



## รายงานการวิจัยฉบับสมบูรณ์

อัลกอริทึมอาณานิคมมดสำหรับการจัดเส้นทางของยานพาหนะในการจัดส่งสินค้า

Ant Colony Optimization Algorithm for Vehicle Routing Problems

รศ.ดร.อาริต ธรรมโน

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ พ.ศ. 2561

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ(ภาษาไทย) อัลกอริทึมอาณานิคมมดสำหรับการจัดเส้นทางของยานพาหนะในการ  
จัดส่งสินค้า

แหล่งเงิน เงินรายได้ คณะเทคโนโลยีสารสนเทศ สจล.

ประจำปีงบประมาณ 2561 จำนวนเงินที่ได้รับการสนับสนุน 50,000 บาท

ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ 1 ตุลาคม 2560 ถึง 30 กันยายน 2561

ชื่อ-สกุล หัวหน้าโครงการ และ ผู้ร่วมโครงการวิจัย พร้อมระบุหน่วยงานต้นสังกัด

1. รศ.ดร. อาริต ธรรมโน คณะเทคโนโลยีสารสนเทศ สจล. Email: arit@it.kmitl.ac.th

บทคัดย่อ

ปัญหาการจัดเส้นทางการขนส่ง (Vehicle routing problem) แบ่งปัญหาออกเป็น 2 ส่วนย่อย คือ 1) จะทำการจัดการส่งสินค้าอย่างไรให้ครบลูกค้าทุกคน เพื่อสร้างความพึงพอใจด้านการส่งมอบให้กับลูกค้า และ 2) จะทำการจัดส่งสินค้าอย่างไร โดยใช้ระยะทางสั้นที่สุด เพื่อลดต้นทุนการขนส่ง ในงานวิจัยนี้ได้ประยุกต์อัลกอริทึมทางปัญญาประดิษฐ์ในกลุ่มความฉลาดทางชีววิทยา (Biologically inspired) ซึ่งมีหลักการ คือ การเลียนแบบวิถีทางธรรมชาติ และการดำรงอยู่ของสิ่งมีชีวิต ที่มีชื่อว่า Ant system (AS) algorithm และ ในกลุ่มความฉลาดทางฟิสิกส์ (Physics-inspired) ที่มีชื่อว่า simulated annealing (SA) algorithm มาแก้ปัญหาการจัดเส้นทางการขนส่ง ซึ่ง AS เป็นอัลกอริทึมที่เลียนแบบพฤติกรรมกรหาอาหารของมด ส่วน SA เป็นอัลกอริทึมอาศัยความรู้จากกระบวนการทางฟิสิกส์ระหว่างการอบเหนียว (annealing) ซึ่งเป็นขั้นตอนการลดอุณหภูมิระหว่างการหลอมโลหะเพื่อให้ได้โลหะที่อยู่ในสภาวะที่เหมาะสมที่สุด โดยเป้าหมายในการประยุกต์ใช้อัลกอริทึม AS และ SA ร่วมกัน คือ นำข้อดีของทั้ง 2 อัลกอริทึมมาทำให้การค้นหาคำตอบของปัญหามีความเหมาะสมมากยิ่งขึ้น

คำสำคัญ: Combinatorial optimization problem, Vehicle routing problem, Ant system, Simulated annealing

Research Title: Ant Colony Optimization Algorithm for Vehicle Routing Problems

Researcher: Arit Thammano

Faculty: Information Technology

Department: Information Technology

## Abstract

Vehicle Routing Problem can be divided into 2 sub-problems: 1) Manage the delivery of products to all customers on time to satisfy customer needs and 2) Make the delivery by using the shortest route to reduce transportation costs. This research employs two powerful artificial intelligence techniques, Ant system (AS) algorithm and Simulated annealing (SA) algorithm. AS is inspired by the foraging behavior of ant colony while SA is inspired by the annealing process of steel. Annealing is the process of reducing the temperature during the melting process in order to obtain the metal that is in the optimal (best) state. The aim of this proposed hybrid algorithm is to combine the advantages of the two techniques to overcome the limitations of each individual technique.

Keywords: Combinatorial optimization problem, Vehicle routing problem, Ant system, Simulated annealing

## กิตติกรรมประกาศ

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากเงินรายได้ประจำปีงบประมาณ 2561 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (รหัสโครงการ/รหัสสัญญาวิจัย 2561-02-06005)

รศ.ดร.อาริต ธรรมโน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อ.....	I
Abstract.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของงานวิจัย.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนของการดำเนินงานวิจัย.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ปัญหาออปติไมเซชัน.....	3
2.1.1 ประเภทของปัญหาออปติไมเซชันที่แบ่งตามจำนวนข้อจำกัด.....	3
2.1.2 ประเภทของปัญหาออปติไมเซชันที่แบ่งตามจำนวนฟังก์ชันวัตถุประสงค์.....	4
2.1.3 ประเภทของปัญหาออปติไมเซชันที่แบ่งตามชนิดของพารามิเตอร์.....	5
2.2 ปัญหาออปติไมเซชันเชิงการจัด.....	5
2.3 ปัญหาการเดินทางของพนักงานขาย.....	6
2.4 ปัญหาการจัดเส้นทางการขนส่ง.....	7
2.5 งานวิจัยที่เกี่ยวข้อง.....	9
2.5.1 A new hybrid method based on Particle Swarm Optimization Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem.....	9
2.5.2 Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem.....	11
2.5.3 An improved ant colony optimization for vehicle routing problem.....	14
2.5.4 An ant colony optimization model: The period vehicle routing problem with time windows.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

2.5.5 Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution .....	19
2.5.6 A discrete firefly algorithm to solve a rich vehicle routing problem modeling a newspaper distribution system with recycling policy .....	20
2.5.7 Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands .....	22
2.4.8 A novel two-level optimization approach for clustered vehicle routing problem .....	26
บทที่ 3 การแก้ปัญหาคำสั่งเส้นทางขนส่งโดยประยุกต์ใช้อัลกอริทึมอาณานิคมมด.....	30
3.1 ขั้นตอนการกำหนดค่าเริ่มต้น .....	30
3.2 ขั้นตอนการสร้างตัวเริ่มต้น .....	30
3.3 ขั้นตอนการปรับปรุงความหนาแน่นของฟีโรโมน .....	31
3.4 ขั้นตอนการทำสำเนาตัวเริ่มต้น .....	32
3.5 ขั้นตอนการปรับปรุงตัวเริ่มต้น .....	35
3.6 ขั้นตอนการพิจารณาหยุดการค้นหาค้นหา.....	38
บทที่ 4 การทดลองและผลการทดลอง .....	40
4.1 ชุดข้อมูลที่ใช้ในการทดลอง.....	40
4.2 ผลการทดลองเปรียบเทียบประสิทธิภาพ.....	41
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ .....	45
5.1 สรุปผลการทดลอง .....	45
5.2 ข้อเสนอแนะ.....	45
บทที่ 6 สรุปผลผลิตงานวิจัย.....	46
6.1 สรุปรายชื่อและรายละเอียดผลผลิตงานวิจัยที่ผลิตได้และที่อยู่ระหว่างดำเนินการ .....	46
บรรณานุกรม .....	47
ภาคผนวก	
สรุปค่าใช้จ่ายการดำเนินโครงการวิจัย.....	49
ประวัตินักวิจัย .....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
4.1 ชุดข้อมูลที่ใช้ในการทดลอง.....	40
4.2 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมพื้นฐาน .....	41
4.3 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมอื่น ๆ.....	43



## สารบัญภาพ

ภาพที่	หน้า
2.1 ตัวอย่างปัญหาการเดินทางของพนักงานขาย.....	7
2.2 ขั้นตอนการทำงานของ SOS-SA.....	14
2.3 กราฟแบบไม่มีทิศทางที่แสดงถึงการแบ่งกลุ่มของลูกค้า.....	26
2.4 global route ที่ได้จากตัวอย่างในภาพที่ 2.3.....	27
2.5 การเพิ่มค่า M ในเส้นทาง TSP.....	29
3.1 ตัวอย่างของทัวร์เริ่มต้น.....	31
3.2 ตัวอย่างความยาวของทัวร์เริ่มต้น (อ้างอิง รูปที่ 3.1).....	31
3.3 ตัวอย่างการจัดเรียงทัวร์เริ่มต้น (อ้างอิง รูปที่ 3.2).....	32
3.4(1) ตัวอย่างทัวร์กรณีที่ 1.....	33
3.4(2) การสุ่มเลือก 2 ตำแหน่งกรณีที่ 1.....	33
3.4(3) แลกเปลี่ยนลูกค้าในตำแหน่งที่สุ่มกรณีที่ 1.....	33
3.4(4) ผลลัพธ์การสำเนาทัวร์กรณีที่ 1.....	34
3.5(1) ตัวอย่างทัวร์กรณีที่ 2.....	34
3.5(2) การสุ่มเลือก 2 ตำแหน่งกรณีที่ 2.....	34
3.5(3) แลกเปลี่ยนลูกค้าในตำแหน่งที่สุ่มกรณีที่ 2.....	34
3.5(4) คัดลอกตำแหน่งก่อนหน้ามาเดินก่อนกรณีที่ 2.....	34
3.5(5) ผลลัพธ์การสำเนาทัวร์กรณีที่ 2.....	34
3.6 สุ่มตำแหน่งที่ 6 ของลูกค้าที่ต้องการลบ.....	35
3.7 สุ่มตำแหน่งที่ 3 ของลูกค้าที่ต้องการแทรก.....	35
3.8 ผลลัพธ์วิธี Insert mutation จากกรณีที่ 1.....	36
3.9 ผลลัพธ์วิธี Insert mutation จากกรณีที่ 2.....	36
3.10 สุ่มตำแหน่งที่ 1.....	36
3.11 สุ่มตำแหน่งที่ 2.....	36
3.12 กลุ่มลูกค้าที่ถูกเลือกเพื่อสลับ และ แทนที่.....	37
3.13 ผลลัพธ์วิธี Inverse mutation.....	37
3.14 แต่ละคู่ของตำแหน่งที่สุ่ม.....	37
3.15 ผลลัพธ์วิธี Exchange mutation.....	37
4.1 ตัวอย่างบางส่วนของชุดข้อมูล cil51.....	41

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการแข่งขันในตลาดมีมากขึ้น การดำเนินธุรกิจทั้งขนาดเล็ก กลาง หรือใหญ่ล้วนมีเป้าหมายเพื่อตอบสนองความต้องการของลูกค้าได้อย่างสูงสุดทั้งในด้านคุณภาพและการส่งมอบ ซึ่งการขนส่งหรือการกระจายสินค้าที่ตรงต่อเวลาเป็นส่วนสำคัญที่จะสร้างความพึงพอใจด้านการส่งมอบให้กับลูกค้า ยิ่งไปกว่านั้นการขนส่งที่ตรงต่อเวลานั้นควรจะต้องเป็นการขนส่งที่มีต้นทุนต่ำ ระยะทางในการขนส่งสินค้าเป็นปัจจัยที่มีผลต่อต้นทุนการขนส่ง ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่จะต้องศึกษาและหาวิธีการที่นำมาใช้ในการจัดเส้นทางขนส่งให้มีระยะทางสั้นที่สุด ซึ่งทำให้สามารถลดต้นทุนการขนส่ง และส่งผลให้บริษัทมีความสามารถในการแข่งขันกับบริษัทคู่แข่งได้มากขึ้น พร้อมก้าวไปสู่การเป็นองค์กรแถวหน้าในธุรกิจผู้ให้บริการโลจิสติกส์ รวมไปถึงการให้บริการที่เต็มศักยภาพ

จากปัญหาที่ว่า จะทำอย่างไรให้บริษัทสามารถวางแผนการจัดเส้นทางขนส่งให้ประสบความสำเร็จ จึงมีนักวิจัยจำนวนมากพยายามประยุกต์ใช้ปัญญาประดิษฐ์ซึ่งเป็นศาสตร์หนึ่งทางคอมพิวเตอร์มาใช้แก้ปัญหาการจัดเส้นทางขนส่ง (Vehicle Routing Problems: VRP) โดยอัลกอริทึมที่ถูกกล่าวถึง คือ อัลกอริทึมในกลุ่มความฉลาดทางชีววิทยา (Biologically inspired algorithm) ซึ่งมีหลักการ คือ การเลียนแบบวิถีทางธรรมชาติและการดำรงอยู่ของสิ่งมีชีวิต ตัวอย่างอัลกอริทึมในกลุ่มนี้ เช่น อัลกอริทึมวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต (Genetic algorithm: GA) อัลกอริทึมการหาอาหารของฝูงนกหรือฝูงปลา (Particle Swarm Optimization algorithm: PSO) และอัลกอริทึมอาณานิคมมด (Ant Colony Optimization: ACO) เป็นต้น

ในงานวิจัยนี้ได้ประยุกต์อัลกอริทึมในกลุ่มความฉลาดทางชีววิทยา (Biologically inspired algorithm) เพื่อใช้แก้ปัญหาการจัดเส้นทางขนส่ง

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อศึกษาปัญหาการจัดเส้นทางขนส่ง และศึกษางานวิจัยที่เกี่ยวข้อง
- 1.2.2 เพื่อศึกษาอัลกอริทึมและทฤษฎีอื่น ๆ ที่จะนำมาใช้ในงานวิจัย
- 1.2.3 เพื่อพัฒนาอัลกอริทึมในการแก้ปัญหาการจัดเส้นทางขนส่ง

### 1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานจาก A Traveling Salesman Problem Library (TSPLIB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.4 ประโยชน์ที่คาดว่าจะได้รับ
  - 1.4.1 มีความรู้ความเข้าใจปัญหาการจัดเส้นทางรถขนส่ง
  - 1.4.2 มีความรู้ความเข้าใจขั้นตอนการทำงานของอัลกอริทึม และทฤษฎีต่าง ๆ ที่นำมาประยุกต์ใช้
  - 1.4.3 ได้แนวทางพัฒนาอัลกอริทึมที่มีประสิทธิภาพในการแก้ปัญหาการจัดเส้นทางรถขนส่ง
  
- 1.5 ขั้นตอนของการดำเนินงานวิจัย
  - 1.5.1 ศึกษาปัญหาการจัดเส้นทางรถขนส่ง และศึกษางานวิจัยอื่น ๆ ที่เกี่ยวข้อง
  - 1.5.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับอัลกอริทึมจัดการเส้นทางรถขนส่ง
  - 1.5.3 พัฒนาอัลกอริทึมจัดการเส้นทางรถขนส่ง
  - 1.5.4 ดำเนินการทดลองและทดสอบประสิทธิภาพของอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐาน A Traveling Salesman Problem Library (TSPLIB) และเปรียบเทียบผลลัพธ์ที่ได้กับผลลัพธ์จากงานวิจัยอื่นที่ทดสอบกับข้อมูลชุดเดียวกัน
  - 1.5.5 สรุปผลการทดลองและผลการเปรียบเทียบ
  - 1.5.6 จัดทำเอกสารประกอบการทำเล่มงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะแบ่งเนื้อหาออกเป็น 2 ส่วนคือ ส่วนแรกกล่าวถึงปัญหาออปติไมเซชัน และปัญหาการจัดเส้นทางขนส่ง ส่วนที่ 2 กล่าวถึงรายละเอียดของงานวิจัยที่เกี่ยวข้อง

#### 2.1 ปัญหาออปติไมเซชัน

ปัญหาออปติไมเซชัน (Optimization problem) คือ การค้นหาค่าพารามิเตอร์ที่เหมาะสมที่สุดที่ส่งผลให้การหาค่าฟังก์ชันวัตถุประสงค์ (Objective function) ได้ค่าที่ดีที่สุดภายใต้ข้อจำกัด (Constraint) ของปัญหา โดยองค์ประกอบหลักของปัญหาออปติไมเซชัน คือ พารามิเตอร์และฟังก์ชันวัตถุประสงค์ สำหรับข้อจำกัดของปัญหาอาจมีหรือไม่มีก็ได้ขึ้นอยู่กับปัญหาที่ทำการศึกษา โดยปัญหาออปติไมเซชันสามารถแบ่งประเภทได้ดังนี้

##### 2.1.1 ประเภทของปัญหาออปติไมเซชันที่แบ่งตามจำนวนข้อจำกัด

ปัญหาออปติไมเซชันหากแบ่งตามจำนวนข้อจำกัดสามารถแบ่งได้เป็น 2 ประเภท คือ ปัญหาออปติไมเซชันแบบไม่มีข้อจำกัด (Unconstrained optimization problem) และปัญหาออปติไมเซชันแบบมีข้อจำกัด (Constrained optimization problem)

1) ปัญหาออปติไมเซชันแบบไม่มีข้อจำกัด คือ การค้นหาพารามิเตอร์ที่เหมาะสมที่สุดที่ทำให้ฟังก์ชันวัตถุประสงค์มีค่าดีที่สุด โดยไม่กำหนดขอบเขตของค่าพารามิเตอร์ รูปแบบทั่วไปของปัญหาออปติไมเซชันแบบไม่มีข้อจำกัด แสดงได้ดังสมการ (2.1)

$$\text{minimize } f(\bar{x}) = f(x_1, x_2, \dots, x_n) \quad (2.1)$$

เมื่อ  $\bar{x}$  คือ เซตของพารามิเตอร์ที่มีสมาชิกจำนวน  $n$  ตัว

$f(\cdot)$  คือ ฟังก์ชันวัตถุประสงค์

2) ปัญหาออปติไมเซชันแบบมีข้อจำกัด คือ การค้นหาพารามิเตอร์ที่เหมาะสมที่สุดที่ทำให้ฟังก์ชันวัตถุประสงค์มีค่าดีที่สุด โดยมีการกำหนดขอบเขตของค่าพารามิเตอร์บางตัวหรือทั้งหมด ข้อจำกัดใช้ในการกำหนดขอบเขตของปัญหาออปติไมเซชัน แบ่งได้เป็น 2 กลุ่ม คือ ข้อจำกัดแบบเท่ากัน เป็นการกำหนดความสัมพันธ์ของพารามิเตอร์บางตัวให้เท่ากับเงื่อนไขของปัญหา ส่วนข้อจำกัดแบบไม่เท่ากัน เป็นการกำหนดความสัมพันธ์ของพารามิเตอร์บางตัวให้น้อยกว่าหรือมากกว่าเงื่อนไขของปัญหา รูปแบบทั่วไปของปัญหาออปติไมเซชันแบบมีข้อจำกัด แสดงได้ดังสมการ (2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{minimize } f(\bar{x}) = f(x_1, x_2, \dots, x_n) \quad (2.2)$$

$$\text{Subject to } g_i(\bar{x}) = 0, \quad i = 1, \dots, k$$

$$h_j(\bar{x}) \leq 0, \quad j = 1, \dots, p$$

เมื่อ  $\bar{x}$  คือ เซตของพารามิเตอร์ที่มีสมาชิกจำนวน  $n$  ตัว

$f(\cdot)$  คือ ฟังก์ชันวัตถุประสงค์

$g_i$  คือ ข้อจำกัดแบบเท่ากัน ซึ่งมีจำนวน  $k$  สมการ

$h_j$  คือ ข้อจำกัดแบบไม่เท่ากัน ซึ่งมีจำนวน  $p$  สมการ

### 2.1.2 ประเภทของปัญหาอพติไมเซชันที่แบ่งตามจำนวนของฟังก์ชันวัตถุประสงค์

การแบ่งประเภทปัญหาอพติไมเซชันตามจำนวนฟังก์ชันวัตถุประสงค์สามารถแบ่งได้เป็น 2 ประเภท คือ ปัญหาอพติไมเซชันแบบหนึ่งเป้าหมาย (Single objective optimization problem) และ ปัญหาอพติไมเซชันแบบหลายเป้าหมาย (Multi objective optimization problem)

1) ปัญหาอพติไมเซชันแบบหนึ่งเป้าหมาย คือ การหาค่าพารามิเตอร์ที่เหมาะสมที่สุดที่ทำให้ฟังก์ชันวัตถุประสงค์มีค่าดีที่สุด รูปแบบของปัญหาอพติไมเซชันแบบหนึ่งเป้าหมาย [1] แสดงได้ดังสมการ (2.3)

$$\text{minimize } F(\bar{x}); \bar{x} = (x_1, x_2, \dots, x_n) \in \mathfrak{R}^n \quad (2.3)$$

เมื่อ  $\bar{x}$  คือ เซตของพารามิเตอร์ที่มีสมาชิกจำนวน  $n$  ตัว

$F(\cdot)$  คือ ฟังก์ชันวัตถุประสงค์

$S$  คือ ขอบเขตพื้นที่การค้นหาที่กำหนด

เมื่อ  $\bar{x} \in \Omega \subseteq S$  โดยฟังก์ชันวัตถุประสงค์  $F$  เป็นกำหนด  $S \subseteq \mathfrak{R}^n$  และ  $\Omega \subseteq S$  กำหนดความเป็นไปได้ของบริเวณ โดยที่  $S$  กำหนดเป็น  $n$ -dimensional พื้นที่สี่เหลี่ยมใน  $\mathfrak{R}^n$  และขอบเขตของตัวแปรกำหนดขอบบนและล่างเป็น  $x_j^L \leq x_j \leq x_j^U, j = 1, 2, \dots, n$ .

2) ปัญหาอพติไมเซชันแบบหลายเป้าหมาย คือ การหาค่าพารามิเตอร์ที่เหมาะสมที่สุดที่ทำให้ฟังก์ชันวัตถุประสงค์มีค่าดีที่สุด โดยการพิจารณาระดับความสำคัญของฟังก์ชันวัตถุประสงค์ที่แตกต่างกัน การหาค่าตอบที่ดีที่สุดโดยให้ระดับความสำคัญของทุกฟังก์ชันวัตถุประสงค์เท่ากันเป็นสิ่งที่เป็นไปได้ยากเนื่องจากฟังก์ชันวัตถุประสงค์แต่ละฟังก์ชันมีค่าต่ำสุดหรือค่าสูงสุด ณ จุดที่แตกต่างกัน รูปแบบของปัญหาอพติไมเซชันแบบหลายเป้าหมาย [2] แสดงได้ดังสมการ (2.4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{minimize}_{x \in \Omega} \{f_1(x), f_2(x), \dots, f_k(x)\} \quad (2.4)$$

เมื่อ  $x$  คือ เวกเตอร์ของตัวแปรตัดสินใจ  
 $\Omega$  คือ บริเวณคำตอบที่เป็นไปได้  
 $f_i(x)$  คือ ฟังก์ชันวัตถุประสงค์ที่  $i$  โดย  $i = 1, 2, \dots, k$

ดังนั้นคำตอบที่ดีที่สุดของปัญหาออปติไมเซชันแบบหนึ่งเป้าหมายจะมีเพียงหนึ่งคำตอบในขณะที่ปัญหาออปติไมเซชันแบบหลายเป้าหมายมีคำตอบที่ดีที่สุดมากกว่าหนึ่งคำตอบ โดยขึ้นอยู่กับข้อกำหนดระดับความสำคัญของฟังก์ชันวัตถุประสงค์ที่แตกต่างกัน เซตของคำตอบที่ดีที่สุดนี้เรียกว่า ชุดคำตอบพारेโต (Pareto optimal solutions) โดยสมาชิกในชุดคำตอบนี้มีคุณลักษณะที่สำคัญ คือ ไม่สามารถเพิ่มค่าฟังก์ชันวัตถุประสงค์ฟังก์ชันใดฟังก์ชันหนึ่งให้ดีขึ้น โดยไม่มีการลดค่าฟังก์ชันวัตถุประสงค์ของฟังก์ชันอื่นๆ ให้น้อยลง

### 2.1.3 ประเภทของปัญหาออปติไมเซชันที่แบ่งตามชนิดข้อมูลของพารามิเตอร์

การแบ่งประเภทปัญหาออปติไมเซชันแบ่งตามชนิดข้อมูลของพารามิเตอร์สามารถแบ่งได้เป็น 2 ประเภท คือ ปัญหาออปติไมเซชันเชิงตัวเลข (Numerical optimization problem) และปัญหาออปติไมเซชันเชิงการจัด (Combinatorial optimization problem)

1) ปัญหาออปติไมเซชันเชิงตัวเลข คือ ปัญหาออปติไมเซชันที่มีค่าพารามิเตอร์แบบต่อเนื่อง (Continuous optimization problem) ซึ่งมีค่าเป็นเลขจำนวนจริง เช่น ต้นทุนต่อหน่วยในการผลิตสินค้า เวลาที่ใช้ในการผลิตสินค้าต่อชิ้น และจำนวนไมล์ที่รถวิ่งไปได้ต่อน้ำมัน 1 แกลลอนของรถชนิดหนึ่ง เป็นต้น

2) ปัญหาออปติไมเซชันเชิงการจัด คือ ปัญหาออปติไมเซชันที่มีค่าพารามิเตอร์แบบไม่ต่อเนื่อง (Discrete optimization problem) เช่น ปัญหาการมอบหมายงาน (Assignment problem) ปัญหาการเดินทางของพนักงานขาย (Traveling salesman problems) ปัญหาการจัดตารางงาน (Scheduling problem) และปัญหาการจัดเส้นทางรถขนส่ง (Vehicle routing problem) เป็นต้น

## 2.2 ปัญหาออปติไมเซชันเชิงการจัด

ปัญหาออปติไมเซชันเชิงการจัด (Combinatorial optimization problem) มีเป้าหมายเพื่อค้นหาโซลูชันที่เหมาะสมมากที่สุด ในขอบเขตของเงื่อนไขที่กำหนด ลักษณะของปัญหาจะเป็นการจัดเรียง (Arrangement), การจัดกลุ่ม (Grouping), การทำตามคำสั่ง (Ordering) หรือการเลือก (Selection) โซลูชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการประเมินความเหมาะสมของโซลูชันอาจจะประเมินจากค่าต่ำที่สุดหรือค่าสูงที่สุดขึ้นอยู่กับปัญหาที่ศึกษา พื้นฐานของปัญหาอพติไมเซชันเชิงการจัดจะมี 3 ส่วนประกอบ [3] ที่สำคัญดังนี้

- $S$  แสดงถึง ขอบเขตพื้นที่การค้นหาที่กำหนด (Search space) และ  $s$  คือ โซลูชันที่เป็นไปได้ (Feasible solution) ที่อยู่ในขอบเขตพื้นที่การค้นหาที่กำหนด
- $f$  ฟังก์ชันวัตถุประสงค์ (Objective function) ที่สามารถเป็นได้ทั้งค่ามากที่สุด (Maximize) หรือค่าน้อยที่สุด (Minimize)
- $\Omega$  เซตของข้อจำกัดที่จะต้องปฏิบัติตามเพื่อให้สามารถแก้ปัญหาได้

อธิบายตามเงื่อนไข  $\{s^* \in S: f(s^*) \leq f(s), \forall s \in S\}$  [4] ได้ว่าโซลูชันที่มีความเหมาะสม ( $s^*$ ) จะถูกค้นหาภายในขอบเขตพื้นที่ที่กำหนดไว้ โดยฟังก์ชันวัตถุประสงค์ของ  $s^*$  จะต้องมีความเหมาะสมมากกว่าโซลูชัน  $s$  ทุกตัวตามเงื่อนไข  $\Omega$  และเป็นสมาชิกใน  $S$  โดยปัญหาอพติไมเซชันเชิงการจัดที่จะกล่าวถึง คือ ปัญหาการจัดเส้นทางรถขนส่ง

### 2.3 ปัญหาการเดินทางของพนักงานขาย

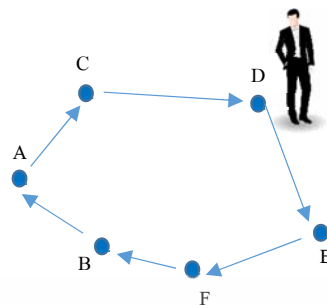
สำหรับปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problems : TSP) เป็นลักษณะของปัญหาการจัดเส้นทางรถขนส่งให้กับพนักงานขาย โดยข้อกำหนดของปัญหาคือการจัดเส้นทางอย่างไรเพื่อให้พนักงานขายมีระยะทางรวม/ค่าใช้จ่ายในการเดินทางไปหาลูกค้าต่ำที่สุด โดยเส้นทางสำหรับพนักงานขายนี้จะต้องเดินทางหาลูกค้าให้ครบทุกเมือง(ราย)โดยไม่ต้องเดินทางไม่ซ้ำเมืองเดิม ในการคำนวณระยะทางรวม/ค่าใช้จ่ายของพนักงานขาย [5] สามารถคำนวณได้ดังสมการที่ 2.5

$$\text{Minimize } z = \sum_{i=1}^n \sum_{j=1, i \neq j}^n d_{ij} x_{ij} \quad (2.5)$$

โดย

- $n$  คือ จำนวนเมืองหรือลูกค้าที่ต้องเดินทางไปพบ
- $i$  คือ เมือง (ลูกค้า) ต้นทาง
- $j$  คือ เมือง (ลูกค้า) ปลายทาง
- $d_{ij}$  คือ ระยะทาง/ค่าใช้จ่ายในการเดินทางจากเมือง  $i$  ไป  $j$
- $x_{ij}$  เป็นตัวแปรไบนารีที่มีค่าเป็น 0 หรือ 1 ซึ่งใช้เพื่อพิจารณาลักษณะการเดินทาง โดยจะมีค่าเป็น 1 เมื่อพนักงานเดินทางจากเมือง  $i$  ไป  $j$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงตัวอย่างปัญหาการเดินทางของพนักงานขาย

## 2.4 ปัญหาการจัดเส้นทางรถขนส่ง

ปัญหาการจัดเส้นทางรถขนส่ง (Vehicle Routing Problem) เป็นปัญหาออปติไมเซชันเชิงการจัดซึ่งอาจเป็นปัญหาที่มีฟังก์ชันวัตถุประสงค์เดียว (Single objective) หรือ ฟังก์ชันหลายวัตถุประสงค์ (Multi objective) แบบค่าน้อยที่สุด (Minimization) และเป็นปัญหาเอ็นพีแบบยาก (NP-hard) ที่เกี่ยวข้องกับการจัดเส้นทางรถขนส่ง [6]

ปัญหาการจัดเส้นทางรถขนส่ง เป็นปัญหาที่มีลักษณะคล้ายกับปัญหาการเดินทางของพนักงานขาย และยังมีความสัมพันธ์เกี่ยวกับการหาทำเลที่ตั้งและการมอบหมาย แต่ปัญหาการจัดเส้นทางรถขนส่ง อาจมองว่าเป็นการต่อยอดมาจากการหาทำเลที่ตั้งที่เหมาะสมเมื่อจัดการและมอบหมายลูกค้าเรียบร้อยแล้ว ต่อมาพิจารณาจัดเส้นทางที่เหมาะสมตามศักยภาพของยานพาหนะ เช่น จะขนส่งกี่คัน ยานพาหนะจำนวนเท่าไร ยานพาหนะคันไหนจะไปส่งสินค้าให้กับลูกค้าใดบ้าง ใช้เส้นทางใด ในกรณีพาหนะขนส่งคันเดียว หรือพาหนะขนส่งหลายคัน จะต้องเพิ่มข้อจำกัดต่าง ๆ เข้าไปตามกรณีหรือตามเหตุการณ์จริงที่เกิดขึ้น เช่น ข้อจำกัดด้านเวลาในการรับส่ง (Vehicle Routing Problem with Time Windows) กล่าวคือเป็นข้อจำกัดของลูกค้าว่าจะรับสินค้าได้ภายในเวลาใด ปัญหาการจัดเส้นทางรถขนส่งโดยปกติวางแผนไปรับหรือไปส่งสินค้าเท่านั้น หรือไม่อาจจะไปรับและส่งสินค้าในคราวเดียวกัน ระยะทางในการขนส่งมีทั้งแบบทราบค่าที่แน่นอน และทราบค่าที่ไม่แน่นอน หรือไม่ทราบค่า สามารถแบ่งกลุ่มปัญหาการจัดเส้นทางรถขนส่งได้ดังต่อไปนี้

### 1. จัดกลุ่มตามวิธีการแก้ปัญหา ประกอบด้วย

1.1 วิธีการแม่นยำ (Exact method) ทำให้ได้ค่าที่ดีที่สุด ใช้วิธี Cutting Plane method และวิธี Branch and Bound method

1.2 วิธีการฮิวริสติกส์ (Heuristics) จะได้ค่าที่ดีแต่ไม่รับประกันว่าจะได้คำตอบที่ดีที่สุด และใช้เวลาสั้นกว่าวิธีการแบบแม่นยำ สำหรับปัญหาที่มีขนาดใหญ่ใช้วิธี เช่น Genetic Algorithm, Ant System Algorithm, Particle Swarm Optimization และอื่น ๆ

1.3 การจำลองแบบปัญหา (Simulation) ส่วนใหญ่ใช้กับปัญหาที่มีความไม่แน่นอน เช่น ความต้องการไม่แน่นอน ระยะทางให้บริการไม่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. จัดกลุ่มตามลักษณะความต้องการของลูกค้า ประกอบด้วย

2.1 ค่าความต้องการของลูกค้าทราบค่าและแน่นอน (Deterministic Demand) คือ มีการสั่งซื้อสินค้าก่อนและจัดเส้นทางขนส่ง หรือ ประเมินค่าจากการใช้ค่าเฉลี่ย หรือ ค่าทางสถิติอย่างใดอย่างหนึ่ง

2.2 ค่าความต้องการของลูกค้าทราบค่าแต่ไม่ทราบค่าที่แน่นอน (Stochastic demand)

2.3 ไม่ทราบค่าความต้องการของลูกค้า ซึ่งจะไม่ทราบค่าในขณะวางแผน แต่จะทราบเมื่อเดินทางไปถึงลูกค้า

3. จัดกลุ่มตามข้อจำกัดด้านเวลา (Time windows) เป็นข้อจำกัดที่สำคัญกับการจัดเส้นทางเนื่องจากบางครั้งเวลาให้บริการลูกค้า กับเวลาในการเดินทางมีผลต่อเส้นทางที่ได้จากการจัดด้วยวิธีการต่าง ๆ ประกอบด้วย

3.1 แบบไม่มีข้อจำกัดด้านเวลา (No time windows) ซึ่งจัดเฉพาะเส้นทางการเดินทาง

3.2 แบบไม่เคร่งครัด (Soft time windows) สามารถส่งสินค้าช้าหรือเร็วกว่ากำหนดได้บ้าง

3.3 แบบเคร่งครัด (Stick time windows) กำหนดถึงระยะเวลาในการให้บริการและระยะเวลาในการเดินทาง

3.4 แบบมีทั้งเคร่งครัดและไม่เคร่งครัด (Mixed) ไม่เคร่งครัดเรื่องเวลามาถึงของรถบรรทุกหรือเวลาในการให้บริการ และไม่เคร่งครัดเรื่องเวลาในปัญหาเดียวกัน

4. จัดกลุ่มตามเวลาในการวางแผนการเดินทาง (Time Horizon) เน้นจัดกลุ่มแบบจัดแบบครั้งเดียวในการวางแผน 1 ครั้ง เช่น ส่งสินค้าทุกวันจะเดินทางด้วยเส้นทางเดียวกัน และวางแผนเป็นเดือนหรือปี แต่ละวันอาจจะเดินทางที่ไม่เหมือนกัน

4.1 แบบคาบเวลาเดียว (Single period) จะวางแผนครั้งเดียวและดำเนินการเช่นเดียวกันในทุกคาบเวลา

4.2 แบบหลายคาบเวลา (Multi Period) วางแผนหลายคาบเวลา เส้นทางเดินทางที่แตกต่างกันไปในแต่ละคาบเวลา

## 5. จัดกลุ่มตามจำนวนของจุดเริ่มต้น (Number of origin point) ประกอบด้วย

5.1 มีจุดเริ่มต้นเดียว (Single origin/depot) ทุกเส้นทางจะเริ่มต้นจากจุดกระจายสินค้าเพียงแห่งเดียว

5.2 มีจุดเริ่มต้นหลายจุด (Multiple origin/depot) วางแผนให้มีศูนย์กระจายสินค้าหลายแห่ง โดยทำการจัดเส้นทางไปพร้อม ๆ กัน

6. อื่น ๆ อาจจะจัดกลุ่มตามลักษณะของการส่งหรือรับอย่างเดียว หรือมีทั้งการส่งและการรับ จำนวนพาหนะที่ใช้ ข้อจำกัดด้านระยะทางสูงสุด หรือ จำนวนลูกค้าสูงสุดที่เดินทางไปได้

ช่วงแรกปัญหาการจัดเส้นทางการขนส่ง ส่วนใหญ่จะแก้ปัญหาด้วยวิธีการแบบแม่นยำ (Exact method) แต่ช่วงหลังหันมาสนใจพัฒนาวิธีการเมตาฮิวริสติกส์เพื่อช่วยแก้ปัญหาที่มีขนาดใหญ่ วิธีการที่นิยม ได้แก่ วิธีการเชิงพันธุกรรม วิธีการค้นหาแบบตามู วิธีการระบบมด และ อื่น ๆ

## 2.5 งานวิจัยที่เกี่ยวข้อง

2.5.1 A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem [7]

ในงานวิจัยนี้ ได้นำเสนอวิธีการผสม (Hybrid) โดยมีวัตถุประสงค์เพื่อหาค่าของพารามิเตอร์ที่เหมาะสมที่สุด ส่งผลต่อประสิทธิภาพของอัลกอริทึมวิธีการอาณานิคมมด (Ant Colony Optimization: ACO) โดยใช้อัลกอริทึมการหาอาหารของฝูงนกหรือฝูงปลา (Particle Swarm Optimization: PSO) ตรวจสอบค่าความเหมาะสมของพารามิเตอร์  $\alpha$  และ  $\beta$  ในอัลกอริทึม ACO สำหรับการเลือกลูกค้า และอัลกอริทึม 3-opt ใช้ปรับปรุงคำตอบเฉพาะที่ (Local solutions)

ขั้นตอนการทำงานของอัลกอริทึม ดังต่อไปนี้

1. กำหนดค่าพารามิเตอร์เริ่มต้น ซึ่งประกอบด้วย จำนวนมด จำนวนลูกค้า ฟิโรโมน และจำนวนรอบการทำงาน
2. กำหนดความยาวของทัวร์เริ่มต้น
3. กำหนดค่าฟิโรโมนจากสมการ

$$\tau_{ij}(t, t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^k(t, t+1) \quad (2.6)$$

$$\Delta\tau_{ij}^k(t, t+1) = \sum_{k=1}^n \Delta\tau_{ij}^k(t, t+1) \quad (2.7)$$

และ

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q}{L_k}, & \text{if } (i, j) \in \text{route performed by the } k\text{th ant} \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

4. หาค่าที่เหมาะสมที่สุดของพารามิเตอร์ ACO ( $\alpha, \beta$ ) ด้วย PSO
  - 4.1 ทำในขณะที่มดน้อยกว่าหรือเท่ากับจำนวนมดทั้งหมด
    - 4.1.1 กำหนดพารามิเตอร์สำหรับ PSO
 

(particle = จำนวนมด และกำหนด  $c_1, c_2$ )
    - 4.1.2 ในขณะที่ t น้อยกว่าจำนวนรอบ
      - 4.1.2.1 ในขณะที่ particle น้อยกว่ามด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.1.1 หากความน่าจะเป็นที่ทีมจะเดินไปลูกค้าที่เป็นไปได้  
จากสมการ

$$P_{i,j}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, & \text{if } j \text{ is allowed city} \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

4.1.2.1.2 ส่งพารามิเตอร์ของ ACO จาก PSO

ค้นหาทัวร์ที่ดีที่สุด

4.1.2.1.3 คำนวณค่า  $\alpha$  และ  $\beta$  ใหม่ จากสูตร

$$V_i(k+1) = wV_i(k) + c_1r_1P_i - x_i(k) + c_2r_2Pg - x_i(k),$$

$$X_i(k+1) = X_i(k) + V_i(k+1)$$

5. ปรับปรุงค่าฟีโรโมนจากสมการที่ (2.6-2.8)

6. การปรับปรุงคำตอบด้วยอัลกอริทึม 3-Opt

(2.10) ในการศึกษาครั้งนี้ กำหนดจำนวนฟีโรโมนภายในเส้นทางทั้งหมดคำนวณได้ดังสมการ

$$\text{Amount of pheromone} = \frac{1}{\text{number of ant} * \text{number of city}} \quad (2.10)$$

จากผลการทดลองค่าพารามิเตอร์ของ  $\alpha$  และ  $\beta$  อยู่ในช่วง  $0 \leq \alpha \leq \beta \leq 2$  ทุก ๆ particle มี 2 มิติ เป็น  $\alpha$  และ  $\beta$  ในอัลกอริทึม PSO พารามิเตอร์  $c_1$  และ  $c_2$  เลือกเป็น 2 ค่าพารามิเตอร์ อัลกอริทึม ACO และ PSO ดำเนินการ 1,000 รอบ ทดสอบแต่ละปัญหาซ้ำ 20 ครั้งอัตราการระเหยของฟีโรโมนเท่ากับ 0.1 ของอัตราการระเหยที่กำหนดในเส้นทาง ประสิทธิภาพของวิธีการนำเสนอโดยการประเมินผลการทำงาน 20 ครั้ง และคำนวณค่าเฉลี่ย, ค่าเบี่ยงเบนมาตรฐาน และเปอร์เซ็นต์ค่าความผิดพลาดสอดคล้องกับคำตอบที่ดีที่สุดได้ดังสมการที่ (2.11)

$$\text{Error}(\%) = \frac{(\text{average solution} - \text{best known solution})}{\text{best known solution}} \times 100 \quad (2.11)$$

จากการศึกษาในครั้งนี้ บนพื้นฐานของอัลกอริทึม PSO, ACO และ 3-Opt เป็นการแก้ปัญหาการเดินทางของพนักงานขาย (TSP) ใน 10 ปัญหามาตรฐานจากวรรณกรรม (Literature) และเปรียบเทียบบางอัลกอริทึม ผลการทดลองแสดงประสิทธิภาพของวิธีการนำเสนอโดยใช้จำนวนมดมากกว่าจำนวนลูกค้าเล็กน้อย สามารถแก้ปัญหา TSPs ได้ดีกว่าเมื่อเปรียบเทียบกับอัลกอริทึมอื่น ๆ ในด้านคุณภาพและความแข็งแกร่งของคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem [8]

งานวิจัยนี้ได้นำเสนออัลกอริทึมแบบผสมที่ใช้ symbiotic organisms search optimization algorithm (SOS) ซึ่งเป็นอัลกอริทึมที่มาจากการศึกษาถึงการอยู่ร่วมกันของสิ่งมีชีวิต ทำงานร่วมกับ Simulated annealing (SA) เพื่อใช้ในการแก้ปัญหาการเดินทางของพนักงานขาย

### 2.5.2.1 SOS algorithm

อัลกอริทึมเอสโอเอส ได้พัฒนาจากการศึกษาถึงการอยู่ร่วมกันของสิ่งมีชีวิต ซึ่งสิ่งมีชีวิต 2 สิ่ง มีลักษณะการอยู่ร่วมกัน 3 ลักษณะด้วยกัน คือ 1) การพึ่งพาอาศัยกัน (Mutualism) คือ สิ่งมีชีวิตทั้ง 2 สิ่งได้รับผลประโยชน์ทั้งคู่ 2) การอิงอาศัย (Commensalism) คือ มีสิ่งมีชีวิตที่ได้รับประโยชน์เพียงฝ่ายเดียว 3) การเป็นกาฝาก (Parasitism) คือ สิ่งมีชีวิตฝ่ายหนึ่งได้รับประโยชน์ ในขณะที่อีกฝ่ายเสียผลประโยชน์

การอยู่ร่วมกัน 3 แบบ มีลักษณะดังต่อไปนี้

#### 1) Mutualism

จะทำการปรับปรุงตำแหน่งของ  $X_i$  และ  $X_j$  และคำนวณค่าความเหมาะสมของตำแหน่งใหม่ ซึ่งหากค่าความเหมาะสมของ  $X_i$  และ  $X_j$  ที่ปรับปรุงใหม่ดีขึ้นกว่าเดิม ก็ให้  $X_i$  และ  $X_j$  ย้ายไปยังตำแหน่งใหม่

$$X'_i = X_i + rand(0,1) * (X_{best} - Mutual_{vec} * K_1) \quad (2.12)$$

$$X'_j = X_j + rand(0,1) * (X_{best} - Mutual_{vec} * K_2) \quad (2.13)$$

$$Mutual_{vect} = (X_i + X_j) / 2 \quad (2.14)$$

#### 2) Commensalism

จะทำการปรับปรุงตำแหน่งของ  $X_i$  และคำนวณค่าความเหมาะสมของตำแหน่งใหม่ ซึ่งหากค่าความเหมาะสมของ  $X_i$  ที่ปรับปรุงใหม่ดีขึ้นกว่าเดิม ก็ให้  $X_i$  ย้ายไปยังตำแหน่งใหม่

$$X'_i = X_i + rand(-1,1) * (X_{best} - X_j) \quad (2.15)$$

#### 3) Parasitism มีขั้นตอนคือ

จะทำการสร้าง Parasit\_vector จาก  $X_i$  และทำการสุ่มสลับค่าใน Parasit\_vector จากนั้นจะทำการหาค่าความเหมาะสมของ Parasit\_vector ซึ่งหากค่าความเหมาะสมของ Parasit\_vector ดีกว่า  $X_j$  ให้นำ Parasit\_vector ไปเก็บแทนที่  $X_j$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.2.2 Simulated annealing algorithm

การจำลองการอบเหนียว หรือ เอสเอ เป็นอีกอัลกอริทึมหนึ่งที่มีใช้ในการหาค่าความเหมาะสมเชิงการจัด โดยลักษณะการทำงานของเอสเอมีแนวคิดมาจากกระบวนการอบเหนียวซึ่งเป็นขั้นตอนในการหลอมโลหะ โดยจะค่อย ๆ ลดอุณหภูมิในการหลอมโลหะลงเรื่อย ๆ เพื่อให้ได้โลหะที่ดีคือเหนียวและไม่เปราะ

พลังงานของโลหะจะคำนวณจากฟังก์ชันวัตถุประสงค์ โดยเมื่อเวลาผ่านไปอุณหภูมิในการหลอมลดลง พลังงานของโลหะก็จะลดลง ค่าความเหมาะสมที่คำนวณได้จากผลรวมของระยะทางก็ควรจะลดลงด้วยซึ่งหมายถึงได้เส้นทางใหม่เป็นเส้นทางที่ดีขึ้นนั่นเอง แต่ในบางครั้งหากค่าความเหมาะสมที่คำนวณได้มีค่าเพิ่มขึ้นเพียงเล็กน้อย หากยอมรับเส้นทางนั้นก็อาจมีโอกาสดูพบเส้นทางที่ดีที่สุดได้เช่นกัน

ขั้นตอนการทำงานของ SA

#### 1. การกำหนดค่าเริ่มต้น

ทำการกำหนดค่าเริ่มต้นต่าง ๆ ในการทำงาน เช่น จำนวนประชากร, การสุ่มสร้างประชากรเริ่มต้น ( $X_i$ ), ค่าเริ่มต้นของอุณหภูมิ, อัตราการลดลงของอุณหภูมิ ( $\alpha$ )

#### 2. สร้างประชากรชุดใหม่ ( $X'_i$ )

โดยทำการสุ่มค่าเลขจำนวนเต็ม 2 ค่าและทำการสลับตำแหน่งของเมืองใน 2 ตำแหน่งนั้น

#### 3. คำนวณ $\Delta f$ เพื่อพิจารณาการปรับเปลี่ยนเส้นทาง

$$1) \Delta f = \frac{f(X'_i) - f(X_i)}{f(X_i)} \quad (2.16)$$

โดย  $f(X_i)$  คือ พลังงานของโลหะซึ่งก็คือฟังก์ชันวัตถุประสงค์

#### 2) ทำการเปรียบเทียบค่า $\Delta f$

หากค่า  $\Delta f$  มีค่าน้อยกว่า 0 คือเส้นทางใหม่เป็นเส้นทางที่มีระยะสั้นลง ดังนั้นจึงยอมรับเส้นทางใหม่ คือ กำหนดให้  $X_i = X'_i$

หาก  $\Delta f$  มีค่ามากกว่าหรือเท่ากับ 0 แสดงว่าเส้นทางใหม่เป็นเส้นทางที่แย่ลง แต่ในบางครั้งอาจค้นพบเส้นทางที่ดีที่สุดจากเส้นทางที่แย่ลงเพียงเล็กน้อยนี้ได้ ดังนั้นจึงทำการคำนวณค่าความน่าจะเป็นในการเลือกเส้นทางดังกล่าว (2.17) และนำค่าความน่าจะเป็นนั้นเปรียบเทียบกับค่าสุ่ม โดยหากค่าที่สุ่มน้อยกว่าค่าความน่าจะเป็นก็จะยอมรับเส้นทางใหม่ที่แย่ลงเพียงเล็กน้อยนั้น โดยกำหนดให้  $X_i = X'_i$

$$P(\Delta f, T_k) = \begin{cases} e^{\left(\frac{-\Delta f}{T_k}\right)}, & \Delta f > 0 \\ 1, & \Delta f \leq 0 \end{cases} \quad (2.17)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการลดความอุณหภูมิสำหรับการทำงานรอบถัดไป ดังสมการ

$$T_{k+1} = \alpha T_k \quad (2.18)$$

5. กำหนดประชากรมีค่าความเหมาะสมที่ดีที่สุด

ทำการเปรียบเทียบประชากรชุดใหม่กับประชากรมีค่าความเหมาะสมที่ดีที่สุด โดยหากประชากรชุดใหม่มีประชากรใดที่มีค่าความเหมาะสมน้อยกว่า  $X_{best}$  ก็ให้นำเส้นทางนั้นมา กำหนดเป็น  $X_{best}$  แทน

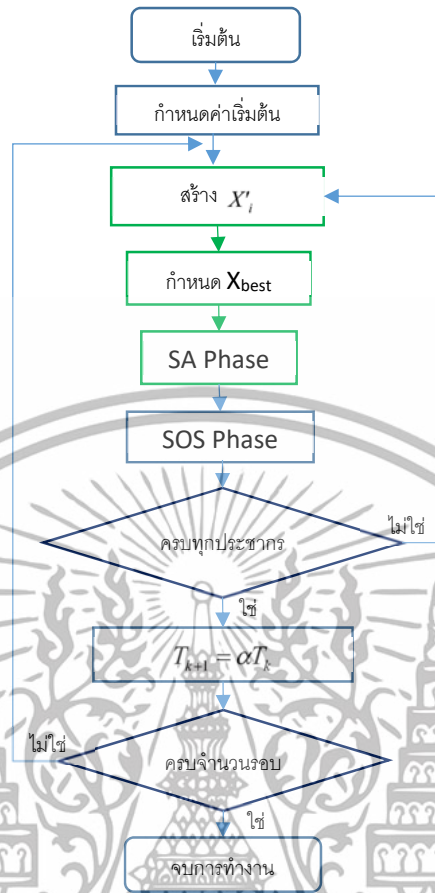
6. ตรวจสอบเงื่อนไขในการจบการทำงาน

หากยังไม่ครบจำนวนรอบที่กำหนดให้วนลูปทำซ้ำการทำงานตั้งแต่ขั้นตอนที่ 2 จนครบตามจำนวนรอบที่กำหนด

### 2.5.2.3 SOS-SA

เนื่องจากการทำงานของ SOS Algorithm นั้นพบว่า โอกาสเกิด local optimum ยังความเป็นไปได้อย่างมาก ดังนั้น SOS-SA จึงนำ SA เข้ามาช่วยเพื่อหลีกเลี่ยงโดยการเกิด local optimum โดยเมื่อทำการสร้างประชากรใหม่และกำหนด  $X_{best}$  แล้วจะนำ SA Phase เข้ามาเพื่อช่วยในการตัดสินใจระหว่างเก็บเส้นทางเดิมไว้หรือเลือกเส้นทางของประชากรใหม่แทน จากนั้นจะเข้าสู่กระบวนการทำงานของ SOS Phase และวนลูปการทำงานจนครบทุกประชากร เมื่อทำงานครบทุกประชากรแล้วจะทำการลดอุณหภูมิในการทำงานและวนลูปการทำงานให้ครบตามจำนวนรอบที่กำหนด ดังภาพที่ 2.2

งานวิจัยนี้ผลลัพธ์ที่ได้เป็นที่น่าพอใจ ในอนาคตผู้วิจัยได้มีแนวคิดในการทดสอบการทำงานแบบกระจายและการทำงานแบบคู่ขนาน (Parallel and distribution) โดยใช้ชุดข้อมูลที่มีขนาดใหญ่ขึ้น (Big data) ในการทดสอบ



ภาพที่ 2.2 แสดงขั้นตอนการทำงานของ SOS-SA

### 2.5.3 An improved ant colony optimization for vehicle routing problem [9]

งานวิจัยนี้ได้นำเสนอการหาค่าที่เหมาะสมในการจัดการโลจิสติกส์เพื่อแก้ปัญหาการจัดเส้นทางรถขนส่ง มีวัตถุประสงค์เพื่อปรับปรุงวิธีอาณานิคมมด (Ant Colony Optimization: ACO) ซึ่งนำเอากลยุทธ์การปรับปรุงการเพิ่มขึ้นของฟีโรโมน เรียกว่า กลยุทธ์ ant-weight และการดำเนินการกลายพันธุ์ (Mutation) ชื่ออัลกอริทึมว่า Improved Ant Colony Optimization (IACO) โดยในงานวิจัยนี้มีข้อกำหนดของปัญหา คือ จุดกระจายสินค้าเดียว มีรถหลายคัน แต่ละคันลูกค้าหลายคนไม่ซ้ำในคันเดียวกันและคันอื่น และแต่ละคันเริ่มต้นและสิ้นสุดจุดเดียวกัน รวมทั้งแต่ละคันไม่เกินความจุยานพาหนะ

ขั้นตอนการทำงานของอัลกอริทึม IACO มีดังนี้

1. กำหนดระยะทางระหว่างลูกค้า และค่าฟีโรโมนของทุกเส้นทาง
2. สร้างประชากรมดและลูกค้าเริ่มต้น โดยการสุ่ม ซึ่งลูกค้าคนแรกของเส้นทางของมดแต่ละตัวจะต้องไม่ซ้ำกัน ถัดมาในการที่มดจะเลือกเดินจากลูกค้า  $i$  ไปยังลูกค้า  $j$  มดจะพิจารณาจากความน่าจะเป็นว่าเส้นทางใดที่มีค่าฟีโรโมนมากจะเป็นเส้นทางที่มีโอกาสถูกเลือกมากที่สุด แสดงได้ดังสมการ (2.19)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_{ij}(k) = \begin{cases} \frac{\tau_{ij}^\alpha \times \eta_{ij}^\beta}{\sum_{h \notin \text{tabu}_k} \tau_{ih}^\alpha \times \eta_{ih}^\beta} & j \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

เมื่อ

$p_{ij}(k)$  คือ ความน่าจะเป็นมดตัวที่  $k$  เลือกเดินจากลูกค้า  $i$  ไปยังลูกค้า  $j$  บนเส้นทางนั้น

$\tau_{ij}$  คือ ความหนาแน่นของฟีโรโมนของเส้นทาง  $(i, j)$

$\eta_{ij}$  คือ ค่าการมองเห็นของเส้นทาง  $(i, j)$

$\alpha$  คือ อิทธิพลของฟีโรโมนบนเส้นทาง

$\beta$  คือ อิทธิพลของค่าการมองเห็น

$\text{tabu}_k$  คือ รายการของลูกค้าที่ห้ามเดินทางไปของมดตัวที่  $k$

3. การดำเนินการกลายพันธุ์ (Mutation operation) เป็นการปรับเปลี่ยนแต่ละลูกค้าตามความน่าจะเป็นที่กำหนดไว้ โดยมีขั้นตอนการกลายพันธุ์ดังนี้

3.1 กำหนดสมการความน่าจะเป็นของการกลายพันธุ์แต่ละรุ่นแสดงได้ดังสมการ (2.20)

$$p_m(t) = p_m^{\text{min}} + (p_m^{\text{max}} - p_m^{\text{min}})^{1-t/T} \quad (2.20)$$

เมื่อ

$p_m^{\text{min}}$  คือ อัตราการกลายพันธุ์ต่ำสุดสำหรับเริ่มต้น

$p_m^{\text{max}}$  คือ อัตราการกลายพันธุ์สูงสุดสำหรับสิ้นสุด

$T$  คือ จำนวนรอบสูงสุด

$t$  คือ รอบปัจจุบัน

โดยกำหนดให้อัตราการกลายพันธุ์ต่ำสุดเป็น  $p_m^{\text{min}} = 1/n_c$  และอัตราการกลายพันธุ์สูงสุดเป็น  $p_m^{\text{max}} = 1/n_v$

เมื่อ

$n_c$  คือ จำนวนของลูกค้า

$n_v$  คือ จำนวนของยานพาหนะ

3.2 สุ่มเลือกยานพาหนะ 2 คันจากตัวรถ ถัดไปสุ่มเลือกจุดกลายพันธุ์แต่ละคัน ดำเนินการแลกเปลี่ยนลูกค้าระหว่างคัน

3.3 ใช้อัลกอริทึม 2-Opt ดำเนินการแลกเปลี่ยนลูกค้าคันเดียวกัน เพื่อปรับปรุงคุณภาพคำตอบที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การปรับปรุงเส้นทางฟีโรโมน ซึ่งฟีโรโมนแต่ละเส้นทางมีการระเหยในปริมาณที่เท่ากัน แสดงได้ดังสมการ (2.21)

$$\tau_{ij}^{new} = \rho \times \tau_{ij}^{old} + \sum_k^K \Delta \tau_{ij}^k \quad \rho \in (0,1) \quad (2.21)$$

เมื่อ

$\tau_{ij}^{new}$  คือ เป็นฟีโรโมนบนเส้นทาง  $(i, j)$  หลังปรับปรุง

$\tau_{ij}^{old}$  คือ เป็นฟีโรโมนบนเส้นทาง  $(i, j)$  ก่อนปรับปรุง

$\rho$  คือ เป็นค่าคงที่ควบคุมความเร็วของการระเหย

$k$  คือ เป็นตัวชี้ที่  $k$

$K$  คือ เป็นจำนวนตัวชี้ทั้งหมด และ  $K > 0$

$\Delta \tau_{ij}^k$  คือ เป็นฟีโรโมนที่เพิ่มขึ้นบนตัวชี้ที่  $k$

4.1 กฎการปรับปรุงการเพิ่มขึ้นของฟีโรโมนโดยใช้กลยุทธ์ ant-weight แสดงได้ดังสมการ (2.22)

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{K \times L} \times \frac{D^k - d_{ij}}{m^k \times D^k} & \text{if link } (i,j) \text{ on the } k\text{th route} \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

เมื่อ

$Q$  คือ เป็นค่าคงที่

$L$  คือ เป็นระยะทางรวมทั้งหมด กล่าวคือ  $L = \sum_k D^k$

$D^k$  คือ เป็นระยะทางของยานพาหนะคันที่  $k$

$d_{ij}$  คือ เป็นระยะทางของ  $(i, j)$

$m^k$  คือ จำนวนของลูกค้านายพาหนะคันที่  $k$  และ  $m^k > 0$

อย่างไรก็ตาม เพื่อป้องกันไม่ให้เกิดค่าความเหมาะสมเฉพาะถิ่น (local optimization) และ การเพิ่มขึ้นของความน่าจะเป็นที่ได้รับคุณภาพคำตอบที่สูงขึ้น จำเป็นต้องจำกัดค่าต่ำสุดและสูงสุดของฟีโรโมนแสดงได้ดังสมการ (2.23) และ (2.24)

$$\tau_{\min} = \frac{Q}{\sum_i 2d_{oi}} \quad (2.23)$$

$$\tau_{\max} = \frac{Q}{\sum_i d_{oi}} \quad (2.24)$$

เมื่อ  $d_{oi}$  คือ ระยะทางจากจุดกระจายสินค้าไปยังลูกค้า  $i$ th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากการทดลองพบว่า อัลกอริทึม IACO เพิ่มประสิทธิภาพของฟีโรโมน และ ปรับอัตราการผลิตพันธุกรรมทั้งส่งเสริมการทำงานให้ลู่เข้าหาคำตอบได้รวดเร็ว ในอนาคตงานวิจัยนี้ ขยายไปสู่ปัญหาที่มีข้อจำกัดด้านเวลา หรือเพิ่มจุดกระจายสินค้าที่มากขึ้น

#### 2.5.4 An ant colony optimization model The period vehicle routing problem with time windows [10]

งานวิจัยนี้ นำเสนอการเพิ่มประสิทธิภาพของการหาค่าความเหมาะสมในวิธีอาณานิคมมด (Ant Colony Optimization: ACO) กับระยะเวลาที่วางแผนด้วยกรอบเวลาจำกัดในปัญหาการจัดเส้นทาง การขนส่ง โดยเมทริกซ์ Multi-dimension pheromone ใช้แก้ปัญหาข้อมูลที่เพิ่มขึ้นบนความต่างของแต่ละวัน และดำเนินการแลกเปลี่ยนลูกค่าแบบ Two-crossover วัตถุประสงค์เพื่อลดระยะทางทุกคันที่อยู่ในตัวรับในช่วงเวลาที่วางแผนไว้ โดยงานวิจัยนี้มีข้อกำหนดของปัญหาคือ จุดกระจายสินค้าเดียว วางแผนช่วงเวลาให้บริการ แต่ละช่วงเวลามีรถหลายคัน แต่ละคันลูกค่าหลายคน กำหนดความถี่ในการส่งสินค้า ได้แก่ ส่งทุกวัน, 2 วันส่งครั้ง และ 3 วันส่งครั้ง ขนาดและประเภทรถเหมือนกัน ไม่เกินความจุยานพาหนะ และส่งไม่ซ้ำลูกค่าแต่ละวัน

ขั้นตอนการทำงานของอัลกอริทึม

1. สร้างคำตอบโดยการเลือกลูกค่าบนพื้นฐานกฎความน่าจะเป็น ได้ดังสมการ (2.25)

$$P_{ij}^h = \begin{cases} \frac{(\tau_{(i,j)}^h)^\alpha \times (\eta_{(i,j)})^\beta}{\sum_{l \in \text{tabu}} (\tau_{(i,l)}^h)^\alpha \times (\eta_{(i,l)})^\beta} & j \notin \text{tabu} \\ 0 & \text{otherwise} \end{cases} \quad (2.25)$$

เมื่อ

$P_{(i,j)}^h$  คือ ความน่าจะเป็นในการเลือกลูกค่าจากเมือง  $i$  ไป  $j$  ของตัวรับใน  $h$  วัน

$\tau_{(i,j)}^h$  คือ ความเข้มข้นของฟีโรโมนขอบ  $(i, j)$

$\eta_{(i,j)}$  คือ ค่าการมองเห็นขอบ  $(i, j)$  โดย  $\eta_{(i,j)} = \frac{1}{d_{ij}}$

$d_{ij}$  คือ ระยะทางระหว่างลูกค่าจากเมือง  $i$  ไปเมือง  $j$

$\alpha$  คือ อิทธิพลของฟีโรโมนบนเส้นทาง

$\beta$  คือ อิทธิพลของค่าการมองเห็น

tabu คือ รายการของลูกค่าที่ห้ามเดินทางไป

2. การดำเนินการปรับเปลี่ยนการโครอสโอเวอร์ 2 ตัวรับในวันเดียวกัน แต่ละวันต้องมีรถอย่างน้อย 2 คัน คือ ต้องมี 2 เส้นทาง มีการดำเนินการ Two-Crossover ได้แก่ one-point crossover operation และ two-point crossover operation มีขั้นตอน ได้แก่ step 0 กำหนดค่าเริ่มต้น, step 1 กำหนดวัน, step 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกเส้นทาง, step 3 เลือกจุดครอส, step 4 ดำเนินการครอส และ step 5 ปรับปรุงคำตอบด้วยอัลกอริทึม 2-Opt

3. ปรับปรุงค่าฟีโรโมน เพื่อให้คำตอบในอนาคตดีขึ้น ได้ดังสมการ (2.26-2.29)

$$\tau_{(i,j)}^{h, new} = (1-\rho) \times \tau_{(i,j)}^{h, old} + \Delta\tau_{(i,j)}^h \quad \rho \in (0,1) \quad (2.26)$$

เมื่อ  $\tau_{(i,j)}^{h, new}$  คือ ฟีโรโมนบนเส้นทาง  $(i, j)$  หลังปรับปรุง ใน  $h$  วัน

$\tau_{(i,j)}^{h, old}$  คือ ฟีโรโมนบนเส้นทาง  $(i, j)$  ก่อนปรับปรุง ใน  $h$  วัน

$\rho$  คือ ค่าคงที่ควบคุมความเร็วของการระเหย

$\Delta\tau_{(i,j)}^h$  คือ ฟีโรโมนที่เพิ่มขึ้นบนเส้นทาง  $(i, j)$  ใน  $h$  วัน

$$\Delta\tau_{(i,j)}^h = \left( \lambda \times \frac{L^{opt}}{L^{current}} \times \delta \right) \quad (2.27)$$

เมื่อ  $L^{current}$  คือ ความยาวรวมของทัวร์ทั้งหมดในช่วงเวลาที่กำหนด

$L^{opt}$  คือ ความยาวรวมของทัวร์ทั้งหมดที่ดีที่สุด ณ ตอนนั้น

$\lambda$  คือ ค่าคงที่

$\delta$  คือ ค่าสัมประสิทธิ์การลงโทษ ซึ่งค่าสัมประสิทธิ์การลงโทษสำหรับคำตอบที่เป็นไปไม่ได้ ได้ดังสมการ (2.28)

$$\delta = \frac{1}{\sum_h \varphi^h + 1} \quad (2.28)$$

เมื่อ  $\varphi^h$  คือ ยานพาหนะที่เกินกำหนดใน  $h$  วัน

$$\varphi^h = \begin{cases} J^h - K & \text{if } J^h > K \\ 0 & \text{otherwise} \end{cases} \quad (2.29)$$

เมื่อ  $J^h$  คือ จำนวนของยานพาหนะของ  $h$  วัน ในคำตอบ

$K$  คือ จำนวนของยานพาหนะที่กำหนดใน  $h$  วัน

นอกจากนี้หลักการเลือกค่าความเหมาะสมเส้นทางย่อย (sub-optimization) และเพิ่มเติมความน่าจะเป็นที่ได้รับคำตอบที่มีคุณภาพสูงกว่า ปรับปรุงข้อจำกัดค่าสูงสุดและต่ำสุดในการศึกษาครั้งนี้ ได้ดังสมการ (2.30-2.31)

$$\tau_{\min} = 1 \quad (2.30)$$

$$\tau_{\max} = N \quad (2.31)$$

เมื่อ  $N$  คือ จำนวนรอบสูงสุดในการปรับปรุงคุณภาพคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปรงานวิจัยนี้ ฟีโรโมนข้อมูลหลายมิติ (Multi-dimension pheromone) เป็นกลยุทธ์ที่มีประสิทธิภาพในการลดการแก้ปัญหาที่เป็นไปไม่ได้ในระหว่างการค้นหา นอกจากนี้การดำเนินการ Two-crossover มีประสิทธิภาพดีกว่า การดำเนินการ One-crossover ในคุณภาพการค้นหา ในขณะที่การดำเนินการ Two-crossover ต้องใช้เวลาในการคำนวณมากขึ้น อัลกอริทึมนี้เหมาะกับข้อมูลจำนวนไม่มาก แต่มีการใช้งานที่เป็นไปได้มากในปัญหาสถานการณ์จริง ซึ่งงานวิจัยนี้ยังมีวัตถุประสงค์เพื่อกำหนดแนวทาง และคำตอบของสถานการณ์ที่จะช่วยกระตุ้นความสนใจของนักวิจัยอื่น ๆ

2.5.5 Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution [11]

งานวิจัยนี้นำเสนอการแก้ปัญหาการจัดเส้นทางรถ (VRP) ด้วย K-nearest neighbor algorithm (KNN) ในการกำหนดลำดับของจุดรับส่งเพื่อให้บริการลูกค้า ซึ่งจะใช้ nearest neighbor rule ในการเลือกจุดรับส่งถัดไปที่มีระยะทางสั้นที่สุดจากจุดรับส่งที่เหลือทั้งหมด โดยในงานวิจัยนี้มีข้อกำหนดของปัญหาคือ มียานพาหนะมากกว่า 1 คัน ในแต่ละวันยานพาหนะจะเริ่มเดินทางออกจากท่ารถในเวลาและเส้นทางที่แตกต่างกันและจะวิ่งย้อนกลับมายังท่าปล่อยรถเดิม โดยจะให้บริการในแต่ละจุดรับส่งเพียง 1 ครั้งในหนึ่งเส้นทางเท่านั้นและยานพาหนะนั้นจะต้องมีความจุเพียงพอในการให้บริการลูกค้าทุกคนในเส้นทาง

KNN มีขั้นตอนในการทำงาน คือ

#### 1. การสร้างโซลูชันเริ่มต้น

ในการสร้างโซลูชันเริ่มต้นจะกำหนดให้ท่าปล่อยรถเป็นจุดเริ่มต้นของเส้นทางย่อย จากนั้นกำหนดจำนวนจุดรับส่งในแต่ละเส้นทางย่อย เมื่อระบุจำนวนจุดในการรับส่งของแต่ละเส้นทางย่อยแล้วจะทำการสุ่มจุดรับส่งจุดแรกของแต่ละเส้นทางย่อย สำหรับจุดรับส่งต่อ ๆ ไปใช้ nearest neighbor rule ก็คือจะเลือกจุดรับส่งที่มีระยะทางสั้นที่สุดจากจุดรับส่งที่เหลือ ตัวอย่างโซลูชันที่มี 2 เส้นทางย่อย เช่น  $w_1 : 1, 5, 4$   $w_2 : 3, 2, 7, 6$  ซึ่งมี 3 และ 4 จุดรับส่งตามลำดับ

#### 2. คำนวณระยะทาง ( $d_s$ ) และความต้องการรวม ( $q_s$ ) ทั้งหมดของเส้นทาง ดังสมการ

$$d_s = d_{Route1} + d_{Route2} + \dots + d_{RouteN}$$

$$q_s = q_{Route1} + q_{Route2} + \dots + q_{RouteN}$$

$$d_{Routei} = d_{0,1} + d_{1,2} + \dots + d_{n-1,n}$$

$$q_{Routei} = q_1 + q_2 + \dots + q_n$$

#### 3. คำนวณค่าความเหมาะสมของโซลูชัน

$$Fitness_s = d_s + q_s$$

#### 4. นำโซลูชันที่สร้างมากำหนดเป็นโซลูชันที่ดีที่สุด

#### 5. วนลูปทำซ้ำจนครบตามเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสร้างโซลูชันใหม่ด้วย nearest neighbor rule จากนั้นให้ทำการคำนวณค่าความเหมาะสมของโซลูชันใหม่ จากนั้นพิจารณาค่าความเหมาะสมโดยหาค่าความเหมาะสมของโซลูชันใหม่ดีกว่าโซลูชันที่ดีที่สุดให้กำหนดโซลูชันใหม่เป็นโซลูชันที่ดีที่สุด

#### 6. ตรวจสอบความถูกต้องของโซลูชัน

หากเป็นไปตามข้อกำหนดของปัญหาจึงแสดงเป็นเส้นทางที่ดีที่สุด

งานวิจัยนี้ใช้ข้อมูลขนาดเล็กและมีจุดให้บริการเพียง 8 จุด ดังนั้นผลการค้นหาโซลูชันที่ดีที่สุดจึงไม่ใช่เรื่องยาก และยังใช้ประสิทธิภาพของ KNN ได้ไม่เต็มที่ อีกทั้งระยะห่างระหว่างแต่ละจุดก็ไม่มากส่งผลให้สัดส่วนที่ลดลงของระยะทางในโซลูชันที่หาได้มีเพียงเล็กน้อย

2.5.6 A discrete firefly algorithm to solve a rich vehicle routing problem modeling a newspaper distribution system with recycling policy [12]

งานวิจัยนี้ได้ศึกษาปัญหาในการจัดส่งหนังสือพิมพ์และเก็บคืนหนังสือพิมพ์ที่ขายไม่หมดในแต่ละวันของบริษัทแห่งหนึ่งใน Spain โดยมีการจัดกลุ่มลูกค้าที่จะจัดส่งแยกตามเมือง ซึ่งนอกจากการพิจารณาถึงความต้องการในการรับส่งหนังสือพิมพ์ของลูกค้าแล้ว ยังต้องพิจารณาการจราจรในช่วงเวลาของการขนส่งนั้น ๆ ซึ่งจะส่งผลต่อค่าใช้จ่ายโดยรวมด้วยเช่นกัน

งานวิจัยนี้นำเสนอการจัดการกับ rich vehicle routing problem โดยใช้ Discrete firefly algorithm ซึ่งมีลักษณะของปัญหาคือ 1) ระยะทาง/ค่าใช้จ่ายในการเดินทางจากเมือง ก ไป ข อาจแตกต่างจากการเดินทางจากเมือง ข ไป ก. 2) มีการแบ่งกลุ่มของลูกค้าที่จะให้บริการโดยพิจารณาตามเมืองที่ลูกค้าอาศัยอยู่ ซึ่งการเลือกยานพาหนะที่จะให้บริการแต่ละกลุ่ม จะต้องให้บริการได้ครบตามความต้องการลูกค้าครบทุกราย 3) ลูกค้าสามารถใช้บริการได้ทั้งลักษณะการนำส่งหนังสือพิมพ์ (delivery nodes) เพียงอย่างเดียว หรือทั้งนำส่งและเก็บคืนหนังสือพิมพ์ที่ขายไม่หมดในแต่ละวัน (collect nodes) ก็ได้ 4) ค่าใช้จ่ายที่เกิดขึ้นในการเดินทางไปให้บริการลูกค้าแต่ละรายอาจแตกต่างกันไป ซึ่งอาจเกิดจากหลายปัจจัย เช่น เป็นช่วงเวลาเร่งด่วน, สภาพการจราจร หรือสภาพดินฟ้าอากาศ เป็นต้น 5) อาจมีบางเส้นทางที่ไม่สามารถเดินทางผ่านได้ เช่น การกำหนดถนนบางสายเป็นถนนคนเดิน บางเส้นทางอาจกำหนดให้เดินรถทางเดียวในบางช่วงเวลา เป็นต้น

#### 2.5.6.1 Firefly algorithm

อัลกอริทึมหิ่งห้อย เป็นอัลกอริทึมหนึ่งที่น่าสนใจในการแก้ปัญหาการหาค่าที่เหมาะสมที่สุดสำหรับค่าต่อเนื่อง แต่การแก้ปัญหาการจัดเส้นทางรถขนส่งเป็นลักษณะของการหาค่าที่เหมาะสมที่สุดเชิงการจัดซึ่งเป็นค่าไม่ต่อเนื่อง ดังนั้น Discrete firefly algorithm (DFA) จึงมีการปรับเปลี่ยนขั้นตอนการทำงานบางอย่างเพื่อให้เหมาะสมกับการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หิ่งห้อยแต่ละตัวคือโซลูชันของปัญหาที่สนใจ ดังนั้นสำหรับปัญหาการจัดเส้นทาง การขนส่งหิ่งห้อยแต่ละตัวคือลำดับของเมืองที่เป็นเส้นทางในการขนส่งที่เป็นไปได้ เช่น  $x_1 : \{3,2,5,4,1\}$  คือเส้นทางกรให้บริการลูกค้าเมืองที่ 3, 2, 5, 4 และ 1 ตามลำดับ

ในการเคลื่อนที่ของหิ่งห้อยนั้นมีสิ่งที่ต้องพิจารณาคือ ความสนใจซึ่งคือความสว่างของหิ่งห้อยและระยะห่างระหว่างหิ่งห้อยแต่ละตัว แสงสว่างของหิ่งห้อยจะลดน้อยลงไปตามระยะห่างที่เพิ่มมากขึ้น นั่นคือหากหิ่งห้อยมีแสงสว่างมากแต่อยู่ไกลแสงที่ส่องมาถึงจะน้อยลงซึ่งอาจสว่างน้อยกว่าหิ่งห้อยที่มีแสงสว่างน้อยแต่อยู่ใกล้กว่า

ค่าฟังก์ชันเป้าหมาย (objective function) คือความสว่างของหิ่งห้อย ดังนั้นสำหรับปัญหาการจัดเส้นทางกรขนส่งนี้ฟังก์ชันเป้าหมายคือการคำนวณค่าใช้จ่ายโดยรวมในการขนส่ง ซึ่งหิ่งห้อยที่มีความน่าสนใจสูงที่สุดคือหิ่งห้อยที่มีค่าฟังก์ชันเป้าหมายต่ำสุดนั่นเอง

ลำดับขั้นตอนการทำงานของอัลกอริทึมหิ่งห้อย มีดังนี้

1. สุ่มสร้างเส้นทางเดินของหิ่งห้อยแต่ละตัว
2. กำหนดค่าเริ่มต้นของ  $\gamma = 0.95$
3. คำนวณค่าความสว่างสำหรับหิ่งห้อยแต่ละตัว ดังสมการ  
ค่าความสว่าง  $I_i = \text{fitness}(x_i)$
4. ทำซ้ำตั้งแต่ขั้นตอนที่ 5 ถึง 7 จนครบจำนวนรอบที่กำหนด
5. ทำซ้ำหิ่งห้อยแต่ละตัว ( $x_i$ )
6. ทำซ้ำสำหรับหิ่งห้อยตัวที่เหลือ ( $x_i$ )
  - 1) กรณี ค่าความสว่าง  $I_i < I_j$  ให้  
ระยะห่างระหว่างหิ่งห้อย  $r_{ij} = \text{HammingDistance}(x_i, x_j)$   
ทำการสุ่มค่า  $n$  ซึ่งมีค่าระหว่าง  $(2, r_{ij} * \gamma^s)$   
ปรับเส้นทางเดินของ  $x_j$  โดยเรียกใช้  $\text{InsertionFunction}(x_j, n)$
  - 2) ปรับปรุงค่าความสว่างของเส้นทางเดินใหม่ และประเมินเส้นทาง
7. ค้นหาหิ่งห้อยที่มีค่าความเหมาะสมที่ดีที่สุด (current best)

#### 2.5.6.2 HammingDistance

$\text{HammingDistance}(x_i, x_j)$  คือการหาระยะห่างระหว่างหิ่งห้อย 2 ตัว คือ  $x_i$  และ  $x_j$  ซึ่งจะพิจารณาจากจำนวนของลูกค้ำที่มีลำดับการเดินที่ต่างกันของหิ่งห้อยทั้ง 2 ตัว เช่น  $x_1 : \{3,2,5,4,1\}$  และ  $x_2 : \{3,5,2,4,1\}$  จะมีค่า  $r_{ij} = 2$  เป็นต้น

#### 2.5.6.3 InsertionFunction

$\text{InsertionFunction}(x_i, n)$  คือลักษณะการเคลื่อนที่ของหิ่งห้อย  $x_i$  โดยในการทำงานของฟังก์ชันนี้จะทำการสุ่มตำแหน่งเมืองของเส้นทางที่ต้องการปรับเปลี่ยนและสุ่มตำแหน่งที่จะนำค่าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดซึ่งจะอยู่ภายในกลุ่ม (cluster) เดียวกัน เช่น สมมติกลุ่ม  $n$  ได้ตำแหน่งที่ 3 และกลุ่มตำแหน่งที่จะโหนดได้ตำแหน่งที่ 5 ดังนั้นก่อนปรับตำแหน่ง  $x_1 = \{3,2,5,4,1\}$  และหลังปรับตำแหน่ง  $x_1 = \{3,2,4,5,1\}$

#### 2.5.6.4 การคำนวณค่าใช้จ่ายในการเดินทาง (travel costs)

ในการคำนวณค่าใช้จ่ายในการเดินทางนั้นแบ่งการคำนวณออกเป็น 2 ช่วงเวลา คือ ช่วงเวลาปกติและช่วงเวลาเร่งด่วน โดยมีค่าพารามิเตอร์เพิ่มเติมในการคำนวณ เช่น

$$\text{ช่วงเวลาปกติ } d_{ij} = \text{EuclideanDistance}(i,j) * 1.2$$

$$\text{ช่วงเวลาเร่งด่วน } d_{ij} = (\text{EuclideanDistance}(i,j) * 1.2) * 1.2$$

งานวิจัยนี้เป็นการปัญหาในการบริหารจัดการการส่งหนังสือพิมพ์และการเก็บคืนหนังสือพิมพ์ที่คงเหลือในแต่ละวัน โดยมีการจัดกลุ่มลูกค้าในการให้บริการและพิจารณาถึงความต้องการในการใช้บริการของลูกค้าควบคู่ไปกับความจุของยานพาหนะที่ให้บริการ นอกจากนั้นในการจัดเส้นทางยังมีปัจจัยที่ต้องพิจารณาเช่น ช่วงเวลาในการเดินทางและเส้นทางบางเส้นทางที่ไม่อนุญาตให้ยานพาหนะเดินทางผ่าน โดยใช้ Discrete firefly algorithm (DFA) ในการแก้ปัญหา ในอนาคตมีแนวคิดที่จะศึกษาถึงตัวแปรอื่น ๆ ให้ครอบคลุมการทำงานจริงมากขึ้น และพยายามลดระยะเวลาในการประมวลผลลงให้มากที่สุด

#### 2.5.7 Particle swarm optimization for the vehicle routing problem with stochastic demands [13]

งานวิจัยนี้ แนะนำอัลกอริทึมระหว่าง PSO กับ Local search (2-opt และ 3-opt) และกลยุทธ์ Path relinking แก้ปัญหาการจัดการเส้นทางที่ขนส่งกับความต้องการของลูกค้าที่แปรปรวนค่าที่ไม่ทราบค่าที่แน่นอน ( Vehicle Routing Problem with Stochastic Demands: VRPSD) ซึ่งยานพาหนะมีความจุจำกัดออกจากจุดกระจายสินค้าด้วยโหนดที่เต็ม เริ่มต้นจากจุดกระจายสินค้าเยี่ยมชมลูกค้าเพียงครั้งเดียวและกลับจุดกระจายสินค้า เรียกว่า a priori tour โดยมีวัตถุประสงค์เพื่อนำสมการของ PSO ที่มีความแตกต่างมาทดสอบกับจำนวนลูกค้าขนาดใหญ่ที่ไม่สามารถแก้ปัญหาในเวลาที่เหมาะสมได้ ซึ่ง VRPSD จะตัดสินใจที่จะกลับไปยังจุดกระจายสินค้าสำหรับการเติมสินค้าใหม่ หรือไปยังลูกค้าคนถัดไป

ในงานวิจัยนี้มีจำนวนตัวแปรที่แตกต่างของ PSO ใช้เพื่อค้นหาตัวแปรที่ทำงานได้ดีกว่าสำหรับปัญหา VRPSD ซึ่งการคำนวณความเร็วของตัวแปรเหล่านั้นจะถูกนำเสนอและวิเคราะห์ ดังสมการ (2.32-2.45)

##### PSO1: Classic Particle Swarm Optimization

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \text{rand}_1 (pbest_{ij} - x_{ij}(t)) + c_2 \text{rand}_2 (gbest_j - x_{ij}(t)) \quad (2.32)$$

##### PSO2: Inertia Particle Swarm Optimization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v_{ij}(t+1) = wv_{ij}(t) + c_1rand_1(pbest_{ij} - x_{ij}(t)) + c_2rand_2(gbest_j - x_{ij}(t)) \quad (2.33)$$

PSO3: Constriction Particle Swarm Optimization 1

$$v_{ij}(t+1) = \chi(v_{ij}(t) + c_1rand_1(pbest_{ij} - x_{ij}(t)) + c_2rand_2(gbest_j - x_{ij}(t))) \quad (2.34)$$

เมื่อ

$$\chi = \frac{2}{2 - c - \sqrt{c^2 - 4c}} \text{ and } c = c_1 + c_2, c > 4 \quad (2.35)$$

PSO4: Constriction Particle Swarm Optimization 2

เหมือนสมการที่ (2.35) แต่ต่างที่ปัจจัยที่ทำให้เล็กลง (Constriction factor) ดังสมการที่ (2.36)

$$\chi = \frac{2k}{2 - c - \sqrt{c^2 - 4c}} \text{ and } c = c_1 + c_2, c > 4 \quad (2.36)$$

PSO5: Constriction Particle Swarm Optimization 3

เป็นตัวแปรที่ง่ายกว่าของสมการ (2.34)

$$v_{ij}(t+1) = \chi(v_{ij}(t) + c(p_{mj} - x_{ij}(t))) \quad (2.37)$$

เมื่อใช้ปัจจัยสมการ (2.35) และสมการ (2.38)

$$p_{mj} = \frac{c_1 pbest_{ij} + c_2 gbest_j}{c} \quad (2.38)$$

PSO6: Cognition-only Particle Swarm Optimization

$$v_{ij}(t+1) = v_{ij}(t) + c_1rand_1(pbest_{ij} - x_{ij}(t)) \quad (2.39)$$

PSO7: Social-only Particle Swarm Optimization

$$v_{ij}(t+1) = v_{ij}(t) + c_2rand_2(gbest_j - x_{ij}(t)) \quad (2.40)$$

PSO8: Local Neighborhood topology Particle Swarm Optimization

$$v_{ij}(t+1) = (v_{ij}(t) + c_1rand_1(pbest_{ij} - x_{ij}(t)) + c_2rand_2(lbest_{ij} - x_{ij}(t))) \quad (2.41)$$

เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$lbest_{ij} \in \{N_i | f(lbest_{ij}) = \min\{f(x_{ij})\}, \forall x \in N_i\} \quad (2.42)$$

$$N_i = \left\{ \begin{array}{l} pbest_{i-n_{N_i}}(t), pbest_{i-n_{N_i}+1}(t), \dots, pbest_{i-1}(t), pbest_i(t), \\ pbest_{i+1}(t), \dots, pbest_{i+n_{N_i}}(t) \end{array} \right\} \quad (2.43)$$

$$pbest_{4j}, pbest_{5j}, pbest_{6j}, \quad (2.44)$$

$$pbest_{3j}, pbest_{4j}, pbest_{5j}, pbest_{6j}, pbest_{7j}, \quad (2.45)$$

ขั้นตอนสร้างเส้นทางของพาร์ทิเคิล (particles) โดยการสุ่ม และความเร็วของพาร์ทิเคิล (particles) เริ่มต้นเป็นศูนย์ในทุกรูปแบบ ตำแหน่งของพาร์ทิเคิล (particles) เปลี่ยนแปลงดังสมการ (2.46)

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2.46)$$

เมื่อ  $t$  เป็นตัวนับซ้ำ ตำแหน่งที่ดี ( $pbest_{ij}$ ) ในฝูง (Swarms) ได้ดังสมการที่ (2.47)

$$pbest_{ij} = \begin{cases} x_{ij}(t+1), & \text{if } f(x_{ij}(t+1)) < f(x_{ij}(t)) \\ pbest_{ij}, & \text{otherwise} \end{cases} \quad (2.47)$$

ตำแหน่งที่เหมาะสมของฝูง (Swarms) ทั้งหมดใน VRPSD ณ เวลารอบใด ๆ เป็นดังสมการ (2.48)

$$\begin{aligned} gbest_j &\in \{pbest_{1j}, pbest_{2j}, \dots, pbest_{N_j} | f(gbest_j)\} \\ &= \min \{f(pbest_{1j}), f(pbest_{2j}), \dots, f(pbest_{N_j})\} \end{aligned} \quad (2.48)$$

กลยุทธ์ Local search จะใช้เพื่อปรับปรุงการหาคำตอบจาก อัลกอริทึม PSO ในแต่ละรอบของอัลกอริทึมคำตอบที่เหมาะสมของฝูง (Swarms) ทั้งหมด และจะเก็บคำตอบที่เหมาะสมของแต่ละพาร์ทิเคิล (particles) อัลกอริทึมจะหยุดเมื่อจำนวนรอบมากที่สุดที่กำหนด

ขั้นตอนการทำงานของอัลกอริทึมพาร์ทิเคิลสวอม (PSO) สำหรับแก้ปัญหา VRPSD มีดังนี้

1. เลือกสมการความเร็วของอัลกอริทึมพาร์ทิเคิลสวอม (PSO)
2. เลือก จำนวนฝูง (Swarms) จำนวนของพาร์ทิเคิล (particles) ในแต่ละ Swarms
3. กำหนดค่าเริ่มต้น โดยสุ่มเส้นทางเดินปรับเปลี่ยนตำแหน่งในรูปแบบต่อเนื่อง กำหนดตำแหน่ง และความเร็วแต่ละพาร์ทิเคิล (particles)
4. คำนวณค่าความเหมาะสม (Fitness) เริ่มต้นของแต่ละ particle ฟังก์ชันเป้าหมาย (Objective function) ของปัญหาซึ่งเป็นระยะทางที่คาดหวังของ a priori tour ได้ดังสมการ (2.49)

$$f_j(q) = \text{Minimum} \{f_j^p(q), f_j^r(q)\} \quad (2.49)$$

เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f_j^p(q) = d_{j,j+1} + \sum_{k \leq q} f_{j+1}(q-k)p_{j+1,k} + \sum_{k > q} [2d_j + 1,0 + f_{j+1}(q+Q-k)]p_{j+1,k} \quad (2.50)$$

$$f_j^r(q) = d_{j,0} + d_{0,j+1} + \sum_{k=1}^K f_{j+1}(Q-k)p_{j+1,k} \quad (2.51)$$

ด้วยเงื่อนไขขอบเขต

$$f_n(q) = d_{n,0}, q \in L_n \quad (2.52)$$

เมื่อ  $s = (0, 1, \dots, n)$  คือ a priori tour

$q$  คือ โหลด (load) ที่เหลือของยานพาหนะหลังเสร็จการบริการลูกค้าคนที่  $j$

$f_j(q)$  คือ ระยะทางที่คาดหวัง (expected cost) จากลูกค้าคนที่  $j$  ต่อไปข้างหน้า

(ดังนั้น  $j=0$  ถ้า ถัดมา  $f_0(q)$  ระยะทางที่คาดหวัง ของ a priori tour)

$f_j^p(q)$  คือ ระยะทางที่คาดหวังของเส้นทาง เมื่อยานพาหนะไม่กลับไปยังจุดกระจายสินค้าแต่ไปยังลูกค้าถัดไป

$f_j^r(q)$  คือ ระยะทางที่คาดหวังเมื่อยานพาหนะกลับไปยังจุดกระจายสินค้าเดิมของให้เต็มโหลดเพื่อป้องกันการใส่ใหม่

5. ค้นหาคำตอบที่ดีในแต่ละพาร์ติเคิล (particles) และค้นหาพาร์ติเคิล (particles) ที่ดีที่สุดของ Swarms
- การทำงานหลัก
6. คำนวณความเร็วแต่ละพาร์ติเคิล (particles) และตำแหน่งใหม่ของแต่ละพาร์ติเคิล (particles)
7. ปรับเปลี่ยนตำแหน่งในรูปแบบตัวเลข
8. คำนวณค่าความเหมาะสม (Fitness) ใหม่ในแต่ละพาร์ติเคิล (particles)
9. ปรับปรุงตำแหน่งด้วยอัลกอริทึม 2-Opt, 3-Opt และกลยุทธ์ Path Relinking และปรับปรุงคำตอบที่ดีแต่ละพาร์ติเคิล (particles)
10. ค้นหาพาร์ติเคิล (particles) ที่ดีของ Swarms ทั้งหมด
11. ปรับเปลี่ยนตำแหน่งในรูปแบบต่อเนื่อง
- จบการทำงาน คืนค่าพาร์ติเคิล (particles) ที่ดี

ผลการวิจัยนี้ อัลกอริทึมพาร์ติเคิลสวอม (PSO) มีสมการความเร็วที่จัดการได้ดีกว่าสำหรับปัญหา VRPSD คือ สมการ PSO3 รวมกับการค้นหาเฉพาะที่ (Local search) และกลยุทธ์ Path Relinking และถูกเลือกให้เปรียบเทียบกับอัลกอริทึมอื่น ในวรรณกรรม แม้ว่าอัลกอริทึมพาร์-

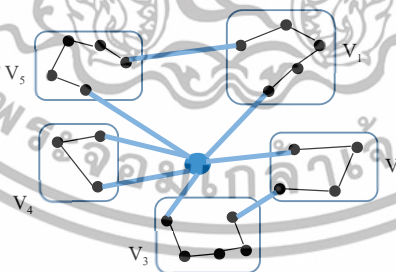
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เคล็ดสวอม (PSO) จะแก้ปัญหาสถานการณ์สำหรับหาค่าเหมาะสมที่ต่อเนื่อง แต่สามารถค้นหาประสิทธิภาพการเปลี่ยนแปลงของคำตอบของอัลกอริทึมพาร์ทิเคิล (PSO) ในค่าที่ไม่ต่อเนื่องโดยปราศจากการสูญเสียสารสนเทศ งานวิจัยในอนาคตมุ่งเป้าใน 2 ทิศทางที่แตกต่าง ทิศทางที่ 1 จะแก้ปัญหา Vehicle Routing Problems กับ ความต้องการของลูกค้าไม่แน่นอน (Stochastic demands) ใช้เทคนิคแรงบันดาลใจจากธรรมชาติชนิดอื่น ๆ เช่น Clonal Selection Algorithm, Honey Bees mating optimization เป็นต้น และทิศทางอื่น ๆ แก้ปัญหากับอัลกอริทึมที่นำเสนอปัญหาที่ซับซ้อนมากยิ่งขึ้นลักษณะ Vehicle Routing Problem with Stochastic Demands และ Time Windows หรือ Dynamic Vehicle Routing Problem

#### 2.5.8 A novel two-level optimization approach for clustered vehicle routing problem [14]

งานวิจัยนี้ได้นำเสนอการแก้ปัญหาการจัดเส้นทางรถที่เหมาะสมที่สุด ในลักษณะการจัดกลุ่มการขนส่ง clustered vehicle routing problem (CluVRP) ซึ่งลูกค้าทั้งหมดจะถูกแตกออกเป็นหลายกลุ่มตามที่กำหนดไว้ก่อนหน้า โดยมียานพาหนะในการให้บริการมากกว่า 1 คัน และยานพาหนะทุกคันจะเริ่มต้นการบริการและสิ้นสุดการให้บริการที่คลังสินค้า

โดยงานวิจัยนี้จะแตกปัญหาเป็น 2 ปัญหาย่อย คือ upper-level (global) subproblem และ lower-level (local) subproblem โดย upper-level หรือ global level จะพิจารณาเส้นทางรถที่เชื่อมระหว่างกลุ่มย่อย (cluster) ซึ่งเป็น global routes โดยใช้ Genetic Algorithm ในการจัดการกับปัญหา ในขณะที่ lower-level จะพิจารณาถึงเส้นทางรถที่ให้บริการลูกค้าภายในกลุ่ม ซึ่งในขั้นตอนนี้จะแปลง global routes ให้เป็นเส้นทางรถเดินทางของพนักงานขายเพื่อคำนวณหาค่า optimum ด้วยโปรแกรม Concorde TSP solver



ภาพที่ 2.3 แสดงกราฟแบบไม่มีทิศทางที่แสดงถึงการแบ่งกลุ่มของลูกค้า

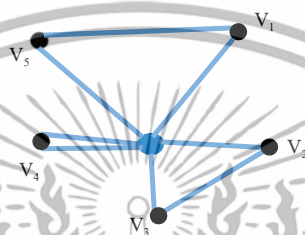
จากภาพ ถึงการแบ่งกลุ่มของลูกค้าใน CluVRP ซึ่งในภาพได้แบ่งลูกค้าจำนวน 22 ราย ออกเป็น 5 กลุ่มคือ  $V_1$  ถึง  $V_5$  โดยมี  $V_0$  เป็นคลังสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นเชื่อม (edge) ลูกค้าที่ใช้ในงานวิจัยแบ่งออกเป็น 2 ประเภท คือ intra-cluster edge เป็นเส้นที่เชื่อมโยงระหว่างลูกค้าที่อยู่ในกลุ่มเดียวกัน กับ inter-cluster edge ซึ่งเป็นเส้นที่เชื่อมโยงระหว่างลูกค้าต่างกลุ่ม

#### 2.5.8.1 The upper-level (global) subproblem

upper-level subproblem เป็นการพิจารณาถึงเส้นทางที่เชื่อมโยงระหว่างกลุ่ม (cluster) ซึ่งเป็น global routes โดยใช้ขั้นตอนวิธีเชิงพันธุกรรมในการค้นหาเส้นทางที่เหมาะสมที่สุด



ภาพที่ 2.4 แสดง global route ที่ได้จากตัวอย่างในภาพที่ 2.3

ในงานวิจัยนี้ได้นำขั้นตอนวิธีเชิงพันธุกรรมหรือ GA มาใช้เพื่อค้นหาเส้นทางเชื่อมระหว่างกลุ่มที่มีค่าความเหมาะสมที่สุด โดยมีโครโมโซมซึ่งใช้หมายเลขกำกับกลุ่มในการแสดงถึงโซลูชันของปัญหา โดยจากภาพที่ จะประกอบด้วย 3 เส้นทางย่อย ได้แก่  $V_0-V_5-V_1-V_0$ ,  $V_0-V_2-V_3-V_0$ ,  $V_0-V_4-V_0$  ซึ่งแสดงด้วยโครโมโซมคือ 51102304

GA ประกอบด้วยขั้นตอนการทำงานคือ

#### 1. กำหนดประชากรเริ่มต้น

GA จะทำการสุ่มค่าเพื่อสร้างประชากรเริ่มต้น โดยในงานวิจัยนี้ได้นำความต้องการของลูกค้าที่จะใช้บริการ และความจุของยานพาหนะที่จะให้บริการมาเป็นส่วนหนึ่งในการพิจารณาสร้างประชากรเริ่มต้น

#### 2. การเลือกประชากรที่จะนำมาทำ Crossover

จะใช้ rank based selection ซึ่งมีขั้นตอนดังนี้

1) ให้สร้าง list ของโซลูชัน โดยเรียงลำดับประชากรตามค่าความเหมาะสมโดยให้ค่าความเหมาะสมที่แย่ที่สุดอยู่เป็นลำดับแรก

2) กำหนด rank และ rank สะสม โดยพิจารณาจากตำแหน่งใน list ของโซลูชันที่เรียงลำดับไว้ โดย

$$\text{rank} = \text{position} / \text{ผลรวมของ position}$$

เช่น มีประชากร 3 โครโมโซม โครโมโซมในตำแหน่งแรกจะมีค่า =  $1 / 6$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$T = \text{rank}$  สะสมที่มากที่สุด

3) สุ่มค่า  $R$  ซึ่งมีค่าอยู่ระหว่าง 0 ถึง  $T$

4) หาก ( $\text{rank}$  สะสม  $\geq R$ ) และ ไม่ใช่โครโมโซมเดิม

ให้เลือกโครโมโซมนั้นเป็นประชากรที่จะนำมาทำ Crossover

### 3. การทำ Crossover

สำหรับงานวิจัยนี้ใช้ order crossover (หรือ OX) โดยเลือก 2 parent มาสร้าง 2 offspring ตามขั้นตอนดังนี้

1) สุ่มเลือกตำแหน่ง cut point ที่จะทำการ crossover จำนวน 2 ตำแหน่ง

2) ทำการสร้าง Offspring โดย

- Offspring<sub>1</sub> เลือก copy ช่วงตำแหน่ง cut point จาก Parent<sub>1</sub>

- ส่วนที่เหลือของ Offspring<sub>1</sub> จะนำข้อมูลจาก Parent<sub>2</sub> มาเติมเริ่มจากตำแหน่ง cut point ที่ 2 โดยเลือกเฉพาะค่าที่ไม่ซ้ำกับค่าที่มีอยู่ เมื่อครบถึงข้อมูลสุดท้ายของ Parent<sub>2</sub> วนกลับมาเติมที่จุดเริ่มต้นตำแหน่งแรก

- Offspring<sub>2</sub> จะทำขั้นตอนเช่นเดียวกับ Offspring<sub>1</sub> แต่จะ copy ช่วง cutpoint จาก Parent<sub>2</sub> แทน แล้วนำส่วนของ Parent<sub>1</sub> มาเติมให้ครบข้อมูล

### 4. การทำ Mutation

การทำ Mutation จะทำการสุ่ม 2 ตำแหน่งเพื่อทำการสลับค่า โดยในการทำ Mutation จะมีการกำหนดค่าความน่าจะเป็นในการ mutate เพื่อใช้ในการพิจารณาว่าจะเกิดการทำ Mutation หรือไม่ และเนื่องจากในงานวิจัยนี้เส้นทางที่จะทำการสลับเป็นส่วนหนึ่งของ global route ดังนั้นการจึงเป็นการทำ inter-cluster mutation

#### 2.5.8.2 The lower-level (local) subproblem

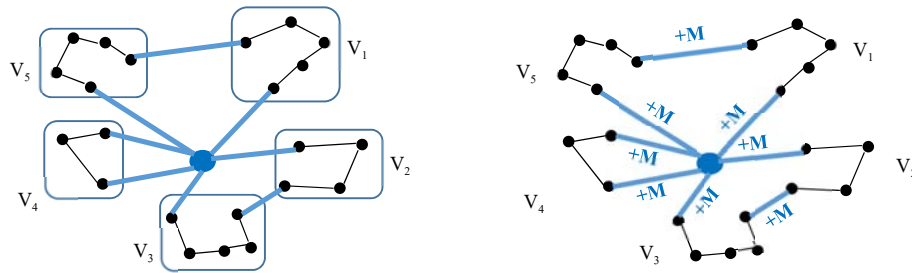
lower-level เป็นขั้นตอนการแปลงจาก global route ไปเป็นเส้นทางการเดินทางของพนักงานขาย เพื่อค้นหาเส้นทางที่เหมาะสมที่สุด โดยใช้ Concorde TSP solver โดยแนวคิดในการทำงานคือจะทำการเพิ่มค่า  $M$  เข้าไปในเส้นเชื่อมระหว่าง cluster ทุกเส้น ดังนี้

1) หากลูกค้า  $v_i$  และ  $v_j$  อยู่ภายในกลุ่มเดียวกัน ให้  $c'(v_i, v_j) = c(v_i, v_j)$

2) แต่หากลูกค้า  $v_i$  และ  $v_j$  อยู่ต่างกลุ่ม ให้ทำการเพิ่มค่า  $M$  ในการคำนวณ

โดยให้  $c'(v_i, v_j) = c(v_i, v_j) + M$

โดยที่  $M$  เป็นค่าที่มากกว่าผลรวมของ cost ทั้งหมด



ภาพที่ 2.5 แสดงการเพิ่มค่า M ในเส้นทาง TSP

งานวิจัยนี้ได้ผลลัพธ์ที่ดีขึ้นจำนวน 47 instance จากทั้งหมด 168 instance



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การแก้ปัญหาการจัดเส้นทางรถขนส่ง โดยประยุกต์ใช้อัลกอริทึมอาณานิคมมด

ในงานวิจัยนี้กล่าวถึง การปรับปรุงอัลกอริทึมอาณานิคมมด (AS) เพื่อแก้ปัญหาการจัดเส้นทางรถขนส่ง ลักษณะของ AS มดจะใช้เส้นทางถัดไปที่มีค่าความหนาแน่นมากที่สุด ซึ่งความหนาแน่นจะถูกปรับปรุงเมื่อมดเดินทางผ่านทุกเส้นทางของลูกค้าทั้งหมด โดยความหนาแน่นของฟีโรโมนขึ้นอยู่กับอัตราส่วนระหว่างค่าคงที่และความยาวของทัวร์ จุดบกพร่องของ AS มีดังต่อไปนี้ 1) ความหนาแน่นของประชากรมีไม่เพียงพอ 2) การลู่เข้าก่อนเวลาอันควร และติดกับดักบริเวณพื้นที่โลคอล และ 3) คุณภาพของคำตอบที่ได้รับแย่ลงเมื่อจำนวนของลูกค้าเพิ่มมากขึ้น เพื่อแก้ไขข้อบกพร่องของ AS ดังกล่าว ผู้วิจัยจึงนำเสนอวิธีแรก โดยเพิ่มวิธีการสร้างทัวร์ใหม่ ให้มดที่แข็งแรงมีโอกาสสร้างทัวร์ได้มากกว่ามดที่อ่อนแอ วิธีต่อมา การเพิ่มน้ำหนักให้กับทัวร์ โดยทัวร์ที่มีระยะทางสั้นจะมีน้ำหนักมากกว่าทัวร์อื่น ๆ เพื่อใช้ในการปรับปรุงฟีโรโมนทุกเส้นทาง และวิธีสุดท้ายปรับปรุงการค้นหาพื้นที่โลคอลโดยใช้วิธีการจำลองแบบอบอ่อน (Simulated Annealing: SA) เพื่อมีความเป็นไปได้ที่จะกระโดดออกจากพื้นที่โลคอล เมื่อจำนวนรอบการทำงานในการหาคำตอบเพิ่มมากขึ้นคำตอบที่ได้ในปัจจุบันจะค่อย ๆ ลู่เข้าหาค่าเหมาะสมที่สุด สามารถอธิบายขั้นตอนการทำงานของอัลกอริทึมที่นำเสนอได้ตามลำดับดังต่อไปนี้

#### 3.1 ขั้นตอนการกำหนดค่าเริ่มต้น

การกำหนดค่าเริ่มต้น ประกอบด้วย จำนวนทัวร์ ( $n$ ) จำนวนลูกค้า ( $m$ ) ฟีโรโมนเริ่มต้น ( $\tau_0$ ) อัตราการระเหย ( $\rho$ ) ค่าคงที่ ( $Q$ ) อุณหภูมิเริ่มต้น ( $t_{start}$ ) อุณหภูมิสิ้นสุด ( $t_{end}$ ) รอบการทำงานมากที่สุดของ SA ( $iter_{max}$ ) และ รอบการทำงานสูงสุดของอัลกอริทึม ( $maximum$ )

#### 3.2 ขั้นตอนการสร้างทัวร์เริ่มต้น

การสร้างทัวร์เริ่มต้นให้จำนวนทัวร์เท่ากับลูกค้า ทำได้โดยการสุ่มสร้างทัวร์ สามารถอธิบายได้ดังนี้

1) สุ่มเลือกลูกค้าให้แต่ละ Tour เริ่มจากทัวร์แรก ( $Tour_1$ ) จนกระทั่งถึงทัวร์สุดท้าย ( $Tour_n$ ) ในแต่ละทัวร์ที่เดินทางลูกค้าคนแรกไม่ซ้ำกันของแต่ละรอบการทำงาน เพื่อสร้างความหลากหลายให้กับคำตอบที่เป็นไปได้ ดังภาพที่ 3.1

Tour_1	2	5	1	6	8	9	7	4	3
Tour_2	7	5	9	3	1	4	2	6	8
Tour_3	4	3	6	8	5	1	7	2	9
Tour_4	6	1	9	8	5	3	2	4	7
Tour_n	9	5	2	3	4	8	1	6	7

ภาพที่ 3.1 ตัวอย่างของทัวร์เริ่มต้น

2) จำนวนความยาวของเส้นทางระหว่างเมือง  $i$  และเมือง  $j$  เป็น  $d_{ij}$  ซึ่ง  $d_{ij}$  เป็นระยะทาง Euclidean  $i$  และ  $j$  (นั่นคือ  $d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$ )

3) จำนวนความยาวแต่ละทัวร์จากจุดเริ่มต้นไปยังจุดสิ้นสุด และจากจุดสิ้นสุดมายังจุดเริ่มต้น จากสมการ

$$L_k = d_{0,1} + \sum_{i=1}^{n-1} d_{i,i+1} + d_n, d_0, \quad (3.1)$$

เมื่อ  $L_k$  เป็นความยาวทัวร์ของมดตัวที่  $k$ ,  $d_{0,1}$  เป็นระยะทางจากจุดกระจายสินค้าไปยังลูกค้าคนแรก,  $\sum_{i=1}^{n-1} d_{i,i+1}$  เป็นระยะทางรวมจากลูกค้าคนแรกไปยังลูกค้าคนสุดท้าย และ  $d_n, d_0$  เป็นระยะทางจากลูกค้าคนสุดท้ายไปยังจุดกระจายสินค้า ดังภาพที่ 3.2

	Tour									$L_k$
Tour_1	2	5	1	6	8	9	7	4	3	30981.89
Tour_2	7	5	9	3	1	4	2	6	8	32444.10
Tour_3	4	3	6	8	5	1	7	2	9	40426.25
Tour_4	6	1	9	8	5	3	2	4	7	25035.93
Tour_n	9	5	2	3	4	8	1	6	7	27677.83

ภาพที่ 3.2 ตัวอย่างความยาวของทัวร์เริ่มต้น (อ้างอิง ภาพที่ 3.1)

### 3.3 ขั้นตอนการปรับปรุงความหนาแน่นของฟีโรโมน

การปรับปรุงความหนาแน่นของฟีโรโมนทุกเส้นทางในแต่ละทัวร์ จากสมการ 3.2

$$\tau_{ij}^{New} = (1 - \rho) \cdot \tau_{ij}^{Old} + \sum_{k=1}^m W \cdot \Delta \tau_{ij}^k \quad (3.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $\tau_{ij}^{New}$  เป็นฟีโรโมนบนเส้นทาง  $(i, j)$  หลังปรับปรุง,  $\tau_{ij}^{Old}$  เป็นฟีโรโมนบนเส้นทาง  $(i, j)$  ก่อนปรับปรุง,  $\Delta\tau_{ij}^k$  ฟีโรโมนบนเส้นทาง  $(i, j)$  และ  $W$  เป็นค่าถ่วงน้ำหนักของแต่ละตัวรับบนเส้นทาง  $(i, j)$  จากสมการ 3.3

$$W = \log \left[ \frac{L_k - L_{shortest}}{L_{longest}} + 0.0001 \right] * -1 \quad (3.3)$$

เมื่อ  $L_{shortest}$  เป็นทัวร์สั้นสุดของรอบการทำงาน และ  $L_{longest}$  เป็นทัวร์ยาวสุดของรอบการทำงาน

### 3.4 ขั้นตอนการทำสำเนาทัวร์

การทำสำเนาทัวร์สามารถอธิบายได้ดังนี้

- 1) จัดเรียงทัวร์ให้สอดคล้องกับความยาวจากทัวร์ที่สั้นสุดไปยังทัวร์ยาวที่สุด ดังภาพที่ 3.3

Sorting the tour										$L_k$
Tour_4	6	1	9	8	5	3	2	4	7	25035.93
Tour_n	9	5	2	3	4	8	1	6	7	27677.83
Tour_1	2	5	1	6	8	9	7	4	3	30981.89
Tour_2	7	5	9	3	1	4	2	6	8	32444.10
Tour_3	4	3	6	8	5	1	7	2	9	40426.25

ภาพที่ 3.3 ตัวอย่างการจัดเรียงทัวร์เริ่มต้น (อ้างอิง ภาพที่ 3.2)

- 2) แบ่งทัวร์ออกเป็น 2 กลุ่ม โดยกลุ่มที่ 1 เป็นทัวร์ผ่านการจัดเรียง 20 % และกลุ่มที่ 2 เป็นส่วนที่เหลือ 80 % ซึ่งทัวร์ในแต่ละกลุ่มถูกสำเนาที่แตกต่างกัน โดยทัวร์กลุ่มที่ 1 ในแต่ละทัวร์ ถูกสำเนาเป็น 2 ทัวร์ ส่วนทัวร์ในกลุ่มที่ 2 ในแต่ละทัวร์ถูกสำเนาเพียง 1 ทัวร์ สามารถอธิบายการทำสำเนาทัวร์ได้ดังนี้

1. สุ่มเลือก 2 ตำแหน่งในทัวร์
2. แลกเปลี่ยนลูกศรระหว่าง 2 ตำแหน่งที่สุ่ม ส่วนลูกศรที่เหลือที่ไม่ถูกแลกเปลี่ยนมี

2 กรณี กรณีแรก ตำแหน่งแรกถูกสุ่มเลือกให้แลกเปลี่ยน โดยตำแหน่งที่อยู่ระหว่างตำแหน่งแรกที่สุ่มและหลังตำแหน่งที่สุ่ม ใช้สมการความน่าจะเป็นในการเลือกลูกศรคนถัดไปที่เป็นไปได้ จากสมการ 3.4 ส่วนกรณี ที่ 2 ตำแหน่งที่สุ่มไม่ใช่ตำแหน่งแรกให้นำตำแหน่งก่อนหน้าที่ยุ่มาเดินก่อน สำหรับตำแหน่งที่อยู่ระหว่างตำแหน่งที่สุ่ม และหลังตำแหน่งที่สุ่ม ใช้สมการความน่าจะเป็นในการเลือกลูกศรคนถัดไปที่เป็นไปได้ จากสมการ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{k \in \text{allowed}_k} (\tau_{ik})^\alpha \cdot (\eta_{ik})^\beta} & \text{if } j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

เมื่อ  $P_{ij}^k$  ความน่าจะเป็นของทัวร์ที่  $k$  เดินทางจากเมือง  $i$  ไปเมือง  $j$   $\alpha$  กับ  $\beta$  เป็นพารามิเตอร์ที่ควบคุมความสัมพันธ์ของฟีโรโมนที่ทิ้งไว้กับการมองเห็น และ  $\eta_{ij} = 1/d_{ij}$  เป็นส่วนผกผันของระยะทาง ทำไปเรื่อย ๆ จนกระทั่งสร้างทัวร์สมบูรณ์

3. ทัวร์ถัดไปสำเนาทัวร์เหมือนกับทัวร์แรกไปเรื่อย ๆ จนกระทั่งสร้างทัวร์ครบในแต่ละรอบการทำงาน

สามารถอธิบายตัวอย่างการสำเนาทัวร์ทั้ง 2 กรณีจากภาพที่ 3.4(1) ถึง ภาพที่ 3.5(5)

Tour_4	6	1	9	8	5	3	2	4	7
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.4(1) ตัวอย่างทัวร์กรณีที่ 1

Tour_4	6	1	9	8	5	3	2	4	7
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.4(2) การสุ่มเลือก 2 ตำแหน่งกรณีที่ 1

Tour_4	5				6				
--------	---	--	--	--	---	--	--	--	--

ภาพที่ 3.4(3) แลกเปลี่ยนลูกค้าในตำแหน่งที่สุ่มกรณีที่ 1

จากภาพที่ 3.4(3) ลูกค้าที่อยู่ระหว่างตำแหน่งแรกที่สุ่มกับตำแหน่งที่ 2 คือ 1, 9, 8, 3, 2, 4, และ 7 ต้องการเลือกลูกค้าคนถัดไป ยกตัวอย่างเช่น 5 เป็นลูกค้าคนแรก ลูกค้าคนถัดไปที่เป็นไปได้ได้แก่ 1, 9, 8, 3, 2, 4, และ 7 ใช้สมการ 3.4 จำนวน  $P_{5,1} = 10, P_{5,9} = 15, P_{5,8} = 12, P_{5,3} = 20, P_{5,2} = 25, P_{5,4} = 19$ , และ  $P_{5,7} = 22$  ดังนั้น ลูกค้าคนถัดไปที่เป็นไปได้ คือ 2 ต่อมาลูกค้าคนถัดไปที่เป็นไปได้ได้แก่ 1, 9, 8, 3, 4, และ 7 ใช้สมการ 3.4 จำนวน  $P_{2,1} = 14, P_{2,9} = 23, P_{2,8} = 15, P_{2,3} = 18, P_{2,4} = 21$ , และ  $P_{2,7} = 12$  ดังนั้น ลูกค้าคนถัดไปที่เป็นไปได้ คือ 9 ทำอย่างนี้ต่อไปเรื่อย ๆ จนกระทั่งสร้างทัวร์สมบูรณ์ ดังภาพที่ 3.4(4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tour_4	5	2	9	3	6	7	1	8	4
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.4(4) ผลลัพธ์การสำเนาทัวร์กรณีที่ 1

Tour_4	6	1	9	8	5	3	2	4	7
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.5(1) ตัวอย่างทัวร์กรณีที่ 2

Tour_4	6	1	9	8	5	3	2	4	7
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.5(2) การสุ่มเลือก 2 ตำแหน่งกรณีที่ 2

Tour_4				2			8		
--------	--	--	--	---	--	--	---	--	--

ภาพที่ 3.5(3) แลกเปลี่ยนลูกคำในตำแหน่งที่สุ่มกรณีที่ 2

จากภาพที่ 3.5(2) ตำแหน่งที่สุ่มไม่ใช่ตำแหน่งแรกให้นำตำแหน่งก่อนหน้าที่ยุ่มาเดินก่อนดังภาพที่ 3.5(4) สำหรับลูกคำที่อยู่ระหว่างตำแหน่งที่สุ่มและหลังตำแหน่งที่สุ่ม คือ 5, 3, 4, และ 7 เพื่อเลือกลูกคำคนถัดไป ยกตัวอย่างเช่น 2 เป็นลูกคำลำดับแรกก่อนใช้สมการ 3.4 ลูกคำคนถัดไปที่เป็นไปได้ ได้แก่ 5, 3, 4, และ 7 ใช้สมการ 3.4 คำนวณ  $P_{2,5} = 17$ ,  $P_{2,3} = 13$ ,  $P_{2,4} = 20$ , และ  $P_{2,7} = 14$  ดังนั้น ลูกคำคนถัดไปที่เป็นไปได้ คือ 4 ต่อมาลูกคำคนถัดไปที่เป็นไปได้ ได้แก่ 5, 3, และ 7 ใช้สมการ 3.4 คำนวณ  $P_{4,5} = 24$ ,  $P_{4,3} = 20$ , และ  $P_{4,7} = 23$  ดังนั้น ลูกคำคนถัดไปที่เป็นไปได้ คือ 5 ทำอย่างนี้ต่อไปเรื่อยๆ จนกระทั่งสร้างทัวร์สมบูรณ์ ดังภาพที่ 3.5(5)

Tour_4	6	1	9	2			8		
--------	---	---	---	---	--	--	---	--	--

ภาพที่ 3.5 (4) คัดลอกตำแหน่งก่อนหน้ามาเดินก่อนกรณีที่ 2

Tour_4	6	1	9	2	4	5	8	7	3
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.5 (5) ผลลัพธ์การสำเนาทัวร์กรณีที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 ขั้นตอนการปรับปรุงทัวร์

การปรับปรุงทัวร์สามารถอธิบายได้ดังนี้

- 1) ปรับปรุงความหนาแน่นของฟีโรโมนทุกเส้นทางในแต่ละทัวร์โดยใช้สมการ 3.2
- 2) จัดเรียงทัวร์ทั้งหมดจากความยาวสั้นสุดไปยังความยาวมากที่สุด
- 3) นำทัวร์ที่ผ่านการจัดเรียงเฉพาะจำนวนทัวร์เท่ากับจำนวนเริ่มต้นที่กำหนด
- 4) ปรับปรุงคุณภาพคำตอบเพื่อให้ได้ทัวร์ที่เหมาะสมที่สุดโดยใช้อัลกอริทึมแบบจำลองอบอ่อน

Simulated Annealing (SA) สามารถอธิบายขั้นตอนดังนี้

1. สำหรับแต่ละทัวร์ เลือกเพียง 1 วิธีจาก 3 วิธีเพื่อสร้างทัวร์ใหม่ วิธีการสร้างทัวร์ใหม่ 3 วิธี สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

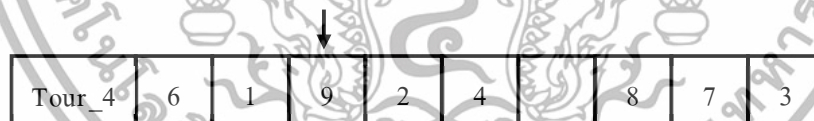
1.1 วิธี Insert mutation เป็นการย้ายตำแหน่งลูกค่าไปแทรกตำแหน่งอื่นของทัวร์ มีขั้นตอนดังต่อไปนี้

- 1) สุ่มตำแหน่งลูกค่าที่ต้องการลบ ดังภาพที่ 3.6



ภาพที่ 3.6 สุ่มตำแหน่งที่ 6 ของลูกค่าที่ต้องการลบ

- 2) สุ่มตำแหน่งลูกค่าที่ต้องการแทรก ดังภาพที่ 3.7

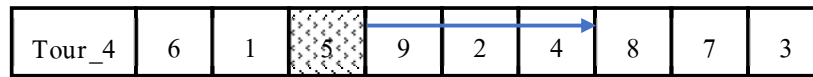


ภาพที่ 3.7 สุ่มตำแหน่งที่ 3 ของลูกค่าที่ต้องการแทรก

- 3) เช็ตำแหน่งของ ข้อ 1) และ 2) ห้ามซ้ำกัน ถ้าซ้ำสุ่มใหม่
- 4) นำลูกค่า ณ ตำแหน่งที่ลบมาแทรก ณ ตำแหน่งที่สุ่มแทรก โดยมี

เงื่อนไข 2 กรณี ดังนี้

กรณีที่ 1 ถ้าตำแหน่งที่แทรกอยู่ก่อนตำแหน่งที่ลบ ตำแหน่งที่มีอยู่แล้วเลื่อนไปทางด้านขวามือ จากภาพที่ 3.7 ตำแหน่งที่แทรก คือ ตำแหน่งที่ 3 อยู่ก่อนตำแหน่งที่ลบ คือ ตำแหน่งที่ 6 ในภาพที่ 3.6 เกิดผลลัพธ์ในกรณีที่ 1 ดังภาพที่ 3.8



ภาพที่ 3.8 ผลลัพธ์วิธี Insert mutation จากกรณีที่ 1

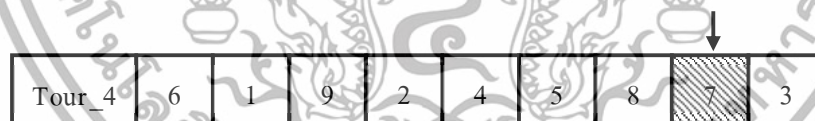
กรณีที่ 2 ถ้าตำแหน่งที่แทรกอยู่หลังตำแหน่งที่ลบให้เลื่อนตำแหน่งที่มีอยู่แล้วไปทางด้านซ้ายมือและแทรกลูกค่าตำแหน่งที่ลบ ณ ตำแหน่งดังกล่าว จากภาพที่ 3.6 ตำแหน่งที่ลบ คือ ตำแหน่งที่ 6 แต่ถ้าตำแหน่งที่แทรก อยู่ตำแหน่งที่ 8 จะเกิดผลลัพธ์ในกรณีที่ 2 ดังภาพที่ 3.9



ภาพที่ 3.9 ผลลัพธ์วิธี Insert mutation จากกรณีที่ 2

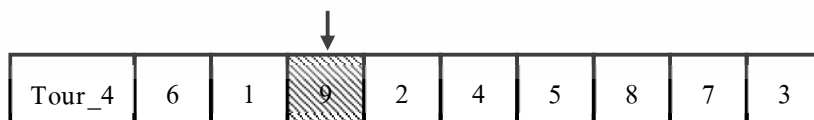
1.2 วิธี Inverse mutation เป็นการกลายพันธุ์แบบเปลี่ยนตำแหน่งทิศทาง (inversion) ของทัวร์เกิดขึ้นเมื่อมีรอยขาด 2 แห่งที่ส่วนใดส่วนหนึ่งของทัวร์ และส่วนของทัวร์นั้นต่อกลับเข้าไปในทัวร์เดิม แต่เปลี่ยนทิศกลับหัวกลับหางต่างไปจากสภาพทัวร์เดิม ดังนั้นการเปลี่ยนแปลงแบบนี้ยังมีจำนวนทัวร์เท่าเดิม แต่ตำแหน่งลูกค่า หรือ กลุ่มของลูกค่าในทัวร์นั้นเปลี่ยนแปลง มีขั้นตอนดังต่อไปนี้

1) สุ่มตำแหน่งที่ 1 ดังภาพที่ 3.10



ภาพที่ 3.10 สุ่มตำแหน่งที่ 1

2) สุ่มตำแหน่งที่ 2 ดังภาพที่ 3.11



ภาพที่ 3.11 สุ่มตำแหน่งที่ 2

3) กลุ่มลูกค่าที่ถูกสุ่มเลือกเพื่อสลับเปลี่ยนทิศทาง ดังภาพที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tour_4	6	1	9	2	4	5	8	7	3
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.12 กลุ่มลูกค้าที่ถูกเลือกเพื่อสลับ และ แทนที่

4) สลับเปลี่ยนทิศทางจากตำแหน่งที่ 1 ไปยัง ตำแหน่งที่ 2 โดยตำแหน่งสุดท้ายของกลุ่มลูกค้าจะสลับและแทนที่ตำแหน่งที่ 1 ของกลุ่มลูกค้าต่อมาตำแหน่งถัดจากตำแหน่งสุดท้ายจะสลับและแทนที่ตำแหน่งที่ 2 ของกลุ่มลูกค้าทำอย่างนี้ไปเรื่อย ๆ จนกระทั่ง สิ้นสุดจำนวนลูกค้าในกลุ่มที่ถูกเลือก ดังภาพที่ 3.13

Tour_4	6	1	7	8	5	4	2	9	3
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.13 ผลลัพธ์วิธี Inverse mutation

1.3 วิธี Exchange mutation เป็นการสลับแลกเปลี่ยน แต่ละคู่ในทัวร์ มีขั้นตอนดังต่อไปนี้

- 1) กำหนดจำนวนคู่ในทัวร์ที่จะแลกเปลี่ยน เช่น จำนวน 2 คู่
- 2) สุ่มตำแหน่งที่ 1 และ 2 ของแต่ละคู่ ถ้าตำแหน่งใดซ้ำกัน ให้สุ่มใหม่ ดังภาพที่ 3.14

Tour_4	6	1	9	2	4	5	8	7	3
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.14 แต่ละคู่ของตำแหน่งที่สุ่ม

- 3) สลับตำแหน่งระหว่างแต่ละคู่ทัวร์จนครบตามที่กำหนดไว้ ดังภาพที่ 3.15

Tour_4	1	6	9	2	4	7	8	5	3
--------	---	---	---	---	---	---	---	---	---

ภาพที่ 3.15 ผลลัพธ์วิธี Exchange mutation

2. คำนวณความยาวของทัวร์ใหม่
3. คำนวณความยาวที่เปลี่ยนไประหว่างทัวร์ใหม่และทัวร์ปัจจุบัน จากสมการ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta E = L^{new} - L^{current} \quad (3.5)$$

เมื่อ  $\Delta E$  ความยาวที่เปลี่ยนไป,  $L^{new}$  ความยาวทัวร์ใหม่,  $L^{current}$  ความยาวทัวร์ปัจจุบัน ถ้า  $\Delta E \geq 0$  คำนวณค่าความเป็นไปได้  $e^{\Delta E/T}$  และสุ่มสร้างเลขจำนวนจริงซึ่งอยู่ในช่วง  $[0, 1]$  ถ้าตัวเลขที่ถูกสุ่มน้อยกว่าหรือเท่ากับ ให้แทนที่ทัวร์ใหม่ อื่น ๆ แทนที่ทัวร์ปัจจุบัน ถ้า  $\Delta E \leq 0$  แทนที่ ทัวร์ใหม่ และ T เป็น อุณหภูมิในแต่ละรอบของ SA ซึ่งจะค่อย ๆ ลดลงเรื่อย ๆ ในแต่ละรอบ จากสมการ 3.6

$$T = t_{start} - \left( \frac{t_{start} - t_{end}}{iter_{max}} \right) * iter \quad (3.6)$$

เมื่อ  $t_{start}$  อุณหภูมิเริ่มต้น,  $t_{end}$  อุณหภูมิสิ้นสุด,  $iter$  รอบปัจจุบัน และ  $iter_{max}$  จำนวนรอบสูงสุด

4. เปรียบเทียบความยาวทัวร์ปัจจุบันกับทัวร์ใหม่เพื่อเก็บความยาวและทัวร์ที่สั้น
5. อนุญาตให้ทัวร์ทำงานในรอบของ SA ถัดไปเรื่อย ๆ จนกระทั่งสิ้นสุดเงื่อนไข

พิมพ์ความยาว และทัวร์เหมาะสมที่สุดในแต่ละรอบ

### 3.6 ขั้นตอนการพิจารณาหยุดการค้นหา

ทำซ้ำขั้นตอน 3.4 – 3.5 จนกระทั่งครบตามจำนวนรอบที่กำหนด

## บทที่ 4

### การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึง ชุดข้อมูลที่ใช้ในการทดลอง และ ผลการเปรียบเทียบประสิทธิภาพ อัลกอริทึมที่นำเสนอกับอัลกอริทึมพื้นฐานและอัลกอริทึมอื่น ๆ

#### 4.1 ชุดข้อมูลที่ใช้ในการทดลอง

งานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานจากฐานข้อมูลออนไลน์ “MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances” ซึ่งประกอบด้วย ชุดข้อมูลจำนวน 15 ชุด สามารถอธิบายชุดข้อมูลที่ใช้ในการทดลองดังตารางที่ 4.1 ได้ดังนี้

ตารางที่ 4.1 ชุดข้อมูลที่ใช้ในการทดลอง

NAME	DIMENSIONS	COMMENT
eil51	51	51-city problem (Christofides/Eilon)
berlin52	52	52 locations in Berlin (Groetschel)
st70	70	70-city problem (Smith/Thompson)
eil76	76	76-city problem (Christofides/Eilon)
rat99	99	Rattled grid (Pulleyblank)
kroA100	100	100-city problem A (Krolak/Felts/Nelson)
kroB100	100	100-city problem B (Krolak/Felts/Nelson)
kroC100	100	100-city problem C (Krolak/Felts/Nelson)
kroD100	100	100-city problem D (Krolak/Felts/Nelson)
kroE100	100	100-city problem E (Krolak/Felts/Nelson)
eil101	101	101-city problem (Christofides/Eilon)
pr107	107	107-city problem (Padberg/Rinaldi)
pr124	124	124-city problem (Padberg/Rinaldi)
pr136	136	136-city problem (Padberg/Rinaldi)
pr144	144	144-city problem (Padberg/Rinaldi)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงชุดข้อมูลที่ใช้ในการทดลอง ในการหาระยะทางระหว่างเมืองใช้วิธีการยูคลิเดียน (Euclidean distance) เช่น ชุดข้อมูล eil51 การคำนวณระยะทางระหว่างเมือง 1 และ 5 จากวิธีการยูคลิเดียน  $d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$  เมือง 1 พิกัด  $x_1 = 37, y_1 = 52$  และ เมือง 5 พิกัด  $x_5 = 40, y_5 = 30$  ได้ผลลัพธ์ 22.20 ดังภาพที่ 4.1

```

NAME : eil51
COMMENT : 51-city problem(Christofides/Eilon)
TYPE : TSP
DIMENSION : 51
EDGE_WEIGHT_TYPE : EUC_2D
NODE_COORD_SECTION
1 37 52
2 49 49
3 52 64
4 20 26
5 40 30

```

ภาพที่ 4.1 ตัวอย่างบางส่วนของชุดข้อมูล eil51

#### 4.2 ผลการทดลองเปรียบเทียบประสิทธิภาพ

เมื่อทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูล “MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances” พบว่าอัลกอริทึมที่นำเสนอสามารถค้นพบคำตอบใกล้เคียงกับคำตอบที่ทราบว่าเหมาะสมที่สุด ดังแสดงในตารางที่ 4.2 โดยคอลัมน์ Instance Name แสดงถึง ชุดข้อมูลที่ใช้ทดลอง BKS แสดงถึง คำตอบที่ทราบว่าเหมาะสมที่สุด โดยอัลกอริทึมที่นำมาเปรียบเทียบเป็นอัลกอริทึมพื้นฐานของการหาค่าความเหมาะสมในอาณานิคมมด ได้แก่ Ant System algorithm (AS) และ Elitist Ant System algorithm (EAS) ซึ่งทั้ง 2 อัลกอริทึมผู้วิจัยเขียน โปรแกรมขึ้นมาเปรียบเทียบกับอัลกอริทึมที่นำเสนอ

ตารางที่ 4.2 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมพื้นฐาน

Instance Name	BKS	Algorithm		
		AS	EAS	Proposed
eil51	426	457.39	473.12	428.87
berlin52	7542	8093.4	8093.4	7544.4
st70	675	761.69	727.13	677.11
eil76	538	578.33	586.98	546.5
rat99	1211	1369.5	1369.5	1220.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ) ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมพื้นฐาน

Instance Name	BKS	Algorithm		
		AS	EAS	Proposed
kroA100	21282	24698	24698	21285
kroB100	22141	25824	25824	22197
kroC100	20749	23566	23248	20751
kroD100	21294	23903	23903	21479
kroE100	22068	24638	24371	22137
eil101	629	722.56	727.76	640.21
pr107	44303	47160	47348	44302
pr124	59030	67015	66893	59031
pr136	96772	111970	110890	97664
pr144	58537	60953	60779	58535

จากตารางที่ 4.2 แสดงคำตอบในแต่ละอัลกอริทึม พบว่า อัลกอริทึมที่นำเสนอค้นพบคำตอบใกล้เคียงกับคำตอบที่ทราบว่าเหมาะสมที่สุดทุกชุดข้อมูลที่ใช้ทดลอง ส่วน อัลกอริทึม AS และ EAS คู่เข้าสู่คำตอบโลกดรอพดิมม้อย่างรวดเร็ว คำตอบที่ได้รับไม่ใช่คำตอบที่ทราบว่าเหมาะสมที่สุดทุกชุดข้อมูลที่ใช้ทดลอง ซึ่งพบว่า อัลกอริทึม AS และ EAS คำตอบเท่ากัน 5 ชุดข้อมูล ได้แก่ berlin52, rat99, kroA100, kroB100 และ kroD100 ในขณะที่อัลกอริทึม EAS ค้นหาคำตอบได้ดีกว่า อัลกอริทึม AS 6 ชุดข้อมูล ได้แก่ st70, kroC100, kroE100, pr124, pr136 และ pr144 และอัลกอริทึม AS ค้นหาคำตอบได้ดีกว่าอัลกอริทึม EAS 4 ชุดข้อมูล ได้แก่ eil151, eil176, eil101 และ pr107

ตารางที่ 4.3 แสดงการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่นำเสนอกับอัลกอริทึมของงานวิจัยอื่นที่ทดลองกับชุดข้อมูลเดียวกัน โดยคอลัมน์ Instance Name แสดงถึง ชุดข้อมูลที่นำมาทดสอบ BKS แสดงถึง คำตอบที่ทราบว่าเหมาะสมที่สุด Best แสดงถึง คำตอบของชุดข้อมูล Average แสดงถึง ค่าเฉลี่ยหรือค่ากลางของชุดข้อมูล PDBest (%) แสดงถึง เปอร์เซ็นต์เบี่ยงเบนของค่า Best เมื่อเทียบกับค่า BKS สูตรคำนวณ  $PDBest(\%) = ((Best-BKS)/BKS)*100$

ในงานวิจัยนี้ได้เปรียบเทียบอัลกอริทึมที่นำเสนอกับงานวิจัยอื่น ๆ จำนวน 2 งาน คือ

- 1) งานวิจัยชื่อ Discrete symbiotic organisms search algorithm for travelling salesman Problem หรือ BasicSOS
- 2) งานวิจัยชื่อ Improving variable neighborhood search to solve the traveling salesman problem หรือ VNS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการเปรียบเทียบอัลกอริทึมที่นำเสนอกับอัลกอริทึมอื่น ๆ

Instance Name	BKS	BasicSOS [2017]			VNS [2018]			Proposed		
		Best	Average	PDBest (%)	Best	Average	PDBest (%)	Best	Average	PDBest (%)
eil51	426	429	438.73	0.70	428.98	428.98	0.690	428.87	432.367	0.674
berlin52	7542	7647	7659.49	1.39	7544.36	7544.36	0.031	7544.4	7564.95	0.032
st70	675	675	699.13	0.00	677.11	677.11	0.313	677.11	682.724	0.313
eil76	538	542	556.31	0.74	545.39	552.57	1.374	546.5	554.384	1.580
rat99	1211	1284	1297.38	6.03	1240.38	1241.26	2.420	1220.4	1231.71	0.776
kroA100	21282	21401	21633.8	0.56	21618.2	21695.79	1.570	21285	21624.4	0.014
kroB100	22141	22155	23142.8	0.07	22139.07	22140.2	-0.009	22197	22431.5	0.253
kroC100	20749	20811	21020.2	0.30	20750.76	20809.29	0.000	20751	20889.7	0.010
kroD100	21294	21492	22044.3	0.93	21294.29	21490.62	0.001	21479	21773.1	0.869
kroE100	22068	22128	22467.1	0.27	22174.6	22193.8	0.483	22137	22427.9	0.313
eil101	629	649	659.71	3.18	642.31	648.27	2.110	640.21	652.26	1.782
pr107	44303	46097	46112.2	4.05	44301.68	44314.92	-0.003	44302	44634.5	-0.002
pr124	59030	68942	69211.1	16.79	59030.74	59051.82	0.001	59030.74	59629.3	0.001
pr136	96772	98018	100461	1.29	97979.11	97985.84	1.240	97664	100895.2	0.922
pr144	58537	58587	60136.9	0.09	58535.22	58563.97	-0.003	58535	58798.9	-0.003

จากตารางที่ 4.3 พบว่า อัลกอริทึมที่นำเสนอดีกว่าอัลกอริทึมอื่นที่นำมาเปรียบเทียบใน 6 ชุดข้อมูล ได้แก่ eil51, rat99, kroA100, eil101, pr107 และ pr136 ค่าตอบที่พบคือ 428.87, 1220.4, 21285, 640.21, 44302 และ 97664 อัลกอริทึม VNS ดีกว่าใน 5 ชุดข้อมูล ได้แก่ berlin52, kroB100, kroC100, kroD100 และ pr144 ค่าตอบที่พบคือ 7544.36, 22139.07, 20750.76, 21294.29 และ 58535.22 ถัดมาอัลกอริทึม BasicSOS ดีกว่าใน 3 ชุดข้อมูล ได้แก่ st70, eil76 และ kroE100 ค่าตอบที่พบคือ 675, 542 และ 22128 ส่วนอัลกอริทึมที่นำเสนอ กับ VNS ค่าตอบเท่ากันใน 2 ชุดข้อมูล ได้แก่ st70 และ pr124 ค่าตอบที่พบคือ 677.11 และ 59030.74

ค่าเฉลี่ยหรือค่ากลางของชุดข้อมูล ทั้งหมด 15 ชุดข้อมูล อัลกอริทึมที่นำเสนอมีประสิทธิภาพดีกว่าใน 2 ชุดข้อมูล ได้แก่ rat99 และ kroA100 และอัลกอริทึม VNS มีประสิทธิภาพดีกว่าใน 13 ชุดข้อมูล ได้แก่ eil51, berlin52, st70, eil76, kroB100, kroC100, kroD100, kroE100, eil101, pr107, pr124, pr136 และ pr144 ส่วนอัลกอริทึม BasicSOS มีประสิทธิภาพที่ไม่เหนือกว่าอัลกอริทึมที่นำเสนอ และ อัลกอริทึม VNS

สำหรับเปอร์เซ็นต์เบี่ยงเบนของค่า Best เมื่อเทียบกับค่า BKS พบว่า อัลกอริทึมที่นำเสนอดีกว่าอัลกอริทึมอื่นที่นำมาเปรียบเทียบใน 6 ชุดข้อมูล ได้แก่ eil51, rat99, kroA100, eil101, pr107 และ pr136 PDBest(%) ที่พบคือ 0.674, 0.776, 0.014, 1.782, -0.002 และ 0.922 ถัดมาอัลกอริทึม VNS ดีกว่าอัลกอริทึมอื่นที่นำมาเปรียบเทียบใน 5 ชุดข้อมูล ได้แก่ berlin52, kroB100, kroC100, kroD100 และ pr144 PDBest(%) ที่พบคือ 0.031, -0.009, 0.000, 0.001 และ -0.003 ส่วนอัลกอริทึม BasicSOS ดีกว่าอัลกอริทึมอื่นที่นำมาเปรียบเทียบใน 3 ชุดข้อมูล ได้แก่ st70, eil76 และ kroE100 PDBest(%) ที่พบคือ 0.00, 0.74 และ 0.27

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

#### 5.1 สรุปผลการทดลอง

งานวิจัยนี้นำเสนอการปรับปรุงประสิทธิภาพของอัลกอริทึมระบบมด (Ant system: AS) โดยนำเสนอวิธีในการเพิ่มความหลากหลายให้กับอัลกอริทึม AS โดยแบ่งคำตอบออกเป็น 2 กลุ่ม คือ คำตอบที่ดี สร้างประชากรตัวเป็น 2 เท่า และคำตอบที่แย่ สร้างประชากรตัวเป็น 1 เท่า ต่อมาเพิ่มน้ำหนักให้กับแต่ละตัว ซึ่งขึ้นอยู่กับความยาวของแต่ละตัวเพื่อนำมาคำนวณการปรับปรุงฟีโรโมนแต่ละเส้นทางในแต่ละตัว และปรับปรุงคำตอบให้เหมาะสมโดยใช้อัลกอริทึม Simulated annealing

ในงานวิจัยนี้ได้ทำการทดสอบอัลกอริทึมที่นำเสนอกับชุดข้อมูลมาตรฐานจากฐานข้อมูลออนไลน์ “MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances” จำนวน 15 ชุดข้อมูล ซึ่งผลการวัดประสิทธิภาพของอัลกอริทึมที่นำเสนอกับชุดข้อมูลที่นำมาทดสอบพบว่า อัลกอริทึมที่นำเสนอสามารถพบคำตอบที่ใกล้เคียงคำตอบที่ทราบว่าเหมาะสมที่สุด โดยเมื่อนำเปอร์เซ็นต์เบี่ยงเบนคำตอบที่ดีเมื่อเทียบกับคำตอบที่ทราบว่าเหมาะสมของงานวิจัยนี้ไปเปรียบเทียบกับเปอร์เซ็นต์เบี่ยงเบนคำตอบที่ดีเมื่อเทียบกับคำตอบที่ทราบว่าเหมาะสมของงานวิจัยอื่น ๆ พบว่า งานวิจัยที่นำเสนอเปอร์เซ็นต์เบี่ยงเบนคำตอบที่ดีเมื่อเทียบกับคำตอบที่ทราบว่าเหมาะสมกว่า ดังนั้นสรุปได้ว่า อัลกอริทึมที่นำเสนอมีประสิทธิภาพ สามารถค้นพบคำตอบที่มีความเหมาะสมได้

#### 5.2 ข้อเสนอแนะ

อัลกอริทึมที่นำเสนอมีเป้าหมายเพื่อค้นหาคำตอบที่มีความเหมาะสมในชุดข้อมูลมาตรฐาน โดยไม่ได้ให้ความสำคัญกับเวลาในการประมวลผลข้อมูล และไม่ได้นำข้อมูลจริงที่ได้จากบริษัทจัดการขนส่งมาทดสอบกับอัลกอริทึม ดังนั้นในอนาคตเพื่อเพิ่มประสิทธิภาพของอัลกอริทึม จึงควรทำการปรับปรุงอัลกอริทึมที่นำเสนอให้ประมวลผลได้เร็วขึ้น และสามารถนำไปประยุกต์ใช้ในงานจริงได้

บทที่ 6  
สรุปผลผลิตงานวิจัย

6.1 สรุปรายชื่อและรายละเอียดผลผลิตงานวิจัยที่ผลิตได้และที่อยู่ระหว่างดำเนินการทั้งหมด

โครงการวิจัยนี้ได้ส่งบทความวิชาการชื่อเรื่อง การแก้ปัญหาการจัดเส้นทางรถขนส่งโดยวิธีการเมตาฮิวริสติกส์ เพื่อเผยแพร่ในวารสารวิชาการคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยกำลังอยู่ในขั้นตอนในการพิจารณาของวารสารฯ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] R. M. Rizk-Allah, Elsayed M. Zaki, Ahmed Ahmed El-Sawy. “Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems” *Applied Mathematics and Computer*. 224, 2013. pp 473-483.
- [2] กรรณ จิตเมตตา, ปารเมศ ชูติมา. “การประยุกต์ใช้วิธีการหาค่าที่เหมาะสมแบบการกระจายของสิ่งมีชีวิตตามภูมิศาสตร์สำหรับปัญหาการจัดลำดับการผลิตบนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้าน” *วารสารวิศวกรรมศาสตร์*. เล่มที่ 3, ฉบับที่ 1 ตุลาคม 2554. หน้า 1-18.
- [3] Frank Neumann, Carsten Witt. “Bioinspired Computation in Combinatorial Optimization Algorithm and Their Computational Complexity”, Springer, 2010.
- [4] Geir Hasle. “Discrete Optimization – Heuristics”, University of Jyvaskyla, Finland, 2009.
- [5] ชิตานันต์ ชิตานนท์. “การพัฒนาระบบสนับสนุนการตัดสินใจสำหรับการจัดเส้นทางยานพาหนะเพื่อการขนส่งโดยใช้เทคโนโลยีแผนที่อิเล็กทรอนิกส์” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ บัณฑิตวิทยาลัย, มหาวิทยาลัยเกษตรศาสตร์. 2557.
- [6] \_\_\_\_\_ . “บทที่ 6 ปัญหาการจัดเส้นทางขนส่ง” [Online]. เข้าถึงได้จาก: [http://www.ubu.ac.th/~pitakaso/1302476/new\\_doc/ch06\\_s.pdf](http://www.ubu.ac.th/~pitakaso/1302476/new_doc/ch06_s.pdf). 2560.
- [7] Mostafa Mahi, Omer Kaan Baykan, Halife Kodaz. “A new hybrid method based on Particle Swarm Optimization Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem” *Applied Soft Computing*. 30, 2015. pp 484-490.
- [8] Absalom El-shamir Ezugwu, Aderemi Oluyinka Adewumi, Marc Eduard Frincu. “Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem” *Expert Systems with Application*, vol. 77, 2017. pp. 189-210.
- [9] Yu Bin, Yang Zhong-Zhen, Yao Baozhen. “An improved ant colony optimization for vehicle routing problem” *European Journal of Operational Research*. 196, 2009. pp 171-176.
- [10] Bin Yu, Zhong Zhen Yang. “An ant colony optimization model: The period vehicle routing problem with time windows” *Transportation Research Part E*. 47, 2011. pp 166-181.
- [11] Mazin Abed Mohammed, Mohd Khanapi Abd Ghani, Raed Ibraheem Hamed, Salama A. Mostafa, Dheyaa Ahmed Ibrahim, Humam Khaled Jameel, Ahmed Hamed Alallah. “Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution” *Journal of Computational Science*, vol. 21, 2017. pp. 232-240.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [12] Eneko Osaba, Xin-She Yang, Fernando Diaz, Enrique Onieva, Antonio D. Masegosa, Asier Perallos. "A discrete firefly algorithm to solve a rich vehicle routing problem modeling a newspaper distribution system with recycling policy" *Soft Computing*, vol. 21, 2017. pp. 5295-5308.
- [13] Yannis Marinakis, Georgia-Roumbini Iordanidou, Magdalene Marinaki. "Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands" *Applied Soft Computing*, 13, 2013. pp 1693-1704.
- [14] Petrica C. Pop, Levente Fuksz, Andrei Horvat Marc, Cosmin Sabo. "A novel two-level optimization approach for clustered vehicle routing problem" *Computers & Industrial Engineering*, vol. 115, 2018. pp. 304-318.



ภาคผนวก  
สรุปค่าใช้จ่ายการดำเนิน โครงการวิจัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## ข้อมูลประวัติคณะผู้วิจัย

ประวัติส่วนตัว

ชื่อ-สกุล นายอาริต ธรรมโน

ตำแหน่งปัจจุบัน รองศาสตราจารย์

ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
วศ.บ.	วิศวกรรมระบบควบคุม	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	1991
MSEE	Electrical and Computer Engineering	University of Missouri, Columbia, MO., USA	1992
Ph.D.	Engineering Management	University of Missouri, Rolla, MO., USA	1995

ผลงานวิจัย/งานสร้างสรรค์

ผลงานวิจัย/งานสร้างสรรค์ที่ตีพิมพ์เผยแพร่ (ระดับนานาชาติ)

1. Firefly mating algorithm for continuous optimization problems, Computational Intelligence and Neuroscience, 2017.
2. Memetic algorithm based on marriage in honey bees optimization for flexible job shop scheduling problem, Memetic Computing, 2017.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้