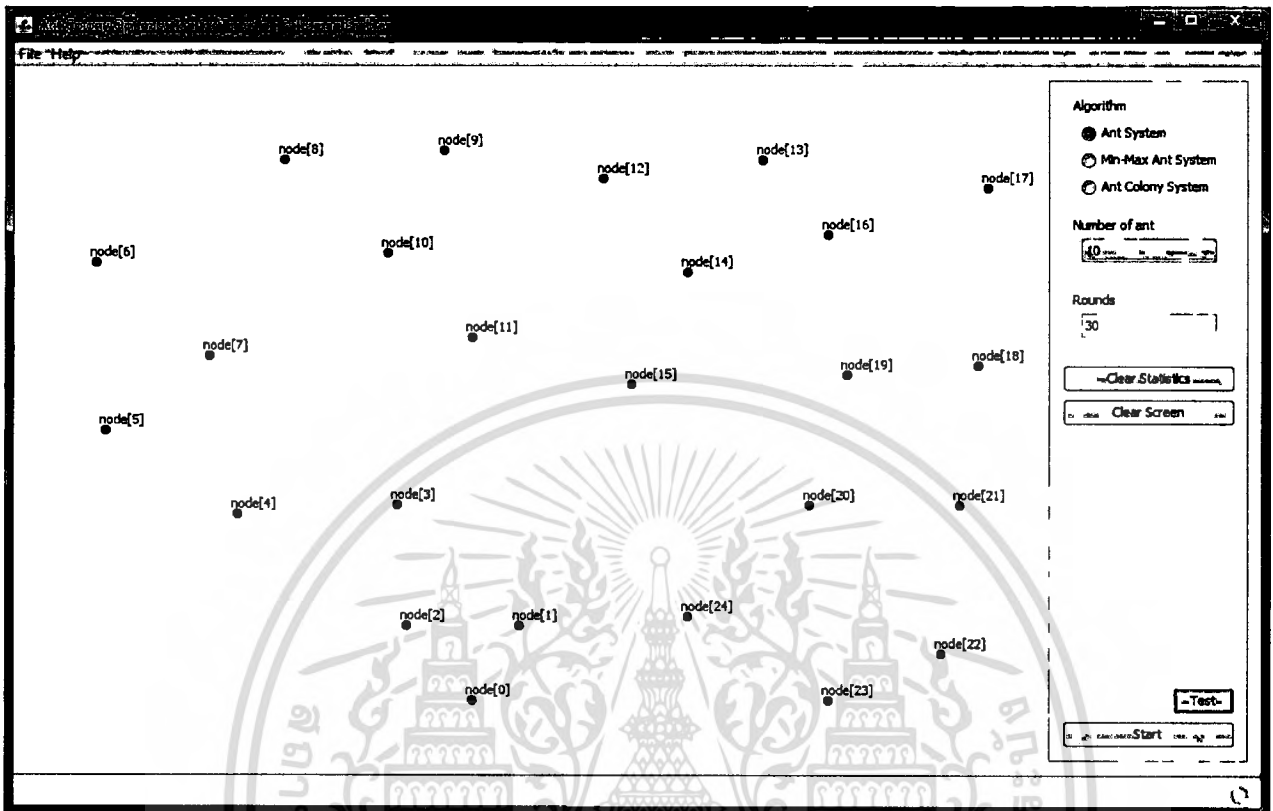
จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ดีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	31	2453.671	0.15	312	[11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6], [7], [11]
30	10	2453.671	0.15	302	[4], [5], [6], [7], [11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4]
50	5	2453.671	0.133	252	[5], [6], [7], [11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5]

ตารางที่ 4.6 ผลลัพธ์ของ Ant Colony System Algorithm สำหรับการจัดวางโหนดแบบที่ 2

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ดีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	26	2453.671	0.13	652	[11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6], [7], [11]
30	13	2453.671	0.0194	1352	[11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6], [7], [11]
50	20	2453.671	0.0496	502	[6], [7], [11], [9], [8], [10], [12], [13], [14], [0], [1], [2], [3], [4], [5], [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.5.3.3 รูปแบบการจัดวาง โหนดแบบที่ 3



รูปที่ 4.10 รูปแบบการจัดวาง โหนดแบบยาก

ตารางที่ 4.7 ผลลัพธ์ของ Ant System Algorithm สำหรับการจัดวาง โหนดแบบที่ 3

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	40	3035.478	0.957	402	[24], [1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24]
30	49	3035.478	3.389	1472	[24], [1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 (ต่อ)

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
50	7	3035.478	6.507	2852	[1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24], [1]

ตารางที่ 4.8 ผลลัพธ์ของ Min-Max Ant Colony System Algorithm สำหรับการจัดวางโหนดแบบที่ 3

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	238	3035.478	5.146	2382	[1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24], [1]
30	68	3224.199	4.26	2042	[15], [11], [3], [1], [0], [2], [4], [5], [7], [6], [8], [10], [9], [12], [14], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24], [15]
50	139	3035.478	14.266	6952	[1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24], [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 ผลลัพธ์ของ Ant Colony System Algorithm สำหรับการจัดวาง โหนดแบบที่ 3

จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน	การเดินทาง
10	260	3208.995	6.471	2602	[24], [1], [0], [2], [3], [4], [7], [5], [6], [8], [10], [11], [9], [12], [13], [14], [15], [20], [19], [16], [17], [18], [21], [22], [23], [24]
30	156	3035.478	11.581	4682	[1], [0], [2], [3], [4], [5], [7], [6], [8], [9], [10], [11], [15], [14], [12], [13], [16], [17], [18], [19], [20], [21], [22], [23], [24], [1]
50	281	3170.707	35.151	14052	[21], [18], [17], [13], [16], [19], [20], [15], [14], [12], [11], [10], [9], [8], [6], [7], [5], [4], [3], [2], [0], [1], [24], [23], [22], [21]

#### 4.5.4 วิเคราะห์ผลการทดลอง

จากผลการทดลองที่ได้ในหัวข้อ 4.5.3 เมื่อนำข้อมูลผลลัพธ์ทั้งหมดมารวมกัน โดยตัดข้อมูลในส่วนของการเดินทางออกไป โดยแยกตามการจัดวางโหนดเป็นดังนี้

##### 4.5.4.1 การจัดวางโหนดแบบที่ 1

ตารางที่ 4.10 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวาง โหนดแบบที่ 1

อัลกอริทึม	จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีที่สุด	โหนดที่ผ่าน
Ant System	10	4	1893.9161	0.003	42
	30	3	1893.9161	0.006	92
	50	2	1893.9161	0.009	102
Min-Max Ant System	10	2	1893.9161	0.002	22
	30	3	1893.9161	0.007	92
	50	2	1893.9161	0.008	102

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์อื่นใด  
 ไม่สามารถแก้ไขหรือเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 (ต่อ)

อัลกอริทึม	จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีดีที่สุด	โหนดที่ผ่าน
Ant Colony System	10	3	1893.9161	0.003	32
	30	1	1893.9161	0.003	32
	50	2	1893.9161	0.009	102

จากข้อมูลในตารางที่ 4.10 จะให้ได้ว่าผลลัพธ์ต่างๆ ที่ได้มีค่าใกล้เคียงกัน ไม่ว่าจะเป็น รอบที่ดีที่สุด, เวลาที่ดีที่สุด หรือจำนวนโหนดที่ผ่าน จึงพอสรุปได้ว่า หากรูปแบบของโหนดมีจำนวนน้อยและไม่มีควมซับซ้อนไม่มากนัก ผลลัพธ์ที่ได้จากทั้ง 3 อัลกอริทึมจะได้ค่าที่ใกล้เคียงหรือแทบไม่แตกต่างกันเลย

## 4.5.4.2 การจัดวางโหนดแบบที่ 2

ตารางที่ 4.11 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวางโหนดแบบที่ 2

อัลกอริทึม	จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ตีดีที่สุด	โหนดที่ผ่าน
Ant System	10	12	2453.671	0.075	122
	30	13	2453.671	0.208	392
	50	10	2453.671	0.254	502
Min-Max Ant System	10	31	2453.671	0.15	312
	30	10	2453.671	0.15	302
	50	5	2453.671	0.133	252
Ant Colony System	10	26	2453.671	0.13	262
	30	13	2453.671	0.0194	392
	50	20	2453.671	0.0496	1002

จากข้อมูลในตารางที่ 4.11 จะเห็นได้ว่าผลลัพธ์ที่ได้ในแตกต่างกันออกไปตามแต่ละอัลกอริทึม ใน Ant System Algorithm นั้นผลลัพธ์ของการทำงานจะแตกต่างกันตามจำนวนมดที่ใช้ในการแก้ปัญหา ส่วน Min-Max Ant System Algorithm นั้น ผลลัพธ์ของแต่ละแบบมีค่าใกล้เคียงกันและจำนวนรอบที่ดีที่สุดที่ได้จากการทำงานจะลดลงตามจำนวนมดที่เพิ่มขึ้น และผลลัพธ์ของ Ant Colony System Algorithm นั้นคล้ายกับผลลัพธ์ของ Ant System Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.5.4.3 การจัดวางโหนดแบบที่ 3

ตารางที่ 4.12 ผลลัพธ์รวมทั้ง 3 อัลกอริทึมของการจัดวางโหนดแบบที่ 3

อัลกอริทึม	จำนวนมด	รอบที่ดีที่สุด	ระยะทางที่ดีที่สุด	เวลาที่ดีที่สุด	โหนดที่ผ่าน
Ant System	10	40	3035.478	0.957	402
	30	49	3035.478	3.389	1472
	50	7	3035.478	6.507	2852
Min-Max Ant System	10	238	3035.478	5.146	2382
	30	68	3224.199	4.26	2042
	50	139	3035.478	14.266	6952
Ant Colony System	10	260	3208.995	6.471	2602
	30	156	3035.478	11.581	4682
	50	281	3170.707	35.151	14052

จากข้อมูลในตารางที่ 4.12 ซึ่งการจัดวางโหนดมีความซับซ้อนมากที่สุด ผลลัพธ์ของทั้ง 3 อัลกอริทึมแตกต่างกันอย่างชัดเจน จะเห็นได้ว่า Ant System Algorithm เมื่อทำงานโดยใช้จำนวนมดที่แตกต่างกันก็ยังสามารถหาระยะทางที่ดีที่สุดได้เหมือนกัน แต่เวลาที่ดีที่สุดและจำนวน โหนดที่ผ่านนั้นจะเพิ่มขึ้นตามจำนวนมดที่ใช้งาน สำหรับ Min-Max Ant System Algorithm และ Ant Colony System Algorithm นั้นแม้จะเพิ่มจำนวนมดขึ้นก็ไม่สามารถหาระยะทางที่ดีที่สุดได้ครบทุกรูปแบบการทดลอง ผลลัพธ์ของเวลาและจำนวนโหนดยังเพิ่มขึ้นเป็นจำนวนมากเมื่อเทียบกับ Ant System ซึ่งหากมีการเพิ่มจำนวนรอบในการทำงานที่มากขึ้น ทั้ง Min-Max Ant System Algorithm และ Ant Colony System Algorithm อาจจะหาระยะทางที่ดีที่สุดได้แต่เวลาที่ใช้ก็จะมากตามขึ้นไปด้วย

จากการวิเคราะห์ผลการทดลองทั้ง 3 แบบนั้น สำหรับรูปแบบโหนดที่มีจำนวนน้อย และไม่ซับซ้อนมากนัก การทำงานของทั้ง 3 อัลกอริทึมให้ผลลัพธ์ที่ได้ใกล้เคียงกัน แต่เมื่อรูปแบบของโหนดมีความซับซ้อนมากขึ้นมาในระดับหนึ่ง ผลลัพธ์ที่ได้ของ Min-Max Ant System Algorithm จะมาสามารถทำได้ดีกว่า อัลกอริทึมอื่นๆ และในแบบสุดท้ายที่โหนดมีจำนวนมากและมีความซับซ้อนมาก การทำงานของแต่ละอัลกอริทึมก็จะใช้เวลาเพิ่มขึ้นตาม หากวิเคราะห์ตามรูปแบบของการจัดวางโหนดแล้ว หากจำนวนของโหนดมีมากแต่การจัดวางโหนดไม่ค่อยมีความซับซ้อน การทำงานของ Min-Max Ant System Algorithm จะให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมอื่น แต่หากโหนดมีความซับซ้อนมากๆ การทำงานของ Ant System Algorithm จะให้ผลลัพธ์ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# บทสรุปและข้อเสนอแนะ

### 5.1 สรุปผลการพัฒนาระบบงาน

Ant Colony Optimization (ACO) เป็นการทำงานในแบบปัญญาประดิษฐ์ (Artificial Intelligence - A.I.) รูปแบบหนึ่ง ซึ่งอาศัยวิธีการทำงานเลียนแบบการเดินทางหาอาหารของมดตามธรรมชาติ โดยมดแต่ละตัวจะช่วยกันกระจายออกหาอาหาร ทำที่ที่สุดเส้นทางที่ได้จากรังไปยังแหล่งอาหารจะได้เส้นทางที่สั้นและมีประสิทธิภาพที่สุดในการเดินทาง

การพัฒนาระบบได้จำลองการทำงานของ ACO เพื่อใช้แก้ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem) เพื่อให้เห็นภาพการทำงานของ ACO ได้ชัดเจนยิ่งขึ้น ปัญหาดังกล่าวนั้น พนักงานขายต้องเดินทางไปยังเมืองต่างๆ ให้ครบทุกเมือง โดยมีข้อแม้ว่าต้องไม่เดินทางผ่านเมืองเดิมและทำที่ที่สุดต้องกลับมายังจุดเริ่มต้นของการเดินทาง

ระบบที่พัฒนาสามารถจำลองการสร้าง โหนดเปรียบเทียบได้กับเมืองต่างๆ ที่พนักงานขายต้องเดินทางผ่าน สามารถกำหนดจำนวนมดที่ใช้แก้ปัญหาและสามารถกำหนดจำนวนรอบของการทำงานได้ตามความเหมาะสม เมื่อเริ่มต้นการทำงานระบบจะทำการวาดกราฟระหว่างเมืองแต่ละเมือง และเมื่อมดเริ่มทำงานในแต่ละรอบ สีของกราฟจะเปลี่ยนไปตามระดับฟีโรโมนที่มีการเพิ่มขึ้นหรือลดลง โดยเมื่อระดับฟีโรโมนมีจำนวนมากสีของกราฟจะเข้มขึ้นในทางกลับกันเมื่อระดับฟีโรโมนมีจำนวนน้อยสีของกราฟจะอ่อนลง จากการเปลี่ยนแปลงของกราฟที่เกิดขึ้นในระบบทำให้สามารถทราบถึงของระดับการเปลี่ยนแปลงของฟีโรโมนได้ชัดเจน นอกจากนี้ยังเห็นรูปแบบการเดินทางระหว่าง โหนดที่เกิดขึ้นว่าควรที่จะเริ่มเดินทาง โหนดใดและสิ้นสุดที่โหนดใดได้อีกด้วย ทำที่ที่สุดเมื่อระบบทำงานครบตามจำนวนรอบที่กำหนดจะแสดงผลการทำงานของการทำงานซึ่งบอกรายละเอียดต่างๆ เช่น ระยะทางที่สั้นที่สุดที่สามารถหาได้ หรือระยะเวลาที่ใช้เมื่อพบเส้นทางที่สั้นที่สุด เป็นต้น

นอกจากนี้ยังมีการเปรียบเทียบการทำงานแบบคร่าวๆ ของแต่ละอัลกอริทึมเพื่อให้เห็นความแตกต่างของการทำงานตามสถานะแวดล้อมที่แตกต่างกัน ซึ่งผลลัพธ์ที่ได้อาจมีความคลาดเคลื่อนตามตัวแปร และข้อจำกัดที่ได้กำหนดในการทดลองแต่ละครั้ง

### 5.2 ข้อจำกัดในการเปรียบเทียบการทำงานของอัลกอริทึม

จากการเปรียบเทียบการทำงานของอัลกอริทึมในรูปแบบต่างๆ พบข้อจำกัดของการเปรียบเทียบการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. จำนวนของโหนดและความซับซ้อนของการจัดเรียงโหนด มีความหลากหลายไม่สามารถหา รูปแบบการจัดเรียงที่เหมาะสมได้ ทำให้ผลลัพธ์ของการทำงานอาจมีการเปลี่ยนแปลงในแต่ละรอบ ของการทำงานถึงแม้เวลาจะเป็นการทำงานภายใต้รูปแบบเดียวกัน
2. เมื่อการจัดวางโหนดมีความซับซ้อนมากๆ ไม่สามารถกำหนดจำนวนรอบที่ใช้ในการทำงานที่คงที่ ได้ เนื่องก้าเดินทางของมดในแต่ละรอบเป็นแบบสุ่มจากข้อมูลของกราฟระหว่างโหนด ในบางครั้ง อาจได้ผลลัพธ์ที่ดีที่สุดตั้งแต่การทำงานในรอบแรกๆ และในบางครั้งเมื่อครบกำหนดการทำงาน ตามรอบที่กำหนดแล้วก็อาจจะยังไม่ได้ผลลัพธ์ที่ดีที่สุดเลยก็ได้
3. ประสิทธิภาพของเครื่องคอมพิวเตอร์มีผลต่อความเร็วในการทำงานของอัลกอริทึม กล่าวคือ เครื่อง คอมพิวเตอร์ที่มีประสิทธิภาพในการทำงานสูงย่อมให้ผลลัพธ์การทำงานที่รวดเร็วกว่าเครื่องที่มี ประสิทธิภาพในการทำงานต่ำ เป็นผลให้ไม่สามารถเปรียบเทียบการทำงานของอัลกอริทึมเดียวกัน ในเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพแตกต่างกันได้
4. เนื่องจาก Ant Colony Optimization เป็นการทำงานเพื่อหาผลลัพธ์ที่เหมาะสมที่สุดที่สามารถ ยอมรับได้ ไม่ใช่การหาผลลัพธ์ที่ดีที่สุดที่เกิดขึ้นได้ ทำให้ในบางครั้งผลลัพธ์ที่ได้ไม่ใช่ผลลัพธ์ที่ดี ที่สุด เช่น มีการทำงานระบบโดยใช้จำนวนรอบของการทำงานที่ 100 รอบ แต่หากทำงานต่อไปอีก 200 รอบจะได้ผลลัพธ์ที่ดีกว่า เป็นต้น

## บรรณานุกรม

- Cay S. Horstmann and Gary Cornell. 2007. **Core Java™, Volumn I-Fundamentals, Eighth Edition.** Prentice Hall.
- Cay S. Horstmann and Gary Cornell. 2008. **Core Java™, Volumn II-Advance Features, Eighth Edition.** Prentice Hall.
- Marco Dorigo. **The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances.** [Online] Available: <http://www.agent.ai/doc/upload/200302/dori02.pdf>
- Marco Dorigo and Thomas Stützle. 2004. **Ant colony optimization.** MIT Press.
- Russell Miles and Kim Hamilton. 2005. **Learning UML 2.0.** O'Reilly Media, Inc.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อ-นามสกุล	นายอนล ชำนาญเวช
วัน เดือน ปีเกิด	7 ธันวาคม 2526 อุบลราชธานี
ที่อยู่	17/7 ซอยวิภาวดี-รังสิต 40 แขวงลาดยาว เขตจตุจักร กรุงเทพมหานคร 10400
ประวัติการศึกษา	
ระดับปริญญาตรี	ศิลปศาสตรบัณฑิต คอมพิวเตอร์ธุรกิจ คณะบริหารธุรกิจ มหาวิทยาลัยราชภัฏสวนดุสิต
ประสบการณ์ทำงาน	
พ.ศ. 2549-2550	บริษัท เทก้า ซอร์ฟ จำกัด
พ.ศ. 2550-2552	บริษัท ไทย เทรค เน็ต จำกัด
พ.ศ. 2552-ปัจจุบัน	บริษัท ชิงก์ สมาร์ท จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้