การประยุกต์ใช้อัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย

APPLICATION OF FIREFLY ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM

นางสาวญาณินท์ หลิมเพียน

MS. YANIN LIMPHIEN

นายธนภูมิ แซ่ลี้

MR. THANAPOOM SAELEE

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

การประยุกต์ใช้อัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย

APPLICATION OF FIREFLY ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM

นางสาวญาณินท์ หลิมเพียน

MS. YANIN LIMPHIEN

นายธนภูมิ แซ่ลี้

MR. THANAPOOM SAELEE

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION OF FIREFLY ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM

MS. YANIN LIMPHIEN

MR. THANAPOOM SAELEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การประยุกต์ใช้อัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย
APPLICATION OF FIREFLY ALGORITHM FOR TRAVELING SALESMAN PROBLEM

นักศึกษา นางสาวญาณินท์ หลิมเพียน รหัสประจำตัวนักศึกษา 56010326
นายธนภูมิ แซ่ลี้ รหัสประจำตัวนักศึกษา 56010543

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(ดร.เชาวลิต หามนตรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้อัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย
นักศึกษา	นางสาวญาณินท์ หลิมเพียน นายธนภูมิ แซ่ลี้
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2559
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร.เชาวลิต หามนตรี

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการแก้ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP) โดยที่ระยะทางที่พนักงานขายเดินทางไปยังเมืองต่างๆ ต้องมีระยะทางรวมที่สั้นที่สุด อัลกอริทึมหิ่งห้อย (Firefly Algorithm, FA) ซึ่งเป็นหนึ่งในวิธีเมตาฮิวริสติก ที่ได้รับแรงบันดาลใจมาจากการเปล่งแสงของหิ่งห้อยที่มีอยู่ในธรรมชาติที่มีพฤติกรรมในการใช้แสงเพื่อดึงดูดหิ่งห้อยตัวอื่นๆ ให้เคลื่อนที่เข้าหามาประยุกต์ใช้ในการแก้ปัญหา TSP โดยผู้จัดทำได้ใช้โปรแกรม MATLAB ในการประมวลผลและนำคำตอบที่ได้มาทำการเปรียบเทียบกับค่าผลลัพธ์ที่ดีที่สุด (Optimal Solution), ผลที่ได้จากการคำนวณด้วยวิธีฮิวริสติก (Huerictic Method) และผลที่ได้จากการคำนวณด้วยวิธีการเชิงพันธุกรรม (Genetic Algorithm,GA) กับปัญหา TSP จำนวน 3 ตัวอย่าง ที่มีจำนวนเมืองต่างกัน ได้แก่ 16 เมือง 22 เมือง และ 52 เมือง จากการทดลองทำให้ทราบว่า อัลกอริทึมหิ่งห้อยมีความเหมาะสมในการแก้ไขปัญหา TSP เนื่องจากสามารถให้คำตอบที่เป็นค่าผลลัพธ์ที่ดีที่สุดได้ในปัญหาขนาด 16 เมือง และ 22 เมือง ส่วนปัญหาขนาด 52 เมือง อัลกอริทึมหิ่งห้อยก็สามารถให้คำตอบที่ดีกว่า การใช้วิธีฮิวริสติก และ วิธีการเชิงพันธุกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Application of Firefly Algorithm for Solving Traveling Salesman Problem
Student	Ms. Yanin Limphien
	Mr. Thanapoom Saelee
Degree	Bachelor of Engineering in Industrial Engineering
	King Mongkut's Institute of Technology Ladkrabang
Academic year	2016
Thesis Advisor	Dr. Chaowalit Hamontree

ABSTRACT

This project presents a method to solve the Traveling Salesman Problem (TSP). The objective of this problem is to find an optimal route with the minimum total distance while satisfies all the constraints of the problem. The Firefly Algorithm (FA) mimics the flashing behavior of fireflies. The idea of this algorithm is that the firefly which is the brighter one will move toward the less bright one. This project, therefore, will be presenting how to apply the FA to solve TSP. In order to obtain the optimum solution, MATLAB is applied to compile the particular problem. Three instances of TSP are used in this project; 16 cities, 22 cities and 52 cities. Then, we discuss the efficiency of FA by comparing its results with other methods including of Heuristic Method, Genetic Algorithm and the optimal solution as well. The experiment showed that the FA is a suitable method for solving TSP and also provided the optimal solution for the first instance at 16 cities and 22 cities. For the instance at 52 cities, FA outperforms Heuristic Method and GA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก ดร. เชาวลิต หามนตรี อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ให้โอกาสในการศึกษาปริญญานิพนธ์ฉบับนี้รวมทั้งที่กรุณาให้คำปรึกษา แนะนำ ตลอดจนชี้แนะแนวทางสำหรับปรับปรุงแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่อย่างดียิ่ง คณะผู้จัดทำมีความซาบซึ้งเป็นอย่างสูงในความกรุณาและความทุ่มเทที่อาจารย์ได้มอบให้นี้ จึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ยิ่งไปกว่านั้นปริญญานิพนธ์ฉบับนี้จะสำเร็จลุล่วงไปได้ด้วยดีไม่ได้เลย หากปราศจากคำแนะนำและความช่วยเหลือจาก ดร.นิรันดร์ พิสุทธอานนท์ ที่ได้ให้ความรู้คำแนะนำที่เป็นประโยชน์อย่างยิ่งในการจัดทำปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำมีความรู้สึกขอบคุณ และประทับใจในความช่วยเหลือที่อาจารย์ได้กรุณามอบให้เป็นอย่างสูง

คณะผู้จัดทำยังใคร่ขอขอบพระคุณ ผศ.ดร.ชุมพล ยวงใย อาจารย์รณน เจียรตระกูล และ ดร.ภาสุ พูนภักดี เป็นอย่างสูง ที่สละเวลาอันมีค่า เพื่อให้คำปรึกษาและแนวคิดต่างๆที่มีประโยชน์แก่คณะผู้จัดทำเป็นผลให้ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี

นอกจากนี้คณะผู้จัดทำขอขอบคุณผู้ที่เกี่ยวข้องทุกๆท่านที่คอยให้กำลังใจ และความช่วยเหลือต่างๆ จนเป็นผลให้คณะผู้จัดทำสามารถจัดทำปริญญานิพนธ์ฉบับนี้ได้สำเร็จ

อนึ่ง คณะผู้จัดทำหวังไว้เป็นอย่างสูงว่า ปริญญานิพนธ์ฉบับนี้ที่เกิดจากความทุ่มเทและความพยายาม จะมีค่า และเกิดประโยชน์แก่ผู้ที่สนใจ ตลอดจนผู้ที่เกี่ยวข้องต่างๆ ดังนั้นคณะผู้จัดทำจึงมีความประสงค์ที่จะขอมอบส่วนดีทั้งหมดนี้ให้แด่เหล่าคณาจารย์ที่ช่วยประสิทธิประสาทวิชา และในส่วนของข้อบกพร่องของปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอน้อมรับผิดไว้แต่เพียงผู้เดียว และยินดีที่จะรับฟังคำแนะนำต่างๆจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนาต่อไป

นางสาวญาณินท์ หลิมเพียน
นายธนภูมิ แซ่ลี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ค

สารบัญ.....ง

สารบัญตาราง.....ฉ

สารบัญรูป.....ช

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎีเกี่ยวกับปัญหาพนักงานขาย.....3

2.1.1 ความเป็นมาของปัญหาการเดินทางของพนักงานขาย.....3

2.1.2 นิยามของปัญหาการเดินทางของพนักงานขาย.....4

2.1.3 แบบจำลองทางคณิตศาสตร์ของปัญหาการเดินทางของพนักงานขาย.....5

2.2 แนวคิดและเทคนิคการแก้ปัญหาการจัดเส้นทางการเดินรถขนส่ง.....6

2.2.1 วิธีแม่นตรง.....7

2.2.2 วิธีฮิวริสติก.....7

2.2.3 วิธีเมตาฮิวริสติก.....8

2.3 วิธีการเชิงพันธุกรรม.....9

2.4 อัลกอริทึมหิ่งห้อย.....10

2.4.1 วิธีวิวัฒนาการอัลกอริทึมหิ่งห้อยสำหรับปัญหาไม่ต่อเนื่อง.....12

2.4.2 แผนการเคลื่อนที่แบบใหม่ของอัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย.....18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินงาน

3.1 ศึกษาและรวบรวมข้อมูล.....24

3.2 การออกแบบเส้นทางการเดินทางของพนักงานขายโดยการประยุกต์ใช้วิธีการเชิงพันธุกรรม.....30

3.3 การออกแบบเส้นทางการเดินทางของพนักงานขายโดยการประยุกต์ใช้วิธีอัลกอริทึมหิ่งห้อย.....33

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองจากการใช้วิธีการเชิงพันธุกรรมในการจัดเส้นทางการเดินทางของพนักงานขาย.....50

4.2 ผลการทดลองจากการใช้อัลกอริทึมหิ่งห้อยในการจัดเส้นทางการเดินทางของพนักงานขาย.....51

4.3 ผลการทดสอบพารามิเตอร์ และการวิเคราะห์ผลการทดลอง.....55

บทที่ 5 สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน.....59

5.2 ปัญหาและอุปสรรคในการดำเนินงาน.....59

5.3 ข้อเสนอแนะ.....59

เอกสารอ้างอิง.....61

ภาคผนวก ก.....ผก1

ภาคผนวก ข.....ผข1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 ข้อมูลพิกัดตำแหน่งของเมือง 16 เมือง.....25
ตารางที่ 3.2 ข้อมูลพิกัดตำแหน่งของเมือง 22 เมือง.....26
ตารางที่ 3.3 ข้อมูลพิกัดตำแหน่งของเมือง 52 เมือง.....27
ตารางที่ 3.4 ตารางแสดงค่าพารามิเตอร์สำหรับวิธีการเชิงพันธุกรรม.....30
ตารางที่ 3.5 ตารางแสดงค่าพารามิเตอร์สำหรับอัลกอริทึมหิ่งห้อย.....33
ตารางที่ 3.6 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 10, γ = 0.0001 และ t = 100.....34
ตารางที่ 3.7 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 100, γ = 0.001 และ t = 500.....35
ตารางที่ 3.8 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 500, γ = 0.01 และ t = 800.....36
ตารางที่ 3.9 ค่าพารามิเตอร์ที่ n = 5, m = 10, 100 และ 500, γ = 0.0001 และ t = 100.....37
ตารางที่ 3.10 ค่าพารามิเตอร์ที่ n = 10, m = 10, 100 และ 500, γ = 0.001 และ t = 500.....38
ตารางที่ 3.11 ค่าพารามิเตอร์ที่ n = 20, m = 10, 100 และ 500, γ = 0.01 และ t = 800.....39
ตารางที่ 3.12 ค่าพารามิเตอร์ที่ n = 5, m = 10, γ = 0.0001, 0.001 และ 0.01 และ t = 100.....40
ตารางที่ 3.13 ค่าพารามิเตอร์ที่ n = 10, m = 100, γ = 0.0001, 0.001 และ 0.01 และ t = 500.....41
ตารางที่ 3.14 ค่าพารามิเตอร์ที่ n = 20, m = 500, γ = 0.0001, 0.001 และ 0.01 และ t = 800.....42
ตารางที่ 3.15 ค่าพารามิเตอร์ที่ n = 5, m = 10, γ = 0.0001 และ t = 100, 500 และ 800.....43
ตารางที่ 3.16 ค่าพารามิเตอร์ที่ n = 10, m = 100, γ = 0.001 และ t = 100, 500 และ 800.....44
ตารางที่ 3.17 ค่าพารามิเตอร์ที่ n = 20, m = 500, γ = 0.01 และ t = 100, 500 และ 800.....45
ตารางที่ 4.1 ผลลัพธ์การจัดเส้นทางด้วยวิธีการเชิงพันธุกรรมของปัญหาขนาด 16, 22 และ 52 เมือง.....50
ตารางที่ 4.2 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 16 เมือง.....51
ตารางที่ 4.3 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 22 เมือง.....52
ตารางที่ 4.4 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 52 เมือง.....53
ตารางที่ 4.5 ตารางแสดงการเปรียบเทียบผลลัพธ์ที่ได้จากการใช้อัลกอริทึมต่างๆในการแก้ปัญหาการการเดินทางของพนักงานขาย.....56
ตารางที่ 4.6 ตารางเปรียบเทียบประสิทธิภาพในการหาคำตอบของอัลกอริทึมหิ่งห้อย และวิธีการเชิงพันธุกรรม ด้วยเปอร์เซ็นต์การเข้าใกล้ค่าผลลัพธ์ที่ดีที่สุด.....56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 เกมปริศนาหาเส้นทาง Icosian Game....4
รูปที่ 2.2 ปัญหาการเดินทางของพนักงานขาย....5
รูปที่ 2.3 รูปแบบโครโมโซมสำหรับปัญหาการเดินทางของพนักงานขาย....10
รูปที่ 2.4 ภาพแสดงการเคลื่อนที่ของหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า ไปยังตัวที่มีแสงมากกว่า....11
รูปที่ 2.5 รูปแบบของหิ่งห้อย 1 ตัว สำหรับปัญหาการเดินทางของพนักงานขาย....12
รูปที่ 2.6 แสดงคู่ขอบที่แตกต่างกันระหว่างหิ่งห้อยทั้ง 2 ตัว....13
รูปที่ 2.7 กราฟแสดงความสัมพันธ์ระหว่างค่าความดึงดูด และระยะทางระหว่างหิ่งห้อยทั้ง 2 ตัว ที่ค่าที่ค่าสัมประสิทธิ์การดูดซับแสงที่ [0.01,0.15]....14
รูปที่ 2.8 การกลายพันธุ์แบบอินเวอร์ชัน ที่นำมาใช้สำหรับการเคลื่อนที่ของหิ่งห้อย....17
รูปที่ 2.9 การเปรียบเทียบหาคู่ขอบที่แตกต่างกันของหิ่งห้อยทั้ง 2 ตัว....19
รูปที่ 2.10 การยืดส่วนตำแหน่ง x และ y....19
รูปที่ 2.11 การสลับส่วน โดยนำส่วน x มาไว้ด้านหน้าส่วน y....20
รูปที่ 2.12 การสลับส่วน โดยนำส่วน x มาไว้ด้านหลังส่วน y....21
รูปที่ 2.13 การสลับส่วน โดยนำส่วน y มาไว้ด้านหน้าส่วน x....22
รูปที่ 2.14 การสลับส่วน โดยนำส่วน y มาไว้ด้านหลังส่วน x....22
รูปที่ 2.15 คำตอบทั้ง 4 ที่ได้จากการเคลื่อนที่แบบคู่ขอบ....23
รูปที่ 3.1 ผังงานแสดงกระบวนการของวิธีการเชิงพันธุกรรม....32
รูปที่ 3.2 รหัสเทียมของอัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงาน....47
รูปที่ 3.3 แผนผังแสดงการทำงานของอัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย....48
รูปที่ 4.1 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 16 เมือง....57
รูปที่ 4.2 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 22 เมือง....57
รูปที่ 4.3 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 52 เมือง....58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP) เป็นปัญหาพื้นฐานที่สำคัญในการจัดการด้านโลจิสติกส์ และมีความสำคัญอย่างมากต่อระบบเศรษฐกิจของประเทศ โดยเฉพาะในภาคอุตสาหกรรม และการขนส่ง ปัญหา TSP เริ่มจากพนักงานขายต้องเดินทางไปพบลูกค้าแต่ละราย ตามสถานที่ หรือเมืองต่างๆ โดยต้องเดินทางไปพบลูกค้าให้ครบทุกราย และต้องเดินทางกลับมาที่จุดเริ่มต้นเสมอ ค่าใช้จ่ายในการเดินทางจึงเกิดจากระยะทางที่พนักงานขายต้องเดินทางไปพบลูกค้าในแต่ละเมือง ด้วยเหตุนี้จึงต้องวางแผนการเดินทาง เพื่อให้ได้เส้นทางที่สั้นที่สุดนั่นเอง แต่เมื่อจำนวนของลูกค้าเพิ่มมากขึ้น ปัญหาก็จะมีความซับซ้อนเพิ่มขึ้นตามไปด้วย ดังนั้นการวางแผนเพื่อหาเส้นทางที่ดีที่สุด และประหยัดที่สุด จึงสามารถเพิ่มความสามารถในการแข่งขัน อีกทั้งยังเป็นการเพิ่มผลกำไรโดยรวมอีกด้วย ดังนั้นหากสามารถลดต้นทุนทางด้านการขนส่งลงได้ ด้วยการจัดเส้นทางการเดินทางที่มีประสิทธิภาพ ก็จะทำให้ธุรกิจมีโอกาสในการประสบความสำเร็จมากขึ้น ทั้งในด้านของการลดต้นทุนและการตอบสนองต่อความต้องการของลูกค้าที่มีหลากหลายมากขึ้นอีกด้วย

1.2 วัตถุประสงค์

1. เพื่อเสนอแนวทางวิธีการจัดเส้นทางการเดินทางของพนักงานขายด้วยวิธีอัลกอริทึมหิ่งห้อย
2. เพื่อเปรียบเทียบผลลัพธ์ของวิธีอัลกอริทึมหิ่งห้อยกับการหาคำตอบด้วยอัลกอริทึมแบบอื่นๆ
3. เพื่อหาเส้นทางการเดินทางของพนักงานขายที่เหมาะสม

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาปัญหาการเดินทางของพนักงานขายที่มีจุดเริ่มต้นเพียงจุดเดียว กับลูกค้าหลายราย
2. พนักงานขายจะต้องเดินทางไปหาลูกค้าที่อยู่ในแต่ละเมือง ให้ครบทุกรายเพียงครั้งเดียว
3. การเดินทางแต่ละครั้ง ต้องเริ่มที่จุดเริ่มต้น และกลับมาสิ้นสุดที่จุดเริ่มต้นเท่านั้น
4. พนักงานขายมีเพียงหนึ่งคนเท่านั้น
5. ข้อมูลที่นำมาใช้ในการทดลองเป็นข้อมูลที่ได้จาก TSPLIB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำอัลกอริทึมหิ่งห้อยไปประยุกต์ใช้กับวิธีการแก้ไขปัญหาการเดินทางของพนักงานขายที่ยากและซับซ้อนขึ้น และปัญหาอื่นๆที่ใกล้เคียง
2. สามารถประยุกต์ใช้ความรู้ที่ได้จากการศึกษากับปัญหาทางด้านโลจิสติกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะได้กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องสำหรับแก้ปัญหาการเดินทางของพนักงานขาย โดยจะทำการศึกษาเกี่ยวกับทฤษฎี และงานวิจัยที่เกี่ยวข้องกับวิธีฮิวริสติก และวิธีเมตาฮิวริสติก รวมทั้งวิธีการอื่นๆที่นำไปประยุกต์ใช้กับปัญหาที่กล่าวมา โดยมีรายละอียดดังต่อไปนี้

2.1 แนวคิด และทฤษฎีเกี่ยวกับปัญหาพนักงานขาย

ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP) แม้จะดูเหมือนว่าเป็นปัญหาที่เรียบง่าย และไม่ซับซ้อน เนื่องจากสามารถให้คำนิยามได้อย่างคร่าวๆว่า เป็นการจัดเส้นทางการเดินทางของพนักงานขาย ให้เดินทางไปยังเมืองต่างๆที่ลูกค้าอาศัยอยู่ โดยที่พนักงานขายต้องเดินทางไปยังลูกค้าทุกคน และระยะทางรวมในการเดินทางของพนักงานขาย มีระยะทางที่สั้นที่สุด แต่ความจริงแล้ว ปัญหา TSP เป็นปัญหาที่มีความซับซ้อนสูง เมื่อขนาดของปัญหาใหญ่ขึ้น หรือจำนวนเมืองที่พนักงานขาย จำเป็นต้องเดินทางไปพบลูกค้ามีจำนวนมากขึ้นนั่นเอง [1] นอกจากนี้ปัญหา TSP ยังเป็นหนึ่งในปัญหาการหาค่าที่เหมาะสมที่สุดเชิงการจัดการ (Combintorial Optimization Problem) ที่ถูกกล่าวถึงมากที่สุดอีกด้วย [2]

2.1.1 ความเป็นมาของปัญหาการเดินทางของพนักงานขาย

ปัญหา TSP เป็นหนึ่งในปัญหาการจัดเส้นทางที่ถูกนิยมนำมาศึกษามากที่สุด ปัญหา TSP เกิดขึ้นครั้งแรกในศตวรรษที่ 18 โดยนักคณิตศาสตร์ชาวไอริช Sir William Rowan Hamilton และนักคณิตศาสตร์ชาวอังกฤษ Thomas Penington Kirkman

Sir William Rowan Hamilton ได้คิดค้นเกมปริศนาที่เรียกว่า Icosian Game ขึ้น โดยเกมนี้ ผู้เล่นจะต้องหาเส้นทางที่สั้นที่สุดผ่านจุดต่างๆ โดยที่ห้ามเดินทางผ่านแต่ละจุดเกิน 1 ครั้ง และสุดท้ายต้องเดินทางกลับไปยังจุดเริ่มต้นเสมอ ซึ่งต่อมาเส้นทางที่เดินจนครบเป็นวงนี้ ถูกเรียกว่า วงจรแฮมมิลตัน (Hamilton Cycle) งานต่างๆในช่วงแรกเริ่มของ Sir William Rowan Hamilton และ Thomas Penyngton Kirkman ได้ถูกอภิปรายเพิ่มเติมในหนังสือ Graph Theory 1736-1936 [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

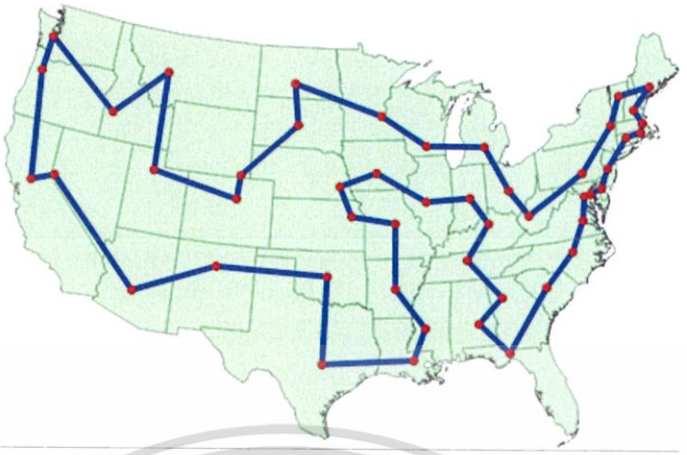
รูปที่ 2.1 เกมปริศนาหาเส้นทาง Icosian Game [4]

แม้ว่าต้นกำเนิดของปัญหา TSP นั้นจะยังไม่เป็นที่กระจ่างชัด แต่เชื่อกันว่ารูปแบบของปัญหานี้ได้ถูกศึกษาเป็นครั้งแรกในช่วงปี ค.ศ. 1930 โดย Karl Menger นักคณิตศาสตร์ ผู้ระบุนิยามของปัญหานี้ ต่อมานักคณิตศาสตร์ชาวอเมริกันทั้งสอง Hassler Whitney และ Merill M. Flood ได้เรียกปัญหาลักษณะนี้ว่า ปัญหา TSP [5]

2.1.2 นิยามของปัญหาการเดินทางของพนักงานขาย

ปัญหา TSP เป็นปัญหาที่ต้องทำการตัดสินใจหาเส้นทางการเดินทางที่สั้นที่สุดของพนักงานขาย เมื่อต้องเดินทางไปยังเมืองต่างๆ จำนวน n เมือง โดยเส้นทางการเดินทางนั้นๆจะต้องเดินทางผ่านเมืองทุกเมืองเพียงหนึ่งครั้งเท่านั้น และต้องกลับมายังเมืองที่เป็นจุดเริ่มต้นในการเดินทางเสมอ [5] โดยการแก้ปัญหา TSP นี้ จะได้คำตอบเป็นเส้นทางการเดินทาง ที่ประกอบไปด้วยลำดับเมืองที่จะต้องเดินทางไป การที่จะหาคำตอบที่ดีที่สุด สำหรับปัญหา TSP นี้ จำเป็นที่จะต้องหาทุกๆเส้นทางที่เป็นไปได้ออกมา และนำมาคัดเลือกเส้นทางที่ดีที่สุดเพียงเส้นทางเดียว และเนื่องจากปัญหา TSP นี้เป็นปัญหา NP-hard กล่าวคือ ยิ่งปัญหามีขนาดใหญ่ขึ้นเท่าไหร่ เวลาที่ใช้ในการคำนวณจะเพิ่มขึ้นแบบโพลิโนเมียล เนื่องจากจำนวนเส้นทางที่เป็นไปได้นั้น เกิดขึ้นเป็นจำนวนมหาศาล จึงทำให้เป็นเรื่องที่ยากในการที่จะคำตอบที่ดีที่สุดได้ [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ปัญหาการเดินทางของพนักงานขาย [7]

2.1.3 แบบจำลองทางคณิตศาสตร์ของปัญหาการเดินทางของพนักงานขาย

ปัญหา TSP เป็นปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย เมื่อต้องไปพบลูกค้ายังเมืองต่างๆ โดยมีเงื่อนไขว่า ต้องเดินทางไปพบลูกค้าที่อยู่ในแต่ละเมือง ให้ครบทุกเมือง โดยที่สามารถเดินทางผ่านแต่ละเมืองได้เพียงแค่ครั้งเดียว และเมื่อเดินทางไปครบทุกเมืองแล้ว จำเป็นที่จะต้องเดินทางกลับสู่เมืองที่เป็นจุดเริ่มต้นด้วย

สำหรับปัญหา TSP ในปริญญานิพนธ์ฉบับนี้ ระยะห่างระหว่างเมืองทั้ง 2 เมือง ที่พนักงานขายต้องเดินทางไป จะเริ่มต้นด้วยการเดินทางจากเมือง i ไปยังเมือง j ซึ่งระยะทางระหว่างเมืองทั้ง 2 นี้สมมาตรกัน และเป็นระยะทางแบบยูคลิด (Euclidean Distance) ดังนั้นจึงสามารถเขียนแบบจำลองทางคณิตศาสตร์ (Mathematical Model) ของปัญหา TSP ได้ดังนี้

ตัวแปรการตัดสินใจ (Decision Variable)

$$x_{ij} \begin{cases} 1 & \text{พนักงานขายเดินทางจากเมือง } i \text{ ไปยัง } j \\ 0 & \text{กรณีอื่นๆ} \end{cases}$$

ดัชนี (Indices)

$i =$ ตำแหน่งจุดเริ่มต้นของเส้นทาง โดยที่ $i = 1,2,3,\dots,n$ แทน เมืองแต่ละเมือง

$j =$ ตำแหน่งปลายทางของเส้นทาง โดยที่ และ $j = 1,2,3,\dots,n$ แทน เมืองแต่ละเมือง

$d_{ij} =$ ระยะทางที่พนักงานขายเดินทางจากเมือง i ไปยังเมือง j

$n =$ จำนวนเมืองที่พนักงานขายต้องเดินทางไป

$S =$ จำนวนเมืองที่อยู่ในเส้นทาง

$V =$ จำนวนเมืองทั้งหมด

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

กำหนดให้

x_i = พิกัดจุด x ของเมือง i

x_j = พิกัดจุด x ของเมือง j

y_i = พิกัดจุด y ของเมือง i

y_j = พิกัดจุด y ของเมือง j

สมการเป้าหมาย (Objective Function)

$$Mininmize\ Z = \sum_{i \neq j} d_{ij} x_{ij} \quad (1)$$

เงื่อนไขบังคับ (Constraints)

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i = 1 \ldots n \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1 \ldots n \quad (3)$$

$$\sum_{i.j \in S}^{n} x_{ij} \leq |S| - 1 \quad S \subset V,\ 2 \leq |S| \leq n - 2 \quad (4)$$

สมการเป้าหมายที่ (1) มีวัตถุประสงค์ในการลดระยะทางรวม ในการเดินทางของพนักงานขาย
เงื่อนไขบังคับที่ (2) และ (3) หมายความว่า พนักงานขายจะเดินทางไปยังเมืองแต่ละเมืองได้เพียง 1 ครั้ง เท่านั้น
เงื่อนไขบังคับที่ (4) เป็นเงื่อนไขป้องกันการเกิดทัวร์ย่อย (Subtour)

2.2 แนวคิด และเทคนิคการแก้ปัญหาการจัดเส้นทางการเดินรถขนส่งสินค้า

ดังที่ได้กล่าวไว้ก่อนหน้านี้ การที่จะหาคำตอบของปัญหา TSP นั้น จะต้องหาเส้นทางที่เป็นไปได้ทั้งหมด ที่พนักงานขายหนึ่งคนจะสามารถเดินทางไปครบทุกๆเมืองได้ โดยมีเงื่อนไขว่า พนักงานขายจะสามารถเดินทางผ่านเมืองแต่ละเมืองได้เพียง 1 ครั้งเท่านั้น และเมื่อเดินทางไปครบทุกเมืองแล้ว ต้องเดินทางจากเมืองสุดท้าย กลับมายังจุดเริ่มต้นอีกด้วย จากนั้นจึงทำการเลือกเส้นทางที่มีระยะทางสั้นที่สุดเพื่อเป็นคำตอบที่ดีที่สุดของปัญหา แต่เนื่องจากว่าปัญหา TSP เป็นปัญหาที่จัดอยู่ในระดับ NP-hard ฉะนั้นเมื่อขนาดของปัญหาใหญ่ขึ้นเพียงเล็กน้อย จำนวนของคำตอบจะเพิ่มขึ้นเป็นจำนวนมาก จึงทำให้การหาคำตอบ หรือเส้นทางที่ดีที่สุดนั้นเป็นไม่ได้ เพราะต้องใช้เวลาในการคำนวณอย่างมหาศาลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 วิธีแม่นตรง

วิธีแม่นตรง (Exact Method) เป็นวิธีที่ใช้ในการหาคำตอบที่มีประสิทธิภาพมากวิธีหนึ่ง เนื่องจากข้อได้เปรียบของวิธีแม่นตรงคือ การได้ผลคำตอบที่มีค่าที่ดีที่สุด แต่เนื่องด้วยวิธีแม่นตรงเป็นวิธีที่คำนวณทุกความเป็นไปได้ของผลเฉลย จากนั้นจึงทำการเลือกค่าผลเฉลยที่ดีที่สุด (Global Optima) ถึงแม้ว่าการใช้วิธีแม่นตรงจะสามารถรับประกันได้ว่าคำตอบที่ได้มา จะเป็นคำตอบที่ดีที่สุดอย่างแน่นอน แต่วิธีแม่นตรงนี้ก็ยังมีข้อจำกัดอยู่ ซึ่งก็คือ เมื่อขนาดของปัญหามีขนาดใหญ่ขึ้น ความซับซ้อนมากขึ้น ทำให้เวลาที่ต้องใช้ในการประมวลผลเพื่อหาคำตอบนั้น มีระยะเวลานานมากจนเกินไป ด้วยเหตุนี้เองสำหรับกรณีที่ปัญหามีขนาดใหญ่มากๆ และความซับซ้อนสูงมากๆ การแก้ปัญหาด้วยวิธีการแม่นตรงอาจไม่สามารถทำได้ เพราะต้องใช้เวลาในการประมวลผลมากเกินไป แม้ว่าจะนำคอมพิวเตอร์ที่มีประสิทธิภาพสูงมากมาใช้ในการประมวลผลแล้วก็ตาม ทำให้ในทางปฏิบัติจึงเป็นไปไม่ได้ที่จะหาคำตอบด้วยวิธีนี้ ฉะนั้นจึงมีความจำเป็นที่จะต้องนำวิธีการอื่นๆที่จะมาใช้แก้ปัญหาแทน โดยที่คำตอบที่ได้ต้องเป็นคำตอบที่ดีมากพอ และเวลาที่ใช้ในการประมวลผลคำตอบไม่นานจนเกินไป ตัวอย่างวิธีการแม่นตรงที่นำมาใช้กับปัญหา TSP ได้แก่ Branch and Bound [8], Branch and Cut [9] เป็นต้น

2.2.2 วิธีฮิวริสติก

วิธีฮิวริสติก (Heuristic Method) เป็นวิธีที่ถูกพัฒนาขึ้นเพื่อหาคำตอบของปัญหาที่มีความซับซ้อนสูง และต้องใช้ระยะเวลาจำนวนมากในการประมวลผลเพื่อหาคำตอบ คำว่า Heuristic มีรากศัพท์มาจากคำในภาษากรีก “Heuriskein” มีความหมายว่า “การหา” หรือ “การค้นพบ” วิธีฮิวริสติกได้ถูกนำมาใช้แก้ปัญหาที่เป็นการหาค่าที่เหมาะสม เมื่อวิธีแม่นตรงไม่สามารถนำมาใช้ได้ เนื่องจากใช้ระยะเวลานานมากเกินไปในการหาคำตอบ วิธีฮิวริสติกสามารถให้คำตอบของปัญหาเหล่านั้นได้ ภายในระยะเวลาอันรวดเร็ว และคุณภาพของคำตอบที่ได้รับก็มีคุณภาพที่ดี แม้ว่าจะไม่ใช่คำตอบที่ดีที่สุดของปัญหาก็ตาม [10] ตัวอย่างวิธีการฮิวริสติกส์ที่สามารถนำมาประยุกต์ใช้กับปัญหา TSP ได้แก่ Insertion Method [11], Nearest Neighbour [12] เป็นต้น

การเปรียบเทียบประสิทธิภาพในการหาคำตอบของปัญหา TSP ในปริญญานิพนธ์ฉบับนี้ เป็นการนำเอาวิธีการต่างๆที่ใช้ในการแก้ปัญหาจำนวน 3 วิธีมาเปรียบเทียบกัน ได้แก่ วิธีฮิวริสติก, ขั้นตอนวิธีเชิงพันธุกรรม และอัลกอริทึมหิ่งห้อย โดยนำคำตอบที่ได้จากวิธีทั้ง 3 นั้น มาเปรียบเทียบกับผลลัพธ์ที่เหมาะสมที่สุด (Optimal Solution) ของแต่ละปัญหา โดยที่วิธีฮิวริสติกที่ได้นำมาใช้ในปริญญานิพนธ์ฉบับนี้คือ วิธี Nearest Neighbour ซึ่งมีแนวคิดในการจัดเส้นทางการเดินทางของพนักงานขายโดย เลือกลำดับเมืองที่จะต้องเดินทางเป็นลำดับถัดไป จากเมืองที่มีระยะห่างที่สั้นที่สุด เมื่อเทียบกับเมืองที่พนักงานขายอยู่ ณ ปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 วิธีเมตาฮิวริสติก

วิธีเมตาฮิวริสติก (Metahueristic Method) เป็นวิธีการหาคำตอบ ที่มีความยืดหยุ่นกว่าวิธีฮิวริสติก คำว่า "Metaheuristic" มีรากศัพท์มาจากคำในภาษากรีก 2 คำด้วยกัน ได้แก่ คำว่า "Metá" ซึ่งมีความหมายว่า "เหนือกว่า" กับคำว่า "Heuriskein" ที่หมายถึง "การหา" หรือ "การค้นพบ" [13] วิธีเมตาฮิวริสติกที่เป็นที่รู้จักกันดี ได้แก่ ขั้นตอนวิธีการจำลองการอบอ่อน (Stimulated Annealing) [14], การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization) [15], วิธีอาณานิคมมด (Ant Colony Optimization) [16] และ วิธีการเชิงพันธุกรรม (Genetic Algorithm) [17] เป็นต้น

วิธีฮิวริสติก และเมตาฮิวริสติกถูกคิดค้นขึ้น เพื่อนำมาแก้ปัญหาที่มีความซับซ้อนสูงมากๆ จนไม่สามารถใช้วิธีแม่นตรงในการหาคำตอบที่เหมาะสมที่สุดได้ภายในเวลาที่จำกัด [18] กล่าวคือวิธีฮิวริสติกและเมตาฮิวริสติกเป็นวิธีที่สามารถแก้ปัญหาที่มีความซับซ้อนสูงได้ โดยคำตอบที่ได้เป็นคำตอบที่ดี และมีประสิทธิภาพมากพอที่จะยอมรับได้ อีกทั้งยังสามารถหาคำตอบได้ในเวลาอันรวดเร็ว แต่กระนั้นคำตอบที่ได้จากวิธีฮิวริสติก และวิธีเมตาฮิวริสติก อาจไม่ใช่คำตอบที่ดีที่สุด และเหมาะสมที่สุด นอกจากนี้ สิ่งที่ต้องคำนึงถึงอีกประการหนึ่งคือ เมื่อนำอัลกอริทึมแบบวิธีฮิวริสติก และเมตาฮิวริสติก มาใช้ในการแก้ปัญหาใดๆ ต้องระลึกเสมอว่า อัลกอริทึมหนึ่งๆที่ใช้แก้ปัญหาหนึ่งๆได้ดี อาจไม่เหมาะสมกับการนำไปใช้กับปัญหาแบบอื่นๆได้ [18] แต่ทั้งนี้ทั้งนั้นแนวคิดของการนำอัลกอริทึมเหล่านี้มาใช้ก็คือ การเลือกใช้อัลกอริทึมให้เหมาะสมกับปัญหา โดยอัลกอริทึมที่เหมาะสมจะนำมาแก้ปัญดาใดๆ มักจะให้คำตอบที่น่าพอใจได้บ่อยครั้ง และมีประสิทธิภาพ เหมาะกับปัญหาที่ต้องการแก้

ปัจจุบันได้มีการคิดค้นและพัฒนาวิธีการทางเมตาฮิวริสติกแบบใหม่ๆขึ้นมาอย่างต่อเนื่อง และเป็นที่น่าสังเกตว่าวิธีเมตาฮิวริสติกสมัยใหม่ ไม่ว่าจะเป็น วิธีการเชิงพันธุกรรม, การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค, วิธีอาณานิคมมด, วิธีอาณานิคมผึ้ง (Artificial Bee Colony) [19], วิธีหาค่าเหมาะสมที่สุดแบบนกดุเหว่า (Cuckoo Search) [20], อัลกอริทึมหิ่งห้อย (Firefly Algorithm) [21][22] และอื่นๆอีกมากมาย มักเป็นอัลกอริทึมที่ได้รับแรงบันดาลใจจากธรรมชาติ และในบรรดาอัลกอริทึมจำพวกที่ได้รับแรงบันดาลใจจากธรรมชาติ อัลกอริทึมหิ่งห้อยก็เป็นอีกอัลกอริทึมหนึ่งที่โดดเด่น และน่าสนใจเนื่องจากเป็นอัลกอริทึมที่ค่อนข้างใหม่ รวมทั้งมีงานวิจัยที่บ่งบอกว่าอัลกอริทึมหิ่งห้อยเป็นอัลกอริทึมที่มีประสิทธิภาพและสามารถประมวลผลได้อย่างรวดเร็ว [23][24][25] และสามารถแก้ปัญหาการเดินทางของพนักงานขายได้อย่างมีประสิทธิภาพ [21][22] ทางคณะผู้จัดทำจึงได้เลือกนำเอาอัลกอริทึมหิ่งห้อยมาประยุกต์ใช้ในการแก้ปัญหา TSP เพื่อศึกษาขั้นตอนการทำงานของอัลกอริทึมหิ่งห้อย รวมทั้งความเหมาะสมในการใช้แก้ปัญหา TSP อีกด้วย

ดังนั้นปริญญานิพนธ์ฉบับนี้ จึงเป็นการนำอัลกอริทึมหิ่งห้อย ซึ่งเป็นหนึ่งในวิธีเมตาฮิวริสติกที่ได้รับแรงบันดาลใจจากสิ่งมีชีวิตในธรรมชาติ มาใช้แก้ปัญหา TSP เพื่อทดสอบดูว่าอัลกอริทึมหิ่งห้อยนี้มีความเหมาะสมในการนำมาประยุกต์ใช้กับแก้ปัญหานี้หรือไม่ และนำผลลัพธ์ที่ได้จากการแก้ปัญหา TSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเปรียบเทียบกับผลลัพธ์ที่ได้จากวิธีฮิวริสติก ขั้นตอนวิธีเชิงพันธุกรรม และคำตอบที่ดีที่สุดจาก TSPLIB เพื่อวัดประสิทธิภาพในการแก้ปัญหาของอัลกอริทึม

2.3 วิธีการเชิงพันธุกรรม

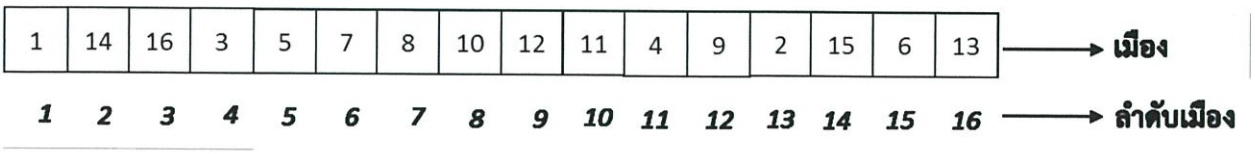
ในช่วงปี ค.ศ. 1960 ถึง 1970 วิธีการเชิงพันธุกรรม ถูกพัฒนาขึ้นโดย John Holland และคณะ โดยยึดตามทฤษฎีการวัฒนาการของสิ่งมีชีวิตของ Charles Darwin (Charles Darwin's Theory of Natural Selection) หลักการของ การครอสโอเวอร์ และ รีคอมไบเนชัน (Crossover and Recombination), มิวเทชัน (Mutation) และ การคัดเลือก (Selection) ได้ถูก John Holland นำมาเป็นต้นแบบในการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมเป็นครั้งแรก เพื่อนำมาเป็นกลยุทธ์ในการแก้ไขปัญหา [18]

การนำวิธีการเชิงพันธุกรรมมาใช้แก้ปัญหา จะต้องเริ่มจากการเข้ารหัส (Encoding) ของปัญหาที่ต้องการหาค่าที่เหมาะสม ให้อยู่ในรูปของ บิท (Bit) หรือ สตริง (String) เพื่อให้อยู่ในรูปของโครโมโซม (Chromosome) ที่พร้อมจะนำไปผ่านกระบวนการต่างๆทางพันธุกรรม จากนั้นจะทำการเลือกโครโมโซมที่ดีที่สุด สำหรับปัญหา TSP โครโมโซมที่ดีที่สุดคือ เส้นทางที่มีระยะทางรวมสั้นที่สุดนั่นเอง โดยขั้นตอนต่างๆ มีดังต่อไปนี้

1. การเข้ารหัสฟังก์ชันวัตถุประสงค์ (Objective Function)
2. ระบุฟังก์ชั่นเหมาะสม (Fitness Function)
3. สร้างจำนวนประชากรเริ่มต้น
4. พัฒนาค่าความเหมาะสมของประชากร
5. สร้างประชากรชุดใหม่ โดยผ่านกระบวนการ เช่น ครอสโอเวอร์, มิวเทชัน, การเพิ่มจำนวน (Reproduction) เป็นต้น
6. พัฒนาค่าความเหมาะสมของประชากรต่อไปเรื่อยๆ จนครบจำนวนรอบที่กำหนดไว้ (Maximum Generation)
7. ถอดรหัส (Decoding) เพื่อให้ได้คำตอบที่ต้องการ

สำหรับปัญหา TSP คำตอบที่เราต้องการคือ ลำดับของเมืองต่างๆ ที่พนักงานขายต้องเดินทางไป และในวิธีการเชิงพันธุกรรมจะแสดงผลลัพธ์ออกมาในรูปแบบโครโมโซม ดังรูปที่ 2.0 โดยที่โครโมโซม 1 โครโมโซม คือ ผลลัพธ์ 1 ผลลัพธ์ สำหรับปัญหา TSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 รูปแบบโครโมโซมสำหรับปัญหาการเดินทางของพนักงานขาย

2.4 อัลกอริทึมหิ่งห้อย

หิ่งห้อยเป็นสิ่งมีชีวิตที่น่ามหัศจรรย์ มีจำนวนสายพันธุ์ประมาณ 2,000 สายพันธุ์ ลักษณะเด่นของแมลงชนิดนี้ คือเป็นแมลงที่สามารถเปล่งแสงสว่างออกมาได้ โดยรูปแบบของการเปล่งแสงในหิ่งห้อยจะแตกต่างกันไปตามแต่ละสายพันธุ์

การเปล่งแสงของหิ่งห้อยมาจากกระบวนการเรืองแสงทางชีวภาพ (Bioluminescence) การเปล่งแสงของหิ่งห้อยมีวัตถุประสงค์แตกต่างกันไป เช่น การเปล่งแสงเพื่อการจับคู่ การเปล่งแสงเพื่อการสื่อสาร การเปล่งแสงเพื่อดึงดูดให้เหยื่อเข้ามาใกล้ การเปล่งแสงเพื่อเป็นสัญญาณเตือนศัตรูไม่ให้เข้ามาใกล้ เป็นต้น

ในปี ค.ศ. 2008 Xin She Yang [18] ได้คิดค้นอัลกอริทึมหิ่งห้อยขึ้น โดยได้รับแรงบันดาลใจจากหิ่งห้อยที่มีอยู่ในธรรมชาติ หลักการทำงานของอัลกอริทึมหิ่งห้อยนี้ ก็คือการเลียนแบบพฤติกรรมการเปล่งแสงของหิ่งห้อยนั่นเอง

ความเข้มแสงจากการเปล่งแสงของหิ่งห้อย (I) จะแปรผกผันกับระยะทางกำลังสอง (r^2) ซึ่งสามารถอธิบายได้ด้วยกฎกำลังสองผกผัน (Inverse square law) ที่แสดงในรูปแบบสมการทางคณิตศาสตร์ได้ดังนี้ $I \propto \frac{1}{r^2}$ กล่าวคือความเข้มแสงจะลดลง เมื่อระยะทางเพิ่มขึ้น นอกจากนี้การลดลงของความเข้มแสง ยังมาจากการที่อากาศซึ่งเป็นตัวกลางในการเดินทางของแสงมีความสามารถในการดูดซับแสงของหิ่งห้อยที่เปล่งออกมาได้ ทำให้ความเข้มแสงของหิ่งห้อยนั้นมีค่าลดลงเมื่อระยะทางเพิ่มขึ้น ด้วยเหตุนี้เองจึงทำให้ความสามารถในการรับรู้แสงของหิ่งห้อยมีขีดจำกัด แต่กระนั้นก็ยังมากพอที่จะทำให้สามารถสื่อสารถึงกันได้

พฤติกรรมการเปล่งแสงในหิ่งห้อยสามารถนำมาประมวลเป็นอัลกอริทึมได้ โดยกำหนดให้พฤติกรรมการเปล่งแสงของหิ่งห้อยที่เกิดขึ้น ดำเนินไปตามกฎอุดมคติ 3 ข้อ ดังนี้

1. กำหนดให้หิ่งห้อยทุกตัวเป็นได้ทั้งเพศผู้ และเพศเมีย ดังนั้นหิ่งห้อยตัวหนึ่งจะบินเข้าหาอีกตัวหนึ่งโดยไม่คำนึงถึงเรื่องเพศ
2. อัตราการดึงดูดกันระหว่างหิ่งห้อยทั้ง 2 ตัว จะเป็นสัดส่วนกับความสว่างที่หิ่งห้อยตัวหนึ่ง ถูกมองโดยหิ่งห้อยอีกตัวหนึ่ง หมายความว่า สำหรับหิ่งห้อย 2 ตัวใดๆ หิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า จะบินเข้าหาหิ่งห้อยตัวที่มีแสงสว่างมากกว่า โดยที่อัตราการดึงดูดจะแปรผกผันกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

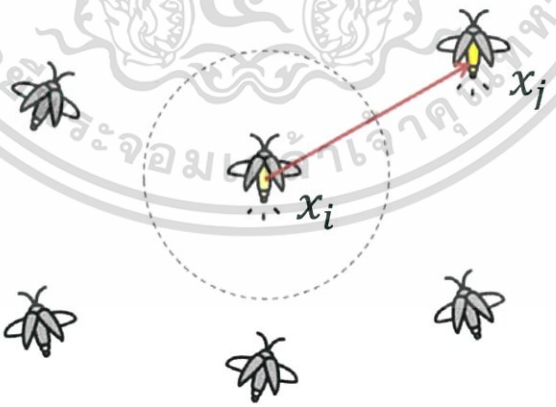
ระยะทาง และในกรณีที่ไม่มีหิ่งห้อยตัวใดสว่างมากกว่าหิ่งห้อยตัวที่กำลังพิจารณาอยู่ หิ่งห้อยตัวนั้นจะเคลื่อนที่แบบสุ่ม

3. ความสว่างของหิ่งห้อยจะถูกกำหนดโดยภูมิทัศน์ของสมการเป้าหมาย

จากกฎอุดมคติ 3 ข้อ สามารถอธิบายพฤติกรรมที่หิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า (x_i) บินเข้าหาหิ่งห้อยตัวที่มีแสงสว่างมากกว่า (x_j) ได้ดังนี้

$$x_i = x_i + \beta_0 e^{-\gamma r^2{}_{ij}}(x_j - x_i) + \alpha\epsilon_i \qquad (5)$$

จากสมการที่ (5) สามารถอธิบายถึงการเคลื่อนตำแหน่งของหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า ไปยังตำแหน่งของหิ่งห้อยที่มีแสงมากกว่าได้ว่า ตำแหน่งเดิมของหิ่งห้อยตัวที่มีแสงน้อยกว่าคือ x_i และตำแหน่งใหม่ของหิ่งห้อยตัวที่มีแสงน้อยกว่า เมื่อเคลื่อนที่เข้าหาหิ่งห้อยตัวที่มีแสงมากกว่า คือผลรวมของตำแหน่งเดิมของหิ่งห้อยตัวที่มีแสงน้อยกว่า (x_i) กับค่าความดึงดูดระหว่างหิ่งห้อย x_i และ x_j ที่สัมพันธ์กับค่าความสว่างของหิ่งห้อย x_j ที่มองเห็นได้โดยหิ่งห้อย x_i ในภูมิทัศน์ที่มีสัมประสิทธิ์การดูดซับแสงค่าหนึ่ง $(\beta_0 e^{-\gamma r^2{}_{ij}}(x_j - x_i))$ และ ค่าการสุ่ม $(\alpha\epsilon_i)$



รูปที่ 2.4 ภาพแสดงการเคลื่อนที่ของหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า ไปยังตัวที่มีแสงมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

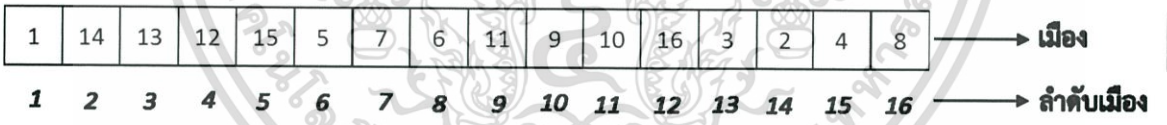
2.4.1 วิธีวิวัฒนาการอัลกอริทึมหิ่งห้อยสำหรับปัญหาไม่ต่อเนื่อง

อัลกอริทึมหิ่งห้อยนี้ เดิมทีถูกออกแบบมาเพื่อให้แก้ปัญหาที่เป็นปัญหาแบบต่อเนื่อง เมื่อนำอัลกอริทึมหิ่งห้อยมาประยุกต์ใช้เพื่อแก้ปัญหา TSP ซึ่งเป็นปัญหาแบบไม่ต่อเนื่อง จึงจำเป็นต้องปรับอัลกอริทึมหิ่งห้อยให้เหมาะกับการแก้ปัญหาที่ไม่ต่อเนื่องเสียก่อน จึงจะสามารถนำไปใช้ได้ วิธีวิวัฒนาการอัลกอริทึมหิ่งห้อยสำหรับปัญหาไม่ต่อเนื่อง (Evolutionary Discrete Firefly Algorithm) ได้นำเอาหลักการพัฒนาปรับปรุงคำตอบจาก วิธีกลยุทธ์การวิวัฒนาการ (Evolution Strategy) [26] มาใช้ โดยเลือกใช้วิธีการกลายพันธุ์แบบอินเวอร์ชัน (Inversion Mutation) [21]

2.4.1.1 รูปแบบหิ่งห้อยที่เกิดขึ้นในปัญหาการเดินทางของพนักงานขาย

สำหรับการประยุกต์ใช้อัลกอริทึมหิ่งห้อยกับปัญหา TSP นี้ หิ่งห้อย 1 ตัว จะประกอบไปด้วยลำดับเมืองที่พนักงานขายต้องเดินทางไปทั้งหมด ดังนั้นจึงกล่าวได้ว่า หิ่งห้อย 1 ตัว คือคำตอบของปัญหา 1 คำตอบนั่นเอง โดยระยะทางรวมทั้งหมดที่พนักงานขายต้องเดินทาง (R) สามารถคำนวนได้ตามสมการที่ 6

$$R = dist\,(x[1], x[n]) + \sum_{k=1}^{n-1} dist(x[k], x[k+1]) \qquad (6)$$



รูปที่ 2.5 รูปแบบของหิ่งห้อย 1 ตัว สำหรับปัญหาการเดินทางของพนักงานขาย

2.4.1.2 ค่าความเข้มแสงของหิ่งห้อยในอัลกอริทึมหิ่งห้อย

หิ่งห้อยแต่ละตัว จะมีค่าความสว่าง (β) โดยที่ค่าความสว่างของหิ่งห้อย จะถูกกำหนดโดยค่าความเข้มแสง (I) หิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า จะเคลื่อนที่เข้าหาหิ่งห้อยตัวที่มีแสงสว่างมากกว่า และความเข้มแสงของหิ่งห้อยตัวหนึ่ง เมื่อถูกมองจากหิ่งห้อยอีกตัวหนึ่ง จะแปรผกผันกับระยะทางดังนี้

$$I = \frac{1}{R} \qquad (7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะเห็นได้ว่า ยิ่งระยะทางเพิ่มมากขึ้นเท่าใด ความเข้มแสงของหิ่งห้อยก็จะลดลง หรือสามารถกล่าวอีกนัยหนึ่งได้ว่า หิ่งห้อยตัวหนึ่งๆ จะมองเห็นหิ่งห้อยตัวที่อยู่ใกล้กว่ามีแสงสว่างมากกว่าหิ่งห้อยตัวที่อยู่ไกลกว่านั่นเอง

2.4.1.3 ระยะห่างระหว่างหิ่งห้อยทั้ง 2 ตัว ในปัญหาการเดินทางของพนักงานขาย

สำหรับปัญหาที่เป็นลักษณะแบบต่อเนื่อง ระยะทางระหว่างหิ่งห้อยทั้ง 2 ตัว นั้นจะเป็นระยะทางแบบยูคลิด แต่ในปัญหาแบบไม่ต่อเนื่อง แบบปัญหา TSP นี้ ระยะห่างระหว่างหิ่งห้อยทั้ง 2 ตัว นั้นจะสามารถหาได้จากจำนวนคู่ขอบที่ต่างกัน (Different Edges) ของหิ่งห้อยทั้ง 2 ตัว

เมื่อหิ่งห้อยตัวที่มีแสงสว่างมากกว่าคือ x_i และหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่าคือ x_j การระบุระยะห่างระห่างระหว่างหิ่งห้อยทั้ง 2 ตัวนั้น สามารถทำได้โดย หาคู่ขอบที่มีอยู่ใน x_i แต่ไม่มีใน x_j ดังเช่นรูปที่ 2.5 เราจะทราบได้ว่าระยะห่างระหว่างหิ่งห้อยทั้ง 2 ตัว คือ 12-15, 15-5, 5-7, 7-6 และ 6-7 ซึ่งรวมแล้ว จำนวนคู่ขอบที่แตกต่างกันทั้งหมดคือ 5 คู่ขอบ จากนั้นเมื่อทราบจำนวนคู่ขอบที่ต่างกันทั้งหมดแล้ว จะสามารถหาระยะห่างระหว่างหิ่งห้อย x_i และ x_j ได้โดย นำไปคำนวณตามสมการที่ 8

$$r = \frac{A}{n} \times 10 \tag{8}$$

โดยที่ r คือระยะห่างระหว่างหิ่งห้อยทั้ง 2 ตัว, A คือจำนวนคู่ขอบที่แตกต่างกันทั้งหมด และ n คือจำนวนเมืองทั้งหมดที่พนักงานขายต้องเดินทางไป

Firefly i	1	14	13	12	15	5	7	6	11	9	10	16	3	2	4	8
Firefly j	1	14	13	12	7	6	15	5	11	9	10	16	3	2	4	8

รูปที่ 2.6 แสดงคู่ขอบที่แตกต่างกันระหว่างหิ่งห้อยทั้ง 2 ตัว

2.4.1.4 ค่าความดึงดูดระหว่างหิ่งห้อยทั้ง 2 ตัว

การที่หิ่งห้อยตัวหนึ่งๆ จะเคลื่อนที่เข้าหาหิ่งห้อยตัวอื่น เกิดจากการดึงดูดที่เกิดขึ้นระหว่างตัวมันเอง และหิ่งห้อยตัวอื่นที่พิจารณา สำหรับค่าความดึงดูดที่เกิดขึ้นระหว่างหิ่งห้อยทั้ง 2 ตัวนั้น คือ β_r โดยที่ r คือระยะห่างระหว่างหิ่งห้อยทั้ง 2 ตัวนั้น และ β_0 คือค่าความสว่างของหิ่งห้อยตัวที่มีแสงสว่างมากกว่า เราสามารถหาค่าความดึงดูดระหว่างหิ่งห้อยทั้ง 2 ตัว ได้จากสมการที่ 9 นอกจากนี้ ในบริเวณที่หิ่งห้อย

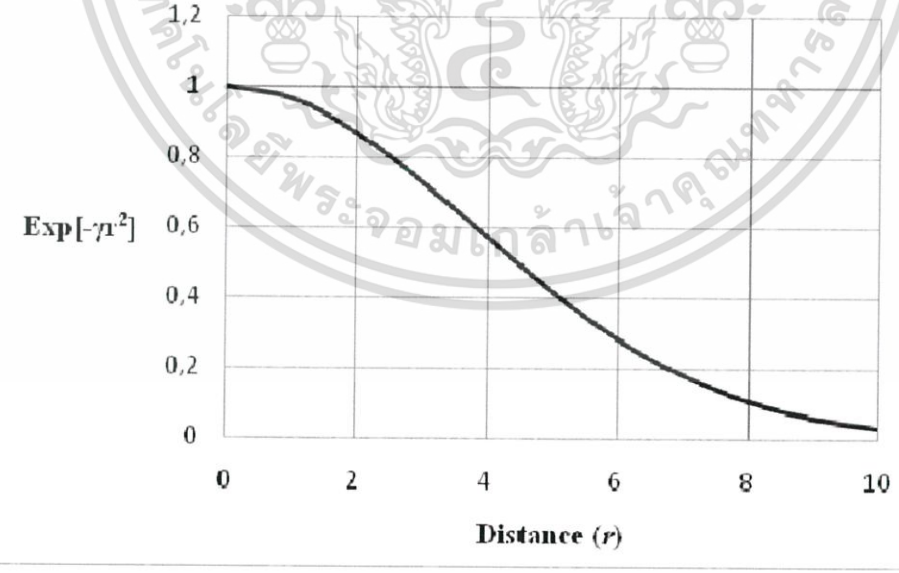
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดอาศัยอยู่นั้น จะมีภูมิทัศน์ที่ถูกกำหนดขึ้น ซึ่งทำให้ความสว่างของหิ่งห้อยตัวที่มีแสงมากกว่า (x_j) ซึ่งถูกมองโดยหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า (x_i) มีค่าเปลี่ยนไป ประหนึ่งว่าในพื้นที่ที่มีภูมิทัศน์ที่เต็มไปด้วยหมอกแน่นหนานั้น ทำให้หิ่งห้อยตัวหนึ่งๆ มองเห็นหิ่งห้อยตัวอื่นๆ มีแสงสว่างลดลง ซึ่งภูมิทัศน์ที่เกิดขึ้นนี้ ถูกกำหนดโดย ค่าสัมประสิทธิ์การดูดซับแสง (γ) นั่นเอง

$$\beta_r = \beta_0 e^{-\gamma r^2} \tag{9}$$

2.4.1.5 ค่าสัมประสิทธิ์การดูดซับแสง

ค่าสัมประสิทธิ์การดูดซับแสง (γ) มีความสำคัญมากในการกำหนดว่า หิ่งห้อยแต่ละตัวนั้นจะเคลื่อนที่แบบใด จากทฤษฎี ค่าสัมประสิทธิ์การดูดซับแสงนั้น สามารถมีค่าได้ตั้งแต่ 0 ไปจนถึง อนันต์ $(\gamma\epsilon[0,\infty))$ กล่าวคือ เมื่อ $\gamma \to 0$ ค่าความดึงดูดจะเป็นค่าคงที่ $(\beta_r = \beta_0)$ ซึ่งในกรณีนี้ ค่าความดึงดูดของหิ่งห้อยตัวหนึ่ง เมื่อถูกมองโดยหิ่งห้อยตัวอื่น จะไม่ลดลง เหตุเพราะว่าการดูดซับแสงแทบไม่เกิดขึ้นนั่นเอง และกรณีที่ $\gamma \to \infty$ จะเปรียบได้กับ หิ่งห้อยแต่ละตัวที่อาศัยอยู่ในภูมิทัศน์ที่มีหมอกหนาแน่นมาก จึงทำให้หิ่งห้อยตัวหนึ่งๆ มองไม่เห็นแสงสว่างของหิ่งห้อยตัวอื่นๆเลย และนำไปสู่การเคลื่อนที่แบบสุ่ม แต่ในทางปฏิบัติแล้วนั้น ค่าสัมประสิทธิ์การดูดซับแสง จะมีค่าอยู่ในช่วงที่จำกัด เพราะว่าเมื่อค่าสัมประสิทธิ์การดูดซับแสงมีค่าเพิ่มขึ้นจนถึงค่าค่าหนึ่งแล้ว หิ่งห้อยทั้งหมดจะเคลื่อนที่แบบสุ่มเพียงอย่างเดียว



รูปที่ 2.7 กราฟแสดงความสัมพันธ์ระหว่างค่าความดึงดูด และระยะทางระหว่างหิ่งห้อยทั้ง 2 ตัว ที่ค่าสัมประสิทธิ์การดูดซับแสงที่ [0.01,0.15] [21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.5 การเคลื่อนที่เข้าหากันของหิ่งห้อย

จากกฎอุดมคติ ข้อที่ 2 หิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า (x_i) จะเคลื่อนที่เข้าหา หิ่งห้อยตัวที่มีแสงสว่างมากกว่า (x_j) แต่สำหรับกรณีที่ หิ่งห้อย x_i พบว่าไม่มีหิ่งห้อยตัวใด ที่มีแสงสว่างมากกว่าหิ่งห้อย x_i จะเคลื่อนที่แบบสุ่ม การที่หิ่งห้อย x_i จะตัดสินใจเคลื่อนที่แบบใดนั้น เกิดขึ้นได้ 2 กรณี ดังนี้

1. ถ้าค่าความดึงดูดระหว่างหิ่งห้อย x_i และหิ่งห้อย x_j มากกว่า ค่าความเข้มแสงของหิ่งห้อย x_i $(\beta_r > I_i)$ หิ่งห้อย x_iจะเคลื่อนที่เข้าหา หิ่งห้อย x_j
2. ถ้าค่าความดึงดูดระหว่างหิ่งห้อย x_i และหิ่งห้อย x_j มากกว่า ค่าความเข้มแสงของหิ่งห้อย x_i $(\beta_r \leq I_i)$ หิ่งห้อย x_i จะเคลื่อนที่แบบสุ่ม

2.4.1.6 แผนการเคลื่อนที่แบบวิธีวิวัฒนาการอัลกอริทึมหิ่งห้อยสำหรับปัญหาไม่ต่อเนื่อง

การตัดสินใจในการเคลื่อนที่ของหิ่งห้อย x_i ในกรณีที่ 1 นั้น เมื่อหิ่งห้อย x_i เคลื่อนที่เข้าหาหิ่งห้อย x_j จะเคลื่อนที่โดยใช้แผนการเคลื่อนที่แบบ วิธีวิวัฒนาการอัลกอริทึมหิ่งห้อยสำหรับปัญหาไม่ต่อเนื่อง (Evolutionary Discrete Firefly Algorithm, EDFA) ซึ่งรูปแบบการเคลื่อนที่แบบนี้ ได้น้อมนำแนวความคิดจากกลยุทธ์การวิวัฒนาการ (Evolution Strategy) มาประยุกต์ โดยเลือกวิธีการพัฒนาปรับปรุงคำตอบแบบ การกลายพันธุ์แบบอินเวอร์ชัน (Inversion Mutation) มาใช้ ฉะนั้นการเคลื่อนที่ของหิ่งห้อย x_i ไปยัง x_j จะเป็นไปดังสมการที่ 10

$$x_i = random(2, r_{ij}) \tag{10}$$

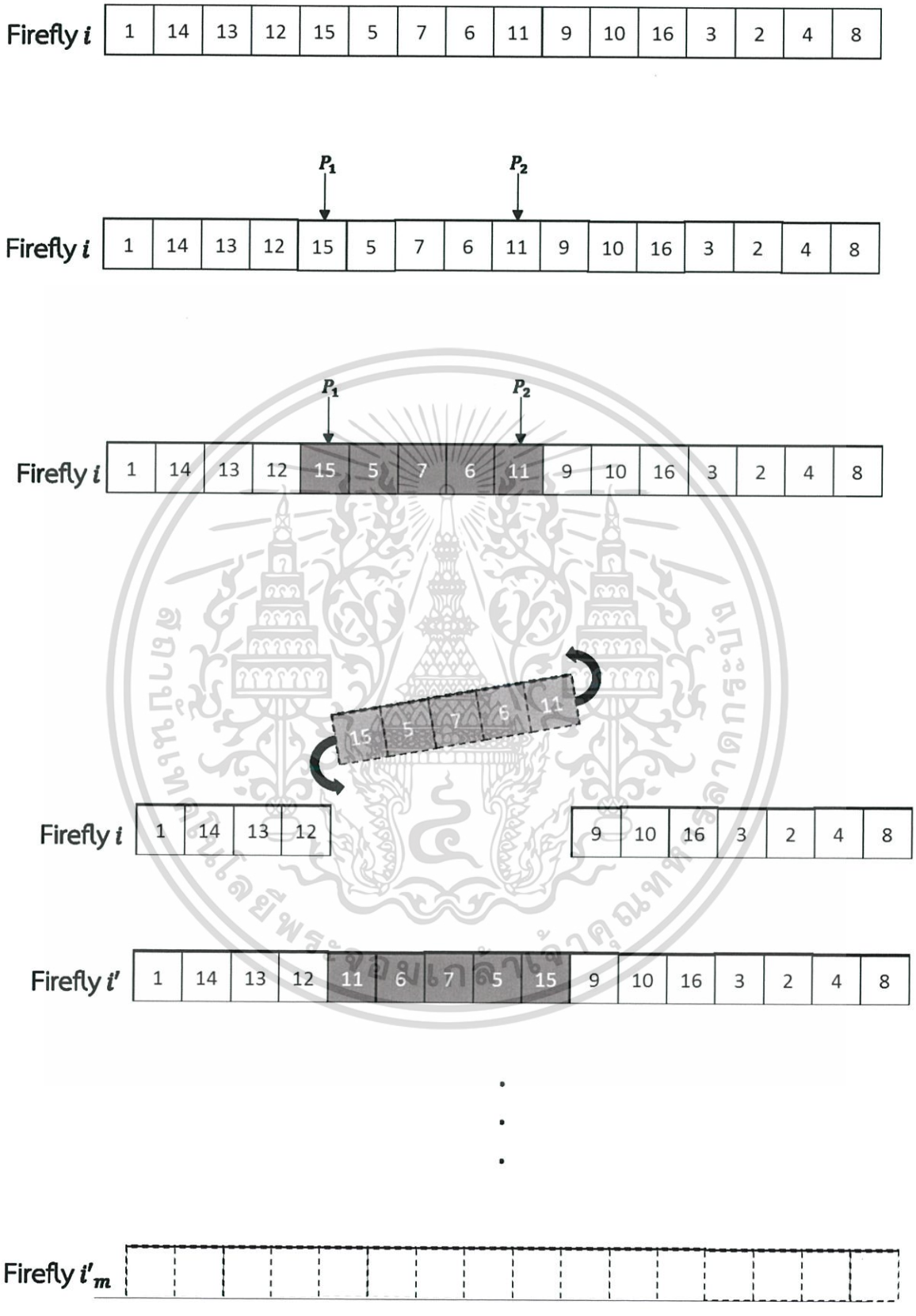
เมื่อ r_{ij} คือระยะห่างระหว่างหิ่งห้อย x_i และหิ่งห้อย x_j และระยะทางในการเคลื่อนหิ่งห้อย x_i เข้าหาหิ่งห้อย x_j จะถูกเลือกแบบสุ่มในช่วง 2 ถึง r_{ij}

เมื่อหิ่งห้อย x_i เคลื่อนที่ ผลลัพธ์ของคำตอบที่ได้จะเกิดการเปลี่ยนแปลง กล่าวคือ เมื่อหิ่งห้อยตัวหนึ่งเคลื่อนที่ ระยะทางรวมที่พนักงานขายต้องเดินทางไปยังเมืองต่างๆให้ครบทุกเมืองนั้น จะเกิดการเปลี่ยนแปลง และการเคลื่อนที่ของหิ่งห้อยนี้จะใช้วิธีการกลายพันธุ์แบบอินเวอร์ชันในการเคลื่อนที่ โดยที่ P_1 เป็นตำแหน่งที่ถูกเลือกมาแบบสุ่ม และตำแหน่ง P_2 คือการนำค่าที่ได้จากสมการที่ 10 มาบวกกับตำแหน่งที่ P_1 จากนั้นเมืองต่างๆที่ถูกเลือกในช่วงตำแหน่ง P_1ถึง P_2 จะถูกนำไปผ่านกระบวนการ การกลายพันธุ์แบบอินเวอร์ชัน การเลือกตำแหน่ง P_1 และ P_2 แบบนี้นั้น จะเกิดขึ้นสำหรับกรณีที่หิ่งห้อย x_i เคลื่อนที่เข้าหา หิ่งห้อย x_j เท่านั้น ส่วนสำหรับกรณีที่หิ่งห้อย x_i เกิดการเคลื่อนที่แบบสุ่ม ตำแหน่ง P_1 และ P_2 จะถูกเลือกแบบสุ่มทั้ง 2 ตำแหน่ง การเคลื่อนที่ของหิ่งห้อยด้วยวิธีการกลายพันธุ์แบบอินเวอร์ชันจำนวน m ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแผนการเคลื่อนที่แบบ EDFA นี้ เริ่มต้นโดย อันดับแรกหิ่งห้อยแต่ละตัวจะสร้างคำตอบเริ่มต้นขึ้นมาแบบสุ่ม จากนั้นหิ่งห้อยตัวหนึ่งๆ จะทำการพิจารณาว่ามีหิ่งห้อยตัวอื่นๆที่มีแสงสว่างมากกว่าตัวมันเองหรือไม่ ถ้ามีหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า จะทำการเคลื่อนที่เข้าหิ่งห้อยตัวที่มีแสงสว่างมากกว่า ถ้าหากเมื่อพิจารณาแล้ว หิ่งห้อยตัวหนึ่งๆนั้น ไม่พบว่ามีหิ่งห้อยตัวที่มีแสงสว่างมากกว่าตัวมันเอง มันจะเคลื่อนที่แบบสุ่ม และเมื่อหิ่งห้อยแต่ละตัวเกิดการเคลื่อนที่ การเคลื่อนที่ของหิ่งห้อยนั้น จะเป็นไปตามวิธีการกลายพันธุ์แบบอินเวอร์ชัน โดยจะเคลื่อนที่เป็นจำนวน m ครั้ง ดังนั้นเมื่อหิ่งห้อยจำนวน n ตัว เคลื่อนที่ด้วยวิธีการกลายพันธุ์แบบอินเวอร์ชั่น จะทำให้เกิดจำนวนคำตอบขึ้นทั้งหมด $m \times n$ คำตอบ และในท้ายที่สุด เมื่อทุกกระบวนการในการหาคำตอบรอบแรกเสร็จสิ้นเรียบร้อยแล้ว จำนวนผลคำตอบที่ได้ จะเท่ากับ $m \times n + 1$ จากนั้นหิ่งห้อยจำนวน n ตัว จะถูกเลือกไปเป็นคำตอบเริ่มต้นสำหรับสำหรับรอบการหาคำตอบถัดไป การกลายพันธุ์แบบอินเวอร์ชัน ที่นำมาใช้ในอัลกอริทึมหิ่งห้อย สำหรับปัญหา TSP ถูกแสดงไว้ในรูปที่ 2.7 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การกลายพันธุ์แบบอินเวอร์ชัน ที่นำมาใช้สำหรับการเคลื่อนที่ของหิ่งห้อย

เอกสารนี้เป็น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ด้านการค้า

2.4.2 แผนการเคลื่อนที่แบบใหม่ของอัลกอริทึมหิ่งห้อย สำหรับปัญหาการเดินทางของพนักงานขาย

เนื่องจากในแผนการเคลื่อนที่แบบ EDFA หิ่งห้อยแต่ละตัวไม่มีทิศทางในการเคลื่อนที่ จึงจำเป็นต้องใช้การกลายพันธุ์แบบอินเวอร์ชันในการเคลื่อนที่ แต่ทว่า การเคลื่อนที่โดยการใช้การกลายพันธุ์แบบอินเวอร์ชันนี้ จะไม่สามารถการันตีได้ว่า ทุกครั้งที่เกิดการเคลื่อนที่ของหิ่งห้อย ระยะทางระหว่างหิ่งห้อยทั้ง 2 ตัวจะลดลง หรือสามารถกล่าวได้อีกนัยหนึ่งว่า เมื่อหิ่งห้อยเคลื่อนที่ด้วยวิธีการกลายพันธุ์แบบอินเวอร์ชัน ไม่สามารถทำให้ผลลัพธ์ของคำตอบ คือระยะทางรวมทั้งหมดที่พนักงานขายต้องเดินทางไปให้ครบทุกเมืองนั้นลดลงได้ในทุกๆครั้งของการปรับปรุงคำตอบ ดังนั้น Gilang Kusuma Jati, Ruli Manurung และ Suyanto จึงได้พัฒนาวิธีการเคลื่อนที่เข้าหากันของหิ่งห้อยทั้ง 2 ตัวขึ้น [22]

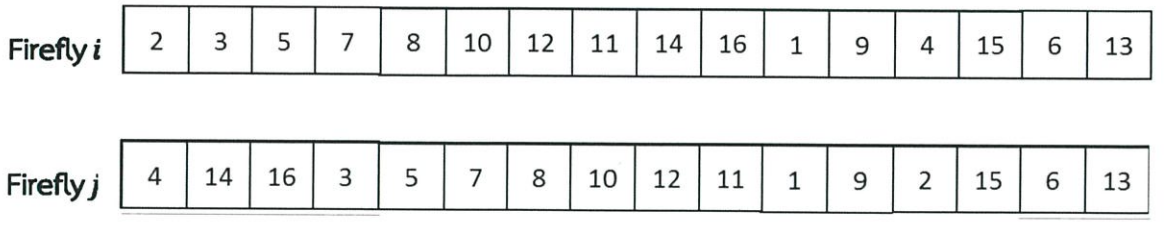
ในแผนการเคลื่อนที่แบบใหม่ของอัลกอริทึมหิ่งห้อย สำหรับปัญหาการเดินทางของพนักงานขาย (Discrete Firefly Algorithm for Taveling Salesman Problem: A New Movement Scheme) ได้นำเสนอวิธีการที่ใช้ในการเคลื่อนที่เข้าหากันของหิ่งห้อยทั้ง 2 ตัว โดย วิธีการเคลื่อนที่แบบคู่ขอบ (Edge-Based Movement Scheme) ซึ่งวิธีการนี้จะแตกต่างจากวิธีการเคลื่อนที่แบบ EDFA ในส่วนการเคลื่อนที่ของหิ่งห้อย x_i ไปยัง หิ่งห้อย x_j แต่การเคลื่อนที่หิ่งห้อยแบบสุ่มในกรณีที่หิ่งห้อยตัวนั้น พบว่าไม่มีหิ่งห้อยตัวใดเลยที่มีแสงสว่างมากกว่าตัวมันเองนั้น ยังคงใช้วิธีการเคลื่อนที่ของการกลายพันธุ์แบบอินเวอร์ชันดังเช่นในวิธี EDFA ดังเดิม

วิธีการเคลื่อนที่แบบคู่ขอบ ได้นำเสนอวิธีการที่สามารถการันตีได้ว่า ทุกๆครั้งที่หิ่งห้อย x_i เคลื่อนที่ไปยัง หิ่งห้อย x_j ระยะห่างระหว่างหิ่งห้อยทั้ง 2 นั้นจะลดลง ซึ่งหมายความว่าผลลัพธ์ที่ได้จากการเคลื่อนที่เข้าหากันของหิ่งห้อย สามารถทำได้โดยนำเอาคู่ขอบที่มีอยู่ในหิ่งห้อยตัวที่มีแสงสว่างมากกว่า (x_j) แต่ไม่มีอยู่ในหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า (x_i) มาใส่ในหิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า (x_i) การทำแบบนี้ จะทำให้ทุกๆครั้งที่มีการปรับปรุงคำตอบ คำตอบที่เกิดขึ้นใหม่ จะไม่เป็นคำตอบที่ให้ผลลัพธ์ที่ด้อยกว่าผลลัพธ์ดั้งเดิมของ x_i

การเคลื่อนที่ด้วยวิธีการเคลื่อนที่แบบคู่ขอบนี้ จะใช้เมื่อหิ่งห้อย x_i ตัดสินใจเคลื่อนที่เข้าหาหิ่งห้อย x_j โดยมีขั้นตอนดังต่อไปนี้

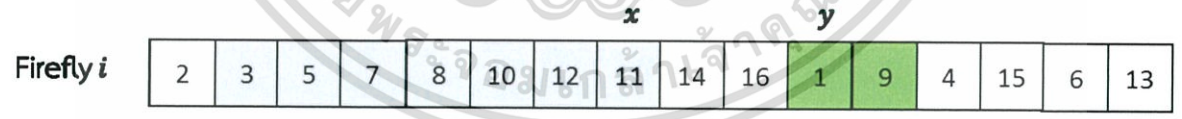
1. พิจารณาคู่ขอบที่แตกต่างกันทั้งหมดของหิ่งห้อยทั้ง 2 ตัว จากรูปที่ 2.8 พบว่า คู่ขอบที่มีอยู่ในหิ่งห้อย x_i แต่ไม่มีอยู่ในหิ่งห้อย x_j ได้แก่ 14-16, 3-5, 11-1, 9-2, 2-15 และ 13-4 ส่วนคู่ขอบที่มีอยู่ในหิ่งห้อย x_i และมีอยู่ในหิ่งห้อย x_j ด้วย ได้แก่ 14-16, 3-5, 5-7, 7-8, 8-10, 10-12, 12-11, 1-9, 15-6 และ 6-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การเปรียบเทียบหาคู่ขอบที่แตกต่างกันของหิ่งห้อยทั้ง 2 ตัว

2. จากจำนวนคู่ขอบที่แตกต่างกันทั้งหมด 6 คู่ขอบ ให้ทำการสุ่มเลือกคู่ขอบขึ้นมา 1 คู่ขอบ เพื่อการนำไปปรับปรุงคำตอบ ในตัวอย่างนี้ สมมติให้เลือกคู่ขอบ 11-1 เป็นคู่ขอบที่จะนำไปใส่ในหิ่งห้อย x_i เพื่อพัฒนาปรับปรุงคำตอบ โดยกำหนดให้เมือง 11 ในหิ่งห้อย x_i คือตำแหน่ง x และ เมือง 1 คือตำแหน่ง y จากนั้นให้พิจารณาที่ตำแหน่ง x เพื่อทำการยืดส่วน (Segment) ออกไป ซึ่งสามารถทำได้โดยให้พิจารณาเมืองข้างเคียงกับเมือง 11 ว่าคู่ขอบใดๆบ้าง ที่มีอยู่ในทั้งหิ่งห้อย x_i และหิ่งห้อย x_j (ได้แก่ 14-16, 3-5, 5-7, 7-8, 8-10, 10-12, 12-11, 1-9, 15-6 และ 6-13) อยู่ในบริเวณข้างเคียงหรือไม่ ถ้าหากพบว่ามี ให้ทำการยืดส่วน x ออกไป สำหรับการยืดส่วนของตำแหน่ง y ให้ทำเช่นเดียวกัน ดังแสดงในรูปที่ 2.9 การยืดส่วนออกไปแบบนี้ เป็นขั้นตอนที่ทำให้หิ่งห้อย x_i สามารถรักษาส่วนของคำตอบที่ดีอยู่แล้วเอาไว้ได้ ซึ่งการปฏิบัติการเช่นนี้ ประกอบกับการเลือกคู่ขอบ 11-1 ซึ่งเป็นคู่ขอบที่มีอยู่ในตัวหิ่งห้อย x_j แต่ไม่มีอยู่ในหิ่งห้อย x_i จะทำให้แผนการเคลื่อนที่แบบใหม่นี้ สามารถการันตีได้ว่า ในทุกๆครั้งของการปรับปรุงคำตอบ ผลลัพธ์ใหม่ที่ได้ออกมา จะไม่เป็นผลลัพธ์ที่ด้อยกว่าเดิมนั่นเอง



รูปที่ 2.10 การยืดส่วนตำแหน่ง x และ y

3. จากนั้นนำส่วน x และส่วน y มาทำการสลับตำแหน่งกัน เพื่อให้การนำคู่ขอบ 11-1 ซึ่งเป็นคู่ขอบที่มีอยู่ในหิ่งห้อย x_j แต่ไม่มีอยู่ในหิ่งห้อย x_i ใส่ลงไปในหิ่งห้อย x_i ให้สมบูรณ์ โดยขั้นตอนการสลับส่วนนั้น ต้องทำด้วยกันทั้งหมด 4 วิธี ได้แก่

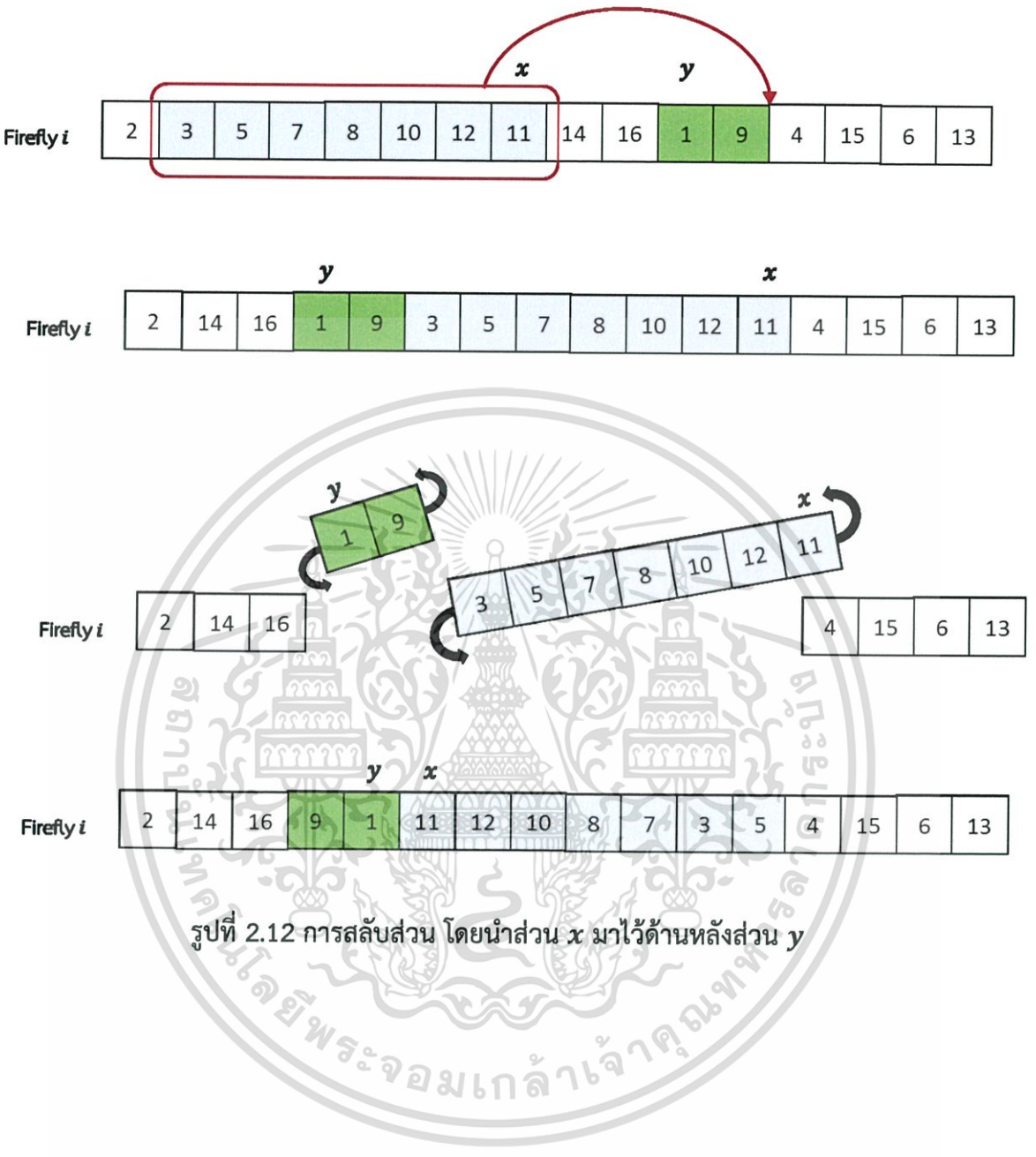
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีที่ 1 คือ การสลับตำแหน่งโดยการนำส่วน x มาไว้ด้านหน้าส่วน y
วิธีที่ 2 คือ การสลับตำแหน่งโดยการนำส่วน x มาไว้ด้านหลังส่วน y
วิธีที่ 3 คือ การสลับตำแหน่งโดยการนำส่วน y มาไว้ด้านหน้าส่วน x
วิธีที่ 4 คือ การสลับตำแหน่งโดยการนำส่วน y มาไว้ด้านหลังส่วน x

จุดประสงค์ของการสลับส่วนนั้นก็เพื่อ นำคู่ขอบที่ถูกเลือก (ในตัวอย่างนี้คือคู่ขอบ 11-1) มาใส่ลงในหิ่งห้อย x_i ให้สมบูรณ์ ดังนั้นทุกๆการสลับคู่ขอบ จำเป็นอย่างยิ่งที่จะต้องสังเกตว่า เมือง 11 และ 1 หรือตำแหน่ง x และ y อยู่ติดกันแล้วหรือไม่ หากพบว่า เมื่อสลับส่วนเรียบร้อยแล้ว ตำแหน่ง x และ y ยังไม่อยู่ติดกัน ให้ทำการกลับด้านส่วน x และ y ทั้งสองส่วน เพื่อให้ตำแหน่ง x และ y มาอยู่ติดกันในท้ายที่สุด

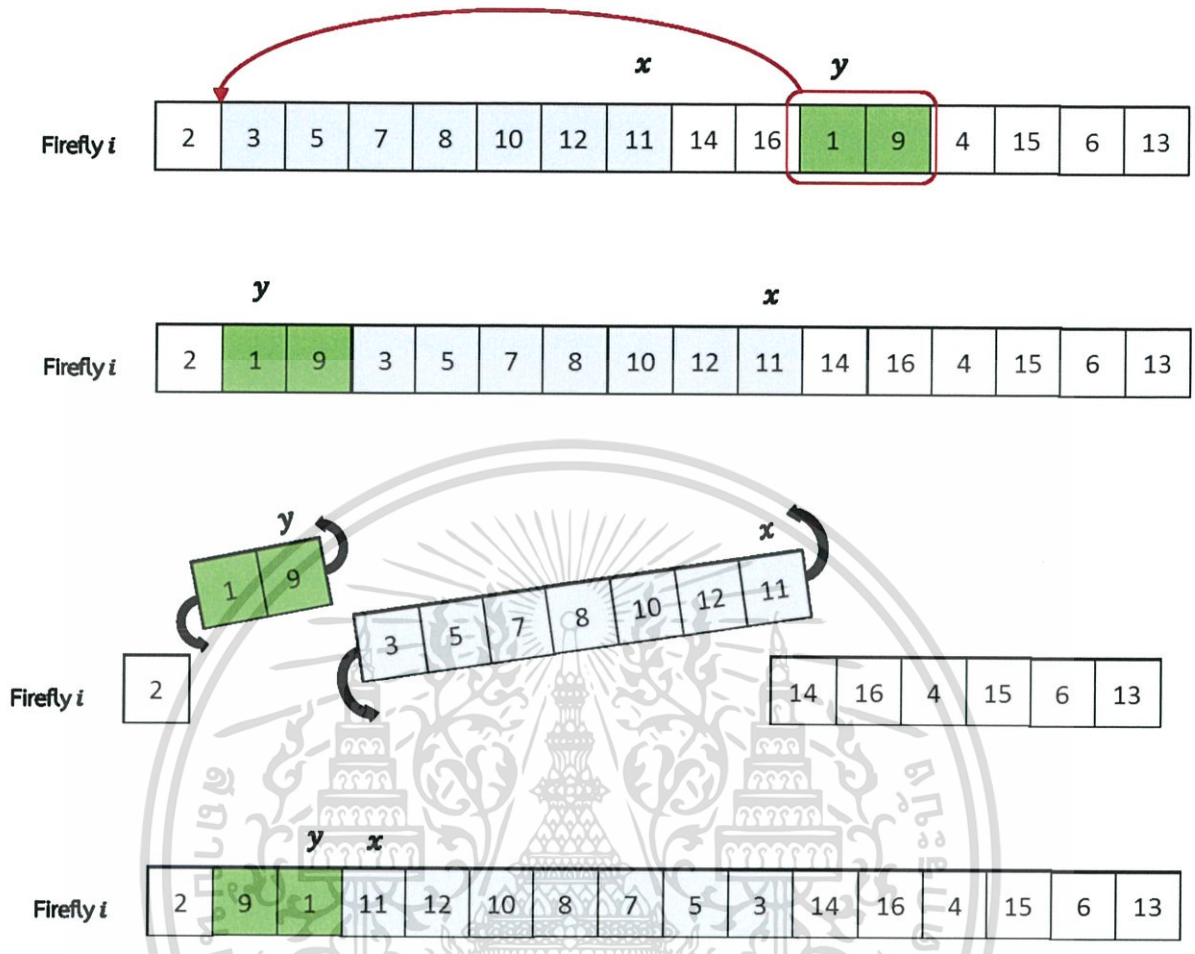
รูปที่ 2.11 การสลับส่วน โดยนำส่วน x มาไว้ด้านหน้าส่วน y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

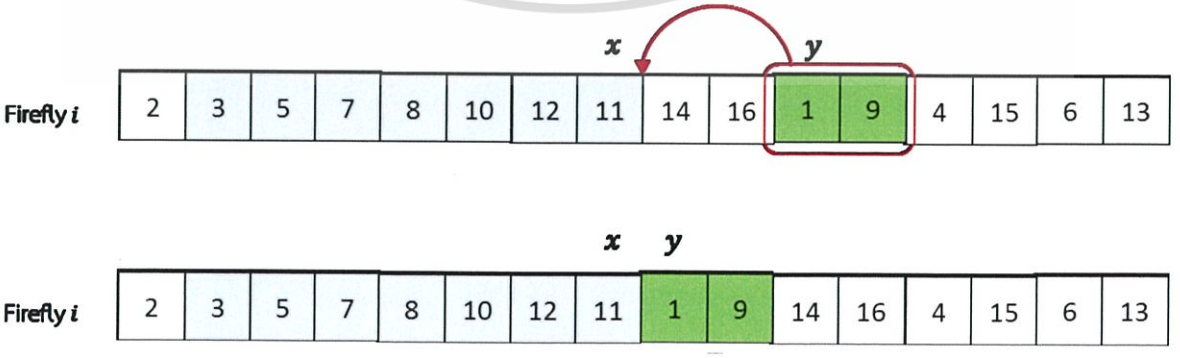


รูปที่ 2.12 การสลับส่วน โดยนำส่วน x มาไว้ด้านหลังส่วน y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การสลับส่วน โดยนำส่วน y มาไว้ด้านหน้าส่วน x

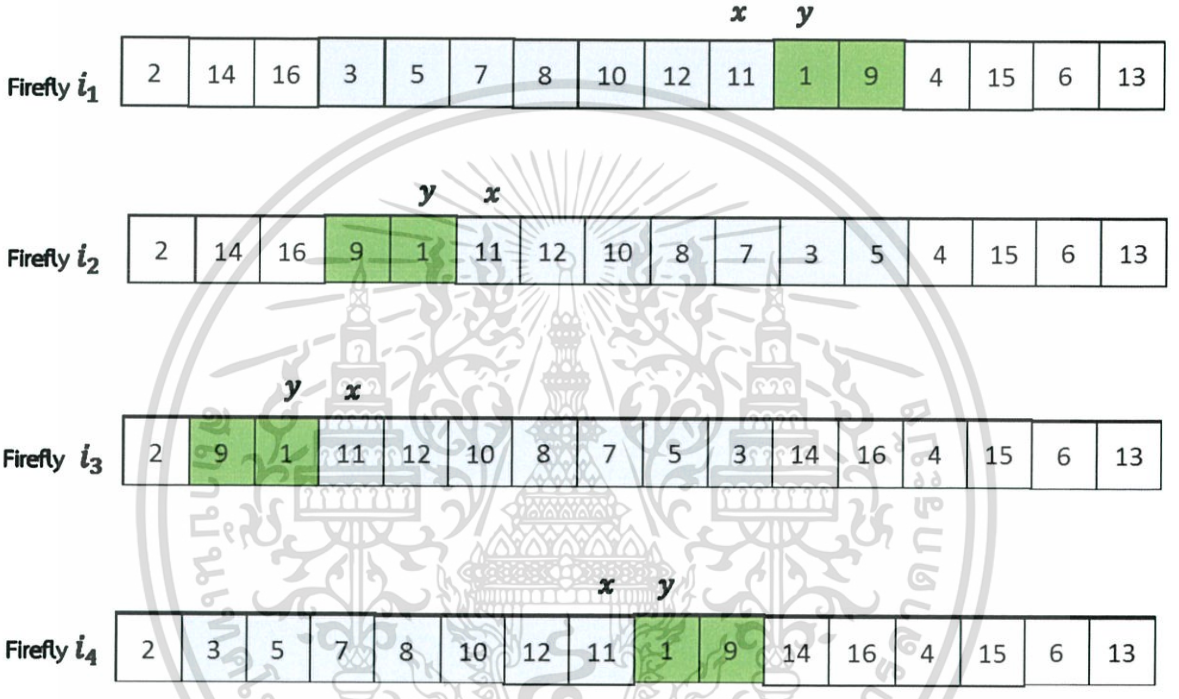


รูปที่ 2.14 การสลับส่วน โดยนำส่วน y มาไว้ด้านหลังส่วน x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อทำการสลับส่วนครบทั้ง 4 วิธีแล้ว ผลลัพธ์ที่ได้จากการสลับที่เกิดขึ้นทั้งหมดเท่ากับ 4 คำตอบ กล่าวคือทุกๆครั้งที่หิ่งห้อยเคลื่อนที่ด้วยวิธีการเคลื่อนที่แบบคู่ขอบ จะก่อให้เกิดคำตอบใหม่ขึ้นทั้งหมด 4 คำตอบด้วยกัน ได้แก่ i_1, i_2, i_3 และ i_4 ดังรูปที่ 2.14

ค



รูปที่ 2.15 คำตอบทั้ง 4 ที่ได้จากการเคลื่อนที่แบบคู่ขอบ

เนื่องจากปัญหา TSP เป็นปัญหาที่ไม่ต่อเนื่อง ดังนั้นการที่จะนำอัลกอริทึมหิ่งห้อยมาประยุกต์ใช้ในการหาคำตอบของปัญหา TSP นั้น จึงจำเป็นที่จะต้องทำให้อัลกอริทึมหิ่งห้อยเหมาะสำหรับแก้ปัญหาที่ไม่ต่อเนื่องเสียก่อน และทุกๆครั้งที่หิ่งห้อยตัวหนึ่งเกิดการเคลื่อนที่ จะทำให้ผลของคำตอบเกิดการเปลี่ยนแปลง การเคลื่อนที่ของหิ่งห้อยเกิดขึ้นได้ 2 แบบ คือ หิ่งห้อยตัวที่มีแสงสว่างน้อยกว่า เคลื่อนที่เข้าหาหิ่งห้อยตัวที่มีแสงสว่างมากกว่า โดยใช้วิธีการเคลื่อนที่แบบคู่ขอบ และสำหรับกรณีที่หิ่งห้อยตัวที่พิจารณา ไม่พบว่ามีหิ่งห้อยตัวอื่นที่ตัวอื่นที่สว่างกว่าตัวมันเอง มันจะเคลื่อนที่แบบสุ่มด้วยวิธีการกลายพันธุ์แบบอินเวอร์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

ขั้นตอนการดำเนินงานในปริญญานิพนธ์บทนี้ จะใช้ข้อมูลตัวอย่างปัญหาการเดินทางของพนักงาน (Traveling Salesman Problem, TSP) จำนวน 3 ขนาดด้วยกัน จากปัญหาตัวอย่างใน TSPLIB ได้แก่ ปัญหา TSP ขนาด 16 เมือง, ปัญหา TSP ขนาด 22 เมือง และปัญหา TSP ขนาด 52 เมือง จากนั้นจะทำการหาเส้นทางการเดินทางของพนักงานขายที่เหมาะสม โดยใช้อัลกอริทึมหิ่งห้อย (Firefly Algorithm, FA) มาประยุกต์ใช้ โดยผลลัพธ์ที่ได้จะถูกนำไปเปรียบเทียบสมรรถนะในการค้นหาคำตอบ กับวิธีฮิวริสติก, วิธีเชิงพันธุกรรม (Genetic Algorithm, GA) รวมทั้งวิธีที่ให้คำตอบที่เหมาะสม (Optimal Solution Method)

3.1 ศึกษาและรวบรวมข้อมูล

ทางกลุ่มผู้วิจัยได้ทำการศึกษา และรวบรวมข้อมูลที่เกี่ยวข้องกับปัญหา TSP โดยชุดข้อมูลตัวอย่างที่นำมาศึกษามี 3 ชุด ได้แก่ ชุดข้อมูล ulysses 16, ulysses 22 และ berlin 52 ที่มีจำนวนเมือง 16, 22 และ 52 เมืองตามลำดับ ดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ข้อมูลพิกัดตำแหน่งของเมือง 16 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
1	38.24	20.42
2	39.57	26.15
3	40.56	25.32
4	36.26	23.12
5	33.48	10.54
6	37.56	12.19
7	38.42	13.11
8	37.52	20.44
9	41.23	9.1
10	41.17	13.05
11	36.08	-5.21
12	38.47	15.13
13	38.15	15.35
14	37.51	15.17
15	35.49	14.32
16	39.36	19.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ข้อมูลพิกัดตำแหน่งของเมือง 22 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
1	38.24	20.42
2	39.57	26.15
3	40.56	25.32
4	36.26	23.12
5	33.48	10.54
6	37.56	12.19
7	38.42	13.11
8	37.52	20.44
9	41.23	9.1
10	41.17	13.05
11	36.08	-5.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2(ต่อ) ข้อมูลพิกัดตำแหน่งของเมือง 22 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
12	38.47	15.13
13	38.15	15.35
14	37.51	15.17
15	35.49	14.32
16	39.36	19.56
17	38.09	24.36
18	36.09	23
19	40.44	13.57
20	40.33	14.15
21	40.37	14.23
22	37.57	22.56

ตารางที่ 3.3 ข้อมูลพิกัดตำแหน่งของเมือง 52 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
1	565	575
2	25	185
3	345	750
4	945	685
5	845	655
6	880	660
7	25	230
8	525	1000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3(ต่อ) ข้อมูลพิกัดตำแหน่งของเมือง 52 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
9	580	1175
10	650	1130
11	1605	620
12	1220	580
13	1465	200
14	1530	5
18	415	635
19	510	875
20	560	365
21	300	465
22	520	585
23	480	415
24	835	625
25	975	580
26	1215	245
27	1320	315
28	1250	400
29	660	180
30	410	250
31	420	555
32	575	665
33	1150	1160
34	700	580

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3(ต่อ) ข้อมูลพิกัดตำแหน่งของเมือง 52 เมือง

ลำดับที่	ตำแหน่งที่ตั้งของเมือง	
	x	y
35	685	595
36	685	610
37	770	619
38	795	645
39	720	635
40	760	650
41	475	960
42	95	260
43	875	920
44	700	500
45	555	815
46	830	485
47	1170	65
48	830	610
49	605	625
50	595	360
51	1340	725
52	1740	245

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบเส้นทางการเดินทางของพนักงานขายโดยการประยุกต์ใช้วิธีการเชิงพันธุกรรม

วิธีการเชิงพันธุกรรม เป็นเทคนิคสำหรับการค้นหาคำตอบของปัญหา โดยอาศัยหลักการจากทฤษฎีวิวัฒนาการจากธรรมชาตินั่นคือ สิ่งมีชีวิตที่เหมาะสมที่สุดจะสามารถอยู่รอดได้เนื่องจากผ่านกระบวนการคัดเลือก เปลี่ยนแปลง และพัฒนาปรับปรุง ให้มีความเหมาะสมมากขึ้น ซึ่งเป็นผลทำให้ได้คำตอบที่ดีขึ้นเรื่อยๆ ค่าพารามิเตอร์ต่างๆของ GA ที่จะใช้ทดสอบกับชุดปัญหาที่นำมาศึกษาดังตารางต่อไปนี้

ตารางที่ 3.4 ตารางแสดงค่าพารามิเตอร์สำหรับวิธีการเชิงพันธุกรรม [27]

ค่า	Crossover rate	Mutaion rate
1	0.3	0.05
2	0.6	0.1
3	0.9	0.2

ขั้นตอนวิธีการเชิงพันธุกรรมสำหรับปัญหาการเดินทางของพนักงานขายเป็นดังนี้

1. การรับข้อมูล (Data Input) ขั้นตอนนี้เป็นการรับข้อมูล ที่จำเป็นสำหรับการนำไปใช้ในการคำนวณ เพื่อกำหนดเส้นทางการเดินทางของพนักงานขาย เช่น ลำดับเมืองที่พนักงานขายต้องเดินทางไป

2. การสร้างประชากรเริ่มต้น (Representation and Initialization) เป็นการนำข้อมูลที่เก็บไว้มาทำการสร้างคำตอบเบื้องต้นแบบสุ่ม โดยผ่านกระบวนการเข้ารหัสคำตอบ (Encoding) และการสร้างจำนวนประชากรเบื้องต้น

3. การถอดรหัส (Decoding) เป็นกระบวนการที่นำรหัสคำตอบของประชากรเบื้องต้นทุกตัว มาทำการถอดรหัสคำตอบเพื่อให้ได้คำตอบที่สมบูรณ์

4. การประเมินค่า (Evaluation) เป็นการนำคำตอบมาประเมิน โดยวัดจากค่าความเหมาะสมที่ได้กำหนดขึ้นไว้สำหรับเป็นตัวชี้วัด และเปรียบเทียบกับคำตอบอื่นๆ

5. การคงค่าคำตอบที่ดีที่สุดเบื้องต้น (Initial Elite Preserve Strategy) เป็นการหาคำตอบที่ดีที่สุดจากกลุ่มคำตอบประชากรชุดแรก และทำการเก็บรักษาค่าคำตอบที่ดีที่สุดนี้ไว้ จนกว่าจะมีค่าคำตอบที่ดีกว่าเกิดขึ้น (Elite Preserve Solution)

6. การคัดเลือก (Selection) เป็นขั้นตอนในการคัดเลือกคำตอบที่ดีที่สุดเข้าสู่เมทติ้ง พูล (Mating Pool) เพื่อทำการจับคู่โดยอาศัยวิธีการคัดเลือก เพื่อหาคำตอบที่มีความเหมาะสมมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. กระบวนการครอสโอเวอร์ (Crossover) เป็นการจับคู่คำตอบที่อยู่ในเมทติ้ง พูล และทำการครอสโอเวอร์ ภายใต้ความน่าจะเป็นในการครอสโอเวอร์ที่ถูกกำหนดขึ้น

8. การเก็บค่าคำตอบที่ดีที่สุดหลังจากผ่านกระบวนการครอสโอเวอร์ (Post-crossover Elite Preserve Strategy) เป็นขั้นตอนในการค้นหาคำตอบที่ดีที่สุด ที่ได้จากการครอสโอเวอร์ จากนั้นนำมาเปรียบเทียบกับค่าคำตอบที่ดีที่สุดเบื้องต้นที่เก็บรักษาไว้ก่อนหน้านี้ ถ้าคำตอบที่ได้จากการครอสโอเวอร์ดีกว่า ให้ทำการเก็บค่าคำตอบนั้น เพื่อเป็นค่าคำตอบที่ดีที่สุดเบื้องต้นแทนค่าคำตอบที่ดีที่สุดเบื้องต้นอันเดิม

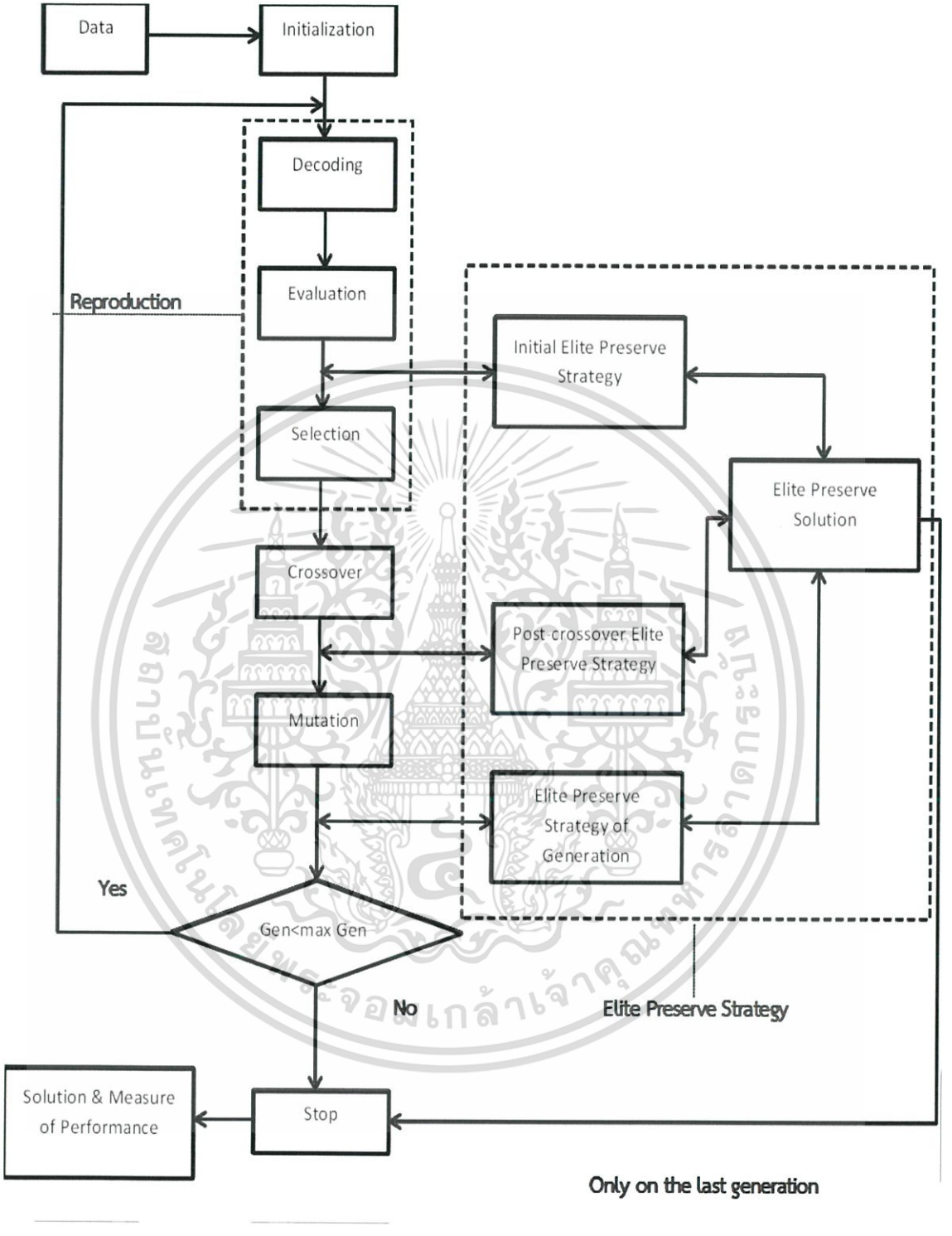
9. กระบวนการมิวเทชัน (Mutation) คือการหาสตริง (String) คำตอบที่มี ภายใต้ความน่าจะเป็นในการเกิดกระบวนการมิวเทชันที่กำหนดขึ้น โดยการสลับตำแหน่งของค่าภายในสตริงคำตอบตัวเดียว

10. การเก็บค่าคำตอบที่ดีที่สุดเบื้องต้นจากกระบวนการมิวเทชัน (Elite Preserve Strategy of Generation) เป็นกระบวนการหาค่าคำตอบที่ดีที่สุดจากการมิวเทชัน แล้วนำมาเปรียบเทียบกับค่าคำตอบเบื้องต้นที่มีอยู่ ถ้าคำตอบที่ได้จากการผ่านกระบวนการมิวเทชันมีค่าที่ดีกว่า ให้ทำการเก็บคำตอบนั้น เป็นค่าคำตอบที่ดีที่สุดเบื้องต้นอันใหม่แทน แต่ถ้าหากว่า เมื่อทำการเปรียบเทียบแล้ว ปรากฏว่าค่าคำตอบที่ดีที่สุดเบื้องต้นที่มีอยู่นั้น เป็นค่าคำตอบที่ดีกว่า ให้นำค่าคำตอบที่ดีที่สุดเบื้องต้นนั้น ไปแทนที่คำตอบที่แย่ที่สุดจากการทำกระบวนการมิวเตชั่น

11. จำนวนรอบของการทำงาน (Gas-loop) เป็นกระบวนการที่ใช้ในการตรวจสอบว่า จำนวนเจเนอเรชัน (Generation) ที่เกิดขึ้นล่าสุดนั้น มีจำนวนเท่ากับจำนวนเจเนอเรชั่นสูงสุดที่ได้กำหนดไว้หรือไม่ ถ้าพบว่าจำนวนเจเนอเรชั่นล่าสุดในปัจจุบัน มีจำนวนน้อยกว่าจำนวนเจเนอเรชั่นสูงสุดที่กำหนดไว้ ให้กลับไปทำซ้ำที่ขั้นตอนที่ 3 ถึง ขั้นตอนที่ 11 อีกครั้ง ในกรณีที่จำนวนเจเนอเรชั่นล่าสุดที่เกิดขึ้น มีจำนวนเท่ากับเจเนอเรชั่นสูงสุดที่ได้กำหนดไว้ ให้ดำเนินการต่อตามขั้นตอนที่ 12

12. ขั้นตอนการหยุด (Stop) เป็นกระบวนการที่เกิดขึ้น เมื่อสิ้นสุดทุกขั้นตอนของกระบวนการปรับปรุงคำตอบด้วยวิธีการเชิงพันธุกรรม และคำตอบที่ดีที่สุดที่เก็บไว้ในรอบการปรับปรุงคำตอบล่าสุด เพื่อนำมาเป็นคำตอบของปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ผังงานแสดงกระบวนการของวิธีการเชิงพันธุกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบเส้นทางการเดินทางของพนักงานขายโดยการประยุกต์ใช้วิธีอัลกอริทึมหิ่งห้อย

การนำอัลกอริที่มหิ่งห้อยมาประยุกต์ใช้กับปัญหา TSP พบว่ามีค่าพารามิเตอร์ที่สำคัญด้วยกัน 4 ค่า ได้แก่ จำนวนประชากรหิ่งห้อยเริ่มต้น (n), ค่าสัมประสิทธิ์การดูดซับแสง (γ), จำนวนครั้งการเกิดกระบวนการกลายพันธุ์แบบอินเวอร์ชัน (m) และจำนวนรอบทั้งหมดในการปรับปรุงคำตอบ (t) โดยค่าพารามิเตอร์ต่างๆที่ใช้ในปริญญานิพนธ์ฉบับนี้ มีดังต่อไปนี้

ตารางที่ 3.5 ตารางแสดงค่าพารามิเตอร์สำหรับอัลกอริทึมหิ่งห้อย

ค่า	n	m	γ	t
1	5	10	0.0001	100
2	10	100	0.001	500
3	20	500	0.01	800

การทดลองที่เกิดขึ้นในปริญญานิพนธ์ฉบับนี้ จะเป็นการนำค่าพารามิเตอร์ต่างๆที่ได้กำหนดไว้ในตารางที่ 3.5 มาทำการทดลองซ้ำทั้งหมด 5 ครั้งด้วยกัน เพื่อดูแนวโน้มของคำตอบที่เกิดขึ้น เมื่อใช้ค่าพารามิเตอร์ที่แตกต่างกัน โดยที่ค่าสัมประสิทธิ์การดูดซับแสง (γ) ที่เลือกใช้ จะมีค่าอยู่ในช่วง [0.0001, 0.21] [22] จากปัญหา TSP ที่นำมาศึกษา 3 ชุด ได้แก่ ulysses16, ulysses22 และ berlin52 สามารถจัดทำตารางการทดลองขึ้นได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 10, γ = 0.0001 และ t = 100

ครั้งที่	n	m	γ	t
1	5	10	0.0001	100
2	5	10	0.0001	100
3	5	10	0.0001	100
4	5	10	0.0001	100
5	5	10	0.0001	100
6	10	10	0.0001	100
7	10	10	0.0001	100
8	10	10	0.0001	100
9	10	10	0.0001	100
10	10	10	0.0001	100
11	20	10	0.0001	100
12	20	10	0.0001	100
13	20	10	0.0001	100
14	20	10	0.0001	100
15	20	10	0.0001	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 100, γ = 0.001 และ t = 500

ครั้งที่	n	m	γ	t
1	5	100	0.001	500
2	5	100	0.001	500
3	5	100	0.001	500
4	5	100	0.001	500
5	5	100	0.001	500
6	10	100	0.001	500
7	10	100	0.001	500
8	10	100	0.001	500
9	10	100	0.001	500
10	10	100	0.001	500
11	20	100	0.001	500
12	20	100	0.001	500
13	20	100	0.001	500
14	20	100	0.001	500
15	20	100	0.001	500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ค่าพารามิเตอร์ที่ n = 5, 10 และ 20, m = 500, γ = 0.01 และ t = 800

ครั้งที่	n	m	γ	t
1	5	500	0.01	800
2	5	500	0.01	800
3	5	500	0.01	800
4	5	500	0.01	800
5	5	500	0.01	800
6	10	500	0.01	800
7	10	500	0.01	800
8	10	500	0.01	800
9	10	500	0.01	800
10	10	500	0.01	800
11	20	500	0.01	800
12	20	500	0.01	800
13	20	500	0.01	800
14	20	500	0.01	800
15	20	500	0.01	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 ค่าพารามิเตอร์ที่ n = 5, m = 10, 100 และ 500, γ = 0.0001 และ t = 100

ครั้งที่	n	m	γ	t
1	5	10	0.0001	100
2	5	10	0.0001	100
3	5	10	0.0001	100
4	5	10	0.0001	100
5	5	10	0.0001	100
6	5	100	0.0001	100
7	5	100	0.0001	100
8	5	100	0.0001	100
9	5	100	0.0001	100
10	5	100	0.0001	100
11	5	500	0.0001	100
12	5	500	0.0001	100
13	5	500	0.0001	100
14	5	500	0.0001	100
15	5	500	0.0001	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 ค่าพารามิเตอร์ที่ n = 10, m = 10, 100 และ 500, γ = 0.001 และ t = 500

ครั้งที่	n	m	γ	t
1	10	10	0.001	500
2	10	10	0.001	500
3	10	10	0.001	500
4	10	10	0.001	500
5	10	10	0.001	500
6	10	100	0.001	500
7	10	100	0.001	500
8	10	100	0.001	500
9	10	100	0.001	500
10	10	100	0.001	500
11	10	500	0.001	500
12	10	500	0.001	500
13	10	500	0.001	500
14	10	500	0.001	500
15	10	500	0.001	500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 ค่าพารามิเตอร์ที่ n = 20, m = 10, 100 และ 500, γ = 0.01 และ t = 800

ครั้งที่	n	m	γ	t
1	20	10	0.01	800
2	20	10	0.01	800
3	20	10	0.01	800
4	20	10	0.01	800
5	20	10	0.01	800
6	20	100	0.01	800
7	20	100	0.01	800
8	20	100	0.01	800
9	20	100	0.01	800
10	20	100	0.01	800
11	20	500	0.01	800
12	20	500	0.01	800
13	20	500	0.01	800
14	20	500	0.01	800
15	20	500	0.01	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 ค่าพารามิเตอร์ที่ n = 5, m = 10, γ = 0.0001, 0.001 และ 0.01 และ t = 100

ครั้งที่	n	m	γ	t
1	5	10	0.0001	100
2	5	10	0.0001	100
3	5	10	0.0001	100
4	5	10	0.0001	100
5	5	10	0.0001	100
6	5	10	0.001	100
7	5	10	0.001	100
8	5	10	0.001	100
9	5	10	0.001	100
10	5	10	0.001	100
11	5	10	0.01	100
12	5	10	0.01	100
13	5	10	0.01	100
14	5	10	0.01	100
15	5	10	0.01	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.13 ค่าพารามิเตอร์ที่ n = 10, m = 100, γ = 0.0001, 0.001 และ 0.01 และ t = 500

ครั้งที่	n	m	γ	t
1	10	100	0.0001	500
2	10	100	0.0001	500
3	10	100	0.0001	500
4	10	100	0.0001	500
5	10	100	0.0001	500
6	10	100	0.001	500
7	10	100	0.001	500
8	10	100	0.001	500
9	10	100	0.001	500
10	10	100	0.001	500
11	10	100	0.01	500
12	10	100	0.01	500
13	10	100	0.01	500
14	10	100	0.01	500
15	10	100	0.01	500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 ค่าพารามิเตอร์ที่ n = 20, m = 500, γ = 0.0001, 0.001 และ 0.01 และ t = 800

ครั้งที่	n	m	γ	t
1	20	500	0.0001	800
2	20	500	0.0001	800
3	20	500	0.0001	800
4	20	500	0.0001	800
5	20	500	0.0001	800
6	20	500	0.001	800
7	20	500	0.001	800
8	20	500	0.001	800
9	20	500	0.001	800
10	20	500	0.001	800
11	20	500	0.01	800
12	20	500	0.01	800
13	20	500	0.01	800
14	20	500	0.01	800
15	20	500	0.01	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.15 ค่าพารามิเตอร์ที่ n = 5, m = 10, γ = 0.0001 และ t = 100, 500 และ 800

ครั้งที่	n	m	γ	t
1	5	10	0.0001	100
2	5	10	0.0001	100
3	5	10	0.0001	100
4	5	10	0.0001	100
5	5	10	0.0001	100
6	5	10	0.0001	500
7	5	10	0.0001	500
8	5	10	0.0001	500
9	5	10	0.0001	500
10	5	10	0.0001	500
11	5	10	0.0001	800
12	5	10	0.0001	800
13	5	10	0.0001	800
14	5	10	0.0001	800
15	5	10	0.0001	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.16 ค่าพารามิเตอร์ที่ n = 10, m = 100, γ = 0.001 และ t = 100, 500 และ 800

ครั้งที่	n	m	γ	t
1	10	100	0.001	100
2	10	100	0.001	100
3	10	100	0.001	100
4	10	100	0.001	100
5	10	100	0.001	100
6	10	100	0.001	500
7	10	100	0.001	500
8	10	100	0.001	500
9	10	100	0.001	500
10	10	100	0.001	500
11	10	100	0.001	800
12	10	100	0.001	800
13	10	100	0.001	800
14	10	100	0.001	800
15	10	100	0.001	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 ค่าพารามิเตอร์ที่ n = 20, m = 500, γ = 0.01 และ t = 100, 500 และ 800

ครั้งที่	n	m	γ	t
1	20	500	0.01	100
2	20	500	0.01	100
3	20	500	0.01	100
4	20	500	0.01	100
5	20	500	0.01	100
6	20	500	0.01	500
7	20	500	0.01	500
8	20	500	0.01	500
9	20	500	0.01	500
10	20	500	0.01	500
11	20	500	0.01	800
12	20	500	0.01	800
13	20	500	0.01	800
14	20	500	0.01	800
15	20	500	0.01	800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการของอัลกอริทึมหิ่งห้อยสำหรับปัญหา TSP มีดังนี้

1. กำหนดให้ฟังก์ชันวัตถุประสงค์ของปัญหา คือการลดระยะทางในการเดินทางของพนักงานขายให้สั้นที่สุด หรือ Minimize $\sum_{i \neq j} d_{ij} x_{ij}$

2. กำหนดจำนวนประชากรเริ่มต้นของหิ่งห้อย (n)

3. กำหนดค่าพารามิเตอร์ต่างๆ อันประกอบไปด้วย ค่าสัมประสิทธิ์การดูดซับแสง (γ), จำนวนครั้งของการเกิดการกลายพันธุ์แบบอินเวอร์ชัน (m) และจำนวนรอบทั้งหมดที่ใช้ในการหาคำตอบ (t)

4. ทำการหาหิ่งห้อยตัวที่มีแสงสว่างมากที่สุด โดยต้องพิจารณาเหตุการณ์ที่เกิดขึ้น 2 กรณี ดังนี้

- กรณีที่ 1 เมื่อหิ่งห้อย x_i พบว่า หิ่งห้อย x_j มีแสงสว่างมากกว่าตัวมันเอง หิ่งห้อย x_i จะเคลื่อนที่เข้าหา หิ่งห้อย x_j โดยใช้วิธีการเคลื่อนที่แบบคู่ขอบ (Edge Based Movement)
- กรณีที่ 2 เมื่อไม่มีหิ่งห้อยที่มีแสงสว่างมากกว่าหิ่งห้อยตัวที่พิจารณาอยู่ หิ่งห้อย x_i จะเคลื่อนที่แบบสุ่ม โดยวิธีการกลายพันธุ์แบบอินเวอร์ชัน (Inversion Mutation)

5. เมื่อหิ่งห้อยเคลื่อนที่ จะส่งผลให้เกิดการสร้างคำตอบชุดใหม่ขึ้น จากนั้นให้เลือกหิ่งห้อยตัวที่ดีที่สุดจำนวน n ตัว เพื่อนำไปเป็นประชากรเริ่มต้นของหิ่งห้อย ในรอบถัดไป

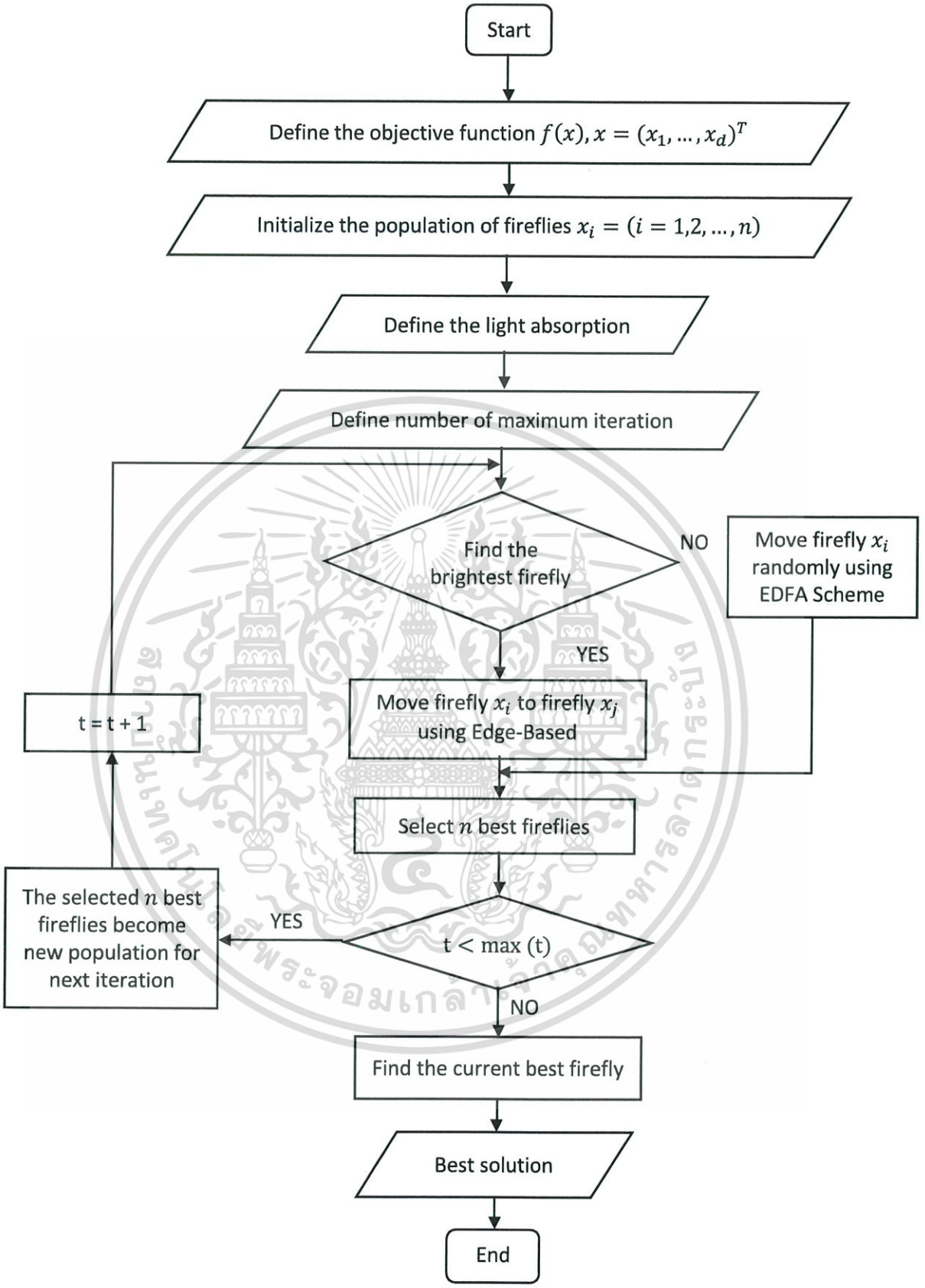
6. เมื่อจำนวนรอบที่ใช้ในการหาคำตอบ เท่ากับจำนวนรอบในการหาคำตอบมากที่สุดที่กำหนดไว้ (Maximum Iteration) ให้ทำการเลือกหิ่งห้อยตัวที่ดีที่สุดมา 1 ตัว และหิ่งห้อยตัวนี้จะเป็นคำตอบของเส้นทางการเดินทางของพนักงานขายที่มีระยะทางรวมต่ำที่สุดนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Input:
    Define objective function f(x), population size p ,light absorption coefficient γ,and
    updating index m
    Begin
        List<Firefly>temp = new List<Firefly>
        //initialize population of firefly
        Firefly[] x = new Firefly [p]
        for i = 1 to p do
            x[i] = Generate_Initial_Solution()
        endfor
        repeat
            for i = 1 to p do
                Firefly f = Get_Most_Attractive_Firefly(x[i])
                if (f != null) then
                    Firefly[]newSolutions = new Firefly[4]
                    newSolutions = Edge-based Movement(x[i])
                    Firefly fnew = newSoulution[Random(1,4)]
                    temp.Add(fnew)
                else
                    for j = 1 to m do
                        Firefly fnew = Move_Random(x[i])
                        temp.Add(fnew)
                    endfor
            endif
            endfor
            //select p brightest fireflies from temp
            x = Get_Brightest_Fireflies(temp)
            temp.Clear()
        until stop condition true
        //output best firefly
        Output(x.min)
end
```

รูปที่ 3.2 รหัสเทียมของอัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย [22]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังแสดงการทำงานของอัลกอริทึมหิ่งห้อยสำหรับปัญหาการเดินทางของพนักงานขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการดำเนินงานในบทนี้ เป็นการทดลองนำค่าพารามิเตอร์ต่างๆที่ได้กำหนดขึ้นมา ของอัลกอริทึมหิ่งห้อย ประกอบไปด้วย จำนวนประชากรหิ่งห้อยเริ่มต้น (n), ค่าสัมประสิทธิ์การดูดซับแสง (γ), จำนวนครั้งการเกิดกระบวนการกลายพันธุ์แบบอินเวอร์ชัน (m) และจำนวนรอบทั้งหมดในการปรับปรุงคำตอบ (t) สำหรับนำไปทำการทดลอง เพื่อดูว่าการเปลี่ยนแปลงของค่าพารามิเตอร์ต่างๆ ส่งผลต่อผลลัพธ์คำตอบที่ได้อย่างไรบ้าง โดยพารามิเตอร์ที่ปรากฏในอัลกอริทึมหิ่งห้อยนี้ มี 4 พารามิเตอร์ด้วยกัน และคณะผู้จัดทำได้ทำการกำหนดค่าต่างๆ สำหรับพารามิเตอร์แต่ละตัว เพื่อใช้ในการทดลองจำนวน 3 ค่าด้วยกัน โดยที่การทดลองนี้ มีการทำซ้ำทั้งหมด 5 ครั้ง กับปัญหา TSP 3 ขนาด ทำให้เกิดจำนวนชุดข้อมูลค่าพารามิเตอร์ที่ใช้ทดลองทั้งหมด 180 ชุด (4x3x5x3 = 180)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

ในบทนี้เป็นการนำเสนอผลการดำเนินงานที่ได้จากการทดลอง โดยข้อมูลที่นำมาใช้ในการดำเนินการ เพื่อหาผลลัพธ์ที่เหมาะสม แบ่งออกเป็นจำนวน 3 ชุดด้วยกัน ได้แก่ ปัญหาขนาด 16 เมือง ปัญหาขนาด 22 เมือง และปัญหาขนาด 52 เมือง และได้ทำการหาเส้นทางการเดินของพนักงานขาย (Traveling Salesman Problem, TSP) ด้วย วิธีการเชิงพันธุกรรม (Genetic Algorithm, GA) และวิธีอัลกอริทึมหิ่งห้อย (Firefly Algorithm, FA)

4.1 ผลการทดลองจากการใช้วิธีการเชิงพันธุกรรมในการจัดเส้นทางการเดินทางของพนักงานขาย

ตารางที่ 4.1 ผลลัพธ์การจัดเส้นทางด้วยวิธีการเชิงพันธุกรรมของปัญหาขนาด 16, 22 และ 52 เมือง

ขนาดของปัญหา	เส้นทาง	ระยะทาง
16	1-16-12-13-14-5-11-9-10-7-6-8-4-2-3	75.00
22	1-8-16-22-4-18-17-2-3-12-13-14-7-6-10-19-20-21-9-11-5-15	81.52
52	1-22-49-32-36-35-34-39-40-37-38-24-48-5-15-6-4-25-46-44-16-50-20-23-31-18-3-19-45-41-8-10-9-43-33-51-12-28-27-11-52-13-14-47-26-29-30-21-17-42-7-2	8779.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองจากการใช้อัลกอริทึมหิ่งห้อยในการจัดเส้นทางการเดินทางของพนักงานขาย

ตารางที่ 4.2 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 16 เมือง

ลำดับ	เส้นทาง	ระยะทาง
1	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
2	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
3	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
4	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
5	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
6	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
7	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
8	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
9	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
10	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
11	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876
12	1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16	73.9876

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 22 เมือง

ลำดับ	เส้นทาง	ระยะทาง
1	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-18-4-22-8	75.3984
2	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
3	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
4	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
5	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
6	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
7	1-22-3-2-17-4-18-8-14-13-12-7-6-15-5-11-9-10-19-20-21-16	75.6666
8	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
9	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
10	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
11	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097
12	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8	75.3097

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 52 เมือง

ลำดับ	เส้นทาง	ระยะทาง
1	1-34-35-36-39-40-43-9-10-33-51-11-52-14-13-47-26-27-28-12-25-46-48-6-4-24-38-15-5-37-44-16-50-29-30-20-23-2-7-42-21-31-18-3-17-41-8-19-45-32-49-22	8846.1
2	1-22-31-18-3-17-7-2-42-21-23-30-29-20-50-16-44-34-35-36-39-37-40-38-5-15-4-6-24-48-46-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49	7794.3
3	1-49-32-45-19-41-8-9-10-43-33-51-12-11-52-14-13-47-26-27-28-25-4-6-15-5-24-48-38-37-40-39-36-35-34-44-46-16-29-50-20-23-30-42-2-7-17-3-18-21-31-22	7854.7
4	1-18-3-17-21-7-2-42-30-20-29-16-46-48-24-38-5-15-6-4-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49-35-36-39-40-37-34-44-50-23-31-22	8039.9
5	1-22-18-3-17-42-7-2-30-21-31-23-20-50-29-16-46-44-34-35-36-39-40-37-38-48-24-5-15-6-4-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49	7690.2
6	1-22-31-18-3-17-21-2-7-42-30-23-20-29-50-16-44-34-35-36-39-40-37-38-46-48-24-5-15-6-4-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49	7745.6
7	1-22-18-31-23-20-50-35-34-44-16-25-12-11-52-13-14-47-26-28-27-51-6-43-33-9-10-8-41-19-45-3-17-21-42-7-2-30-29-46-48-38-40-15-4-5-24-37-39-36-49-32	9254.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4(ต่อ) ผลลัพธ์การจัดเส้นทางด้วยอัลกอริทึมหิ่งห้อย ของปัญหาจำนวนเมือง 52 เมือง

ลำดับ	เส้นทาง	ระยะทาง
8	1-49-32-45-19-41-8-9-10-43-33-51-11-52-14-13-47-26-27-28-12-25-4-6-15-5-24-48-38-37-40-39-36-35-34-46-44-16-29-50-23-20-30-2-7-42-21-17-3-18-31-22	7688.3
9	1-22-31-18-3-17-21-42-7-2-30-29-23-20-50-16-46-44-34-35-36-39-40-37-38-48-24-5-15-6-4-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49	7643.4
10	1-22-18-31-21-42-2-7-17-3-45-19-8-41-9-10-33-43-4-6-15-5-24-38-40-39-37-48-46-25-12-51-11-52-14-13-27-28-26-47-29-30-23-20-50-16-44-34-35-36-49-32	8091
11	1-23-20-30-29-50-16-44-34-35-36-39-40-37-38-5-15-6-24-48-46-25-4-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-45-19-3-17-7-2-42-21-31-18-22-32-49	8066.1
12	1-22-31-18-3-17-21-42-7-2-30-23-20-50-29-16-44-34-35-36-39-40-38-37-46-48-24-5-15-6-4-25-12-28-27-26-47-13-14-52-11-51-33-43-10-9-8-41-19-45-32-49	7599.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบพารามิเตอร์ และการวิเคราะห์ผลการทดลอง

การประยุกต์ใช้อัลกอริทึมหิ่งห้อย สำหรับปัญหา TSP ในปริญญานิพนธ์เล่มนี้ มีพารามิเตอร์ที่ส่งผลต่อคุณภาพของคำตอบด้วยกันทั้งหมด 4 พารามิเตอร์ ได้แก่ จำนวนประชากรหิ่งห้อยเริ่มต้น (n), ค่าสัมประสิทธิ์การดูดซับแสง (γ), จำนวนครั้งการเกิดกระบวนการกลายพันธุ์แบบอินเวอร์ชัน (m) และจำนวนรอบทั้งหมดในการปรับปรุงคำตอบ (t) ดังนั้นทางคณะผู้จัดทำ จึงได้ทำการกำหนดค่าพารามิเตอร์ต่างๆขึ้นมา เพื่อทำการทดลอง และสังเกตว่าค่าของพารามิเตอร์ที่กำหนดขึ้นค่าใดมีความเหมาะสมมากที่สุด ในการนำมาใช้แก้ปัญหา TSP โดยการระบุค่าพารามิเตอร์ที่เหมาะสมในการแก้ปัญหา จะพิจารณาจากความสามารถในการให้คำตอบที่ดีที่สุด ซึ่งในที่นี้คือ ระยะทางรวมของการเดินทางของพนักงานขายไปยังเมืองต่างๆนั่นเอง

จากการทำการดำเนินงานใน บทที่ 3 ทำให้พบว่า สำหรับปัญหา TSP ที่มีขนาด 16 เมือง ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธีฮิวริสติก (Heuristic Method) เท่ากับ 104.73 กิโลเมตร, ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี GA เท่ากับ 75.18 กิโลเมตร และ ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี FA เท่ากับ 74.05 กิโลเมตร โดยที่ระยะทางที่เป็นค่าผลลัพธ์ที่ดีที่สุด (Optimal Solution) ที่ได้จากการคำนวณด้วยวิธีแม่นตรงของปัญหานี้ คือ 74.11 กิโลเมตร

สำหรับปัญหา TSP ที่มีขนาด 22 เมือง ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธีฮิวริสติก เท่ากับ 89.63 กิโลเมตร, ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี GA เท่ากับ 80.51 กิโลเมตร และ ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี FA เท่ากับ 75.41 กิโลเมตร โดยที่ระยะทางที่เป็นค่าผลลัพธ์ที่ดีที่สุด ที่ได้จากการคำนวณด้วยวิธีแม่นตรงของปัญหานี้ คือ 75.67 กิโลเมตร

สำหรับปัญหา TSP ที่มีขนาด 52 เมือง ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธีฮิวริสติก เท่ากับ 8971.70 กิโลเมตร, ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี GA เท่ากับ 8863.84 กิโลเมตร และ ระยะทางรวมเฉลี่ยที่สั้นที่สุด เมื่อใช้วิธี FA เท่ากับ 8069.00 กิโลเมตร โดยที่ระยะทางที่เป็นค่าผลลัพธ์ที่ดีที่สุด ที่ได้จากการคำนวณด้วยวิธีแม่นตรงของปัญหานี้ คือ 7528.77 กิโลเมตร ดังแสดงในตารางที่ 4.5 และเปอร์เซ็นต์ของการเข้าใกล้ค่าผลลัพธ์ที่ดีที่สุด เมื่อใช้วิธีฮิวริสติก, GA และ FA ในการแก้ปัญหา TSP ทั้ง 3 ขนาด ถูกแสดงไว้ในตารางที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ตารางแสดงการเปรียบเทียบผลลัพธ์ที่ได้จากการใช้อัลกอริทึมต่างๆในการแก้ปัญหาการเดินทางของพนักงานขาย

ขนาดปัญหา	ชุดข้อมูล	การเปรียบเทียบผลลัพธ์ระหว่างอัลกอริทึมและคำตอบที่เหมาะสมที่สุด				วิธีที่ให้ผลลัพธ์ที่ดีที่สุด
		ระยะทางรวมเฉลี่ย				
		วิธีฮิวริสติก	GA	FA	วิธีแม่นตรง	
16	ulysses16	104.73	75.18	74.05	74.11	FA
22	ulysses22	89.63	80.51	75.41	75.67	FA
52	berlin52	8971.7	8863.84	8069	7528.77	วิธีแม่นตรง

ค่าพารามิเตอร์ต่างๆที่กำหนดขึ้นเพื่อใช้กับอัลกอริทึมหิ่งห้อย สำหรับการแก้ปัญหา TSP ขนาด 16 เมือง และได้คำตอบที่ดีที่สุดที่ 74.05 กิโลเมตร ได้แก่ n = 20, m = 500, γ = 0.01 และ t = 100, 500 และ 800 (ดูเพิ่มเติมที่ ภาคผนวก ข)

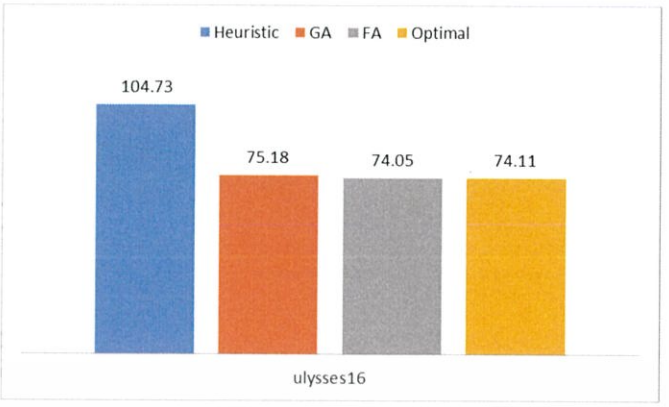
ค่าพารามิเตอร์ต่างๆที่กำหนดขึ้นเพื่อใช้กับอัลกอริทึมหิ่งห้อย สำหรับการแก้ปัญหา TSP ขนาด 22 เมือง และได้คำตอบที่ดีที่สุดที่ 75.41 กิโลมตร ได้แก่ n = 20, m = 10, 100 และ 500 γ = 0.01 และ t = 800 (ดูเพิ่มเติมที่ ภาคผนวก ข)

ค่าพารามิเตอร์ต่างๆที่กำหนดขึ้นเพื่อใช้กับอัลกอริทึมหิ่งห้อย สำหรับการแก้ปัญหา TSP ขนาด 52 เมือง และได้คำตอบที่ดีที่สุดที่ 8069.00 กิโลเมตร ได้แก่ n = 20, m = 500 γ = 0.01 และ t =100, 500 และ 800 (ดูเพิ่มเติมที่ ภาคผนวก ข)

ตารางที่ 4.6 ตารางเปรียบเทียบประสิทธิภาพในการหาคำตอบของอัลกอริทึมหิ่งห้อย และวิธีการเชิงพันธุกรรม ด้วยเปอร์เซ็นต์การเข้าใกล้ค่าผลลัพธ์ที่ดีที่สุด

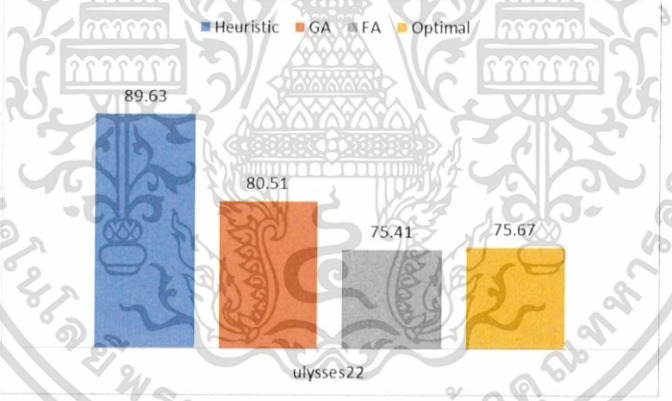
ชุดข้อมูล	ผลลัพธ์ที่ดีที่สุด	FA		GA	
		เปอร์เซ็นต์การเข้าใกล้ผลลัพธ์ที่ดีที่สุด (%)	ค่าเบี่ยงเบนมาตรฐาน	เปอร์เซ็นต์การเข้าใกล้ผลลัพธ์ที่ดีที่สุด (%)	ค่าเบี่ยงเบนมาตรฐาน
ulysses16	74.11	100.08	0.2272	98.56	0.32
ulysses22	75.67	100.34	0.2105	93.6	79.75
berlin52	7528.77	92.82	310	82.27	58.73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 16 เมือง

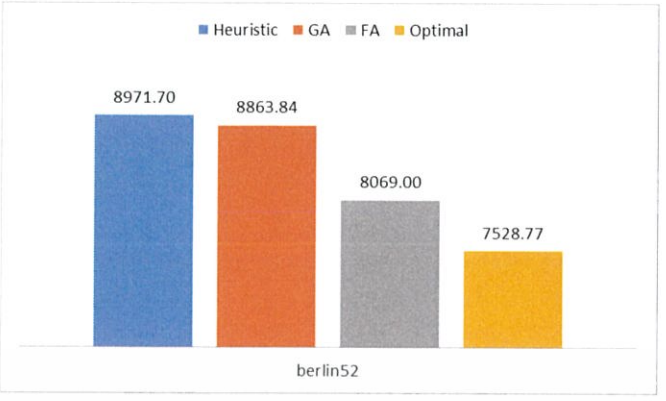
จากรูปที่ 4.1 จะเห็นว่าอัลกอริทึมหิ่งห้อย มีประสิทธิภาพมากที่สุดในการแก้ปัญหา TSP ขนาด 16 เมือง เนื่องระยะทางรวมเฉลี่ยที่เกิดขึ้น คือ 74.05 กิโลเมตร และเป็นผลลัพธ์ที่มีค่าที่ดีที่สุด



รูปที่ 4.2 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 22 เมือง

จากรูปที่ 4.2 จะเห็นว่าอัลกอริทึมหิ่งห้อย มีประสิทธิภาพมากที่สุดในการแก้ปัญหา TSP ขนาด 22 เมือง เนื่องจากระยะทางรวมเฉลี่ยที่เกิดขึ้น คือ 75.41 กิโลเมตร และเป็นผลลัพธ์ที่มีค่าที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 การเปรียบเทียบผลคำตอบที่ได้จากอัลกอริทึมต่างๆ เมื่อนำมาแก้ไขปัญหาขนาด 52 เมือง

จากรูปที่ 4.3 จะเห็นว่าอัลกอริทึมหิ่งห้อย มีประสิทธิภาพมากที่สุดในการแก้ปัญหา TSP ขนาด 52 เมือง เนื่องจากระยะทางรวมเฉลี่ยที่เกิดขึ้น คือ 7528.77 กิโลเมตร

ผลการทดลองที่เกิดขึ้น เมื่อนำอัลกอริทึมต่างๆมาใช้ในการแก้ไขปัญหา TSP 3 ขนาด จากนั้นนำผลลัพธ์คำตอบที่ได้มาทำการเปรียบเทียบกัน โดยนำคำตอบที่ได้จากการแก้ปัญหา ซึ่งก็คือค่าเฉลี่ยระยะทางรวมที่สั้นที่สุด จากวิธีฮิวริสติก วิธีการเชิงพันธุกรรม และอัลกอริทึมหิ่งห้อย มาเปรียบกับผลลัพธ์ที่ดีที่สุด ที่เกิดขึ้นด้วยวิธีแม่นตรง ปรากฏว่า สำหรับปัญหา TSP ขนาด 16 เมือง และ 22 เมือง อัลกอริทึมหิ่งห้อยได้ให้คำตอบที่ดีกว่า วิธีฮิวริสติก และวิธีการเชิงพันธุกรรม อีกทั้งคำตอบที่เกิดขึ้นนั้น ยังเป็นคำตอบที่เป็นค่าผลลัพธ์ที่ดีที่สุดอีกด้วย และสำหรับปัญหา TSP ขนาด 52 เมือง แม้ว่าอัลกอริทึมหิ่งห้อยจะไม่สามารถ ให้คำตอบที่เป็นค่าผลลัพธ์ที่ดีที่สุด ดังเช่นใน 2 ปัญหาข้างต้นที่ได้กล่าวมา แต่อัลกอริทึมหิ่งห้อยก็ยังสามารถให้ผลคำตอบที่ดีกว่า วิธีฮิวริสติก และขั้นตอนวิธีเชิงพันธุกรรม เช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงาน

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์ เพื่อแก้ไขปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP) โดยนำอัลกอริทึมหิ่งห้อย (Firefly Algorithm, FA) มาประยุกต์ใช้ในการแก้ปัญหา และทำการเปรียบเทียบผลที่ได้กับอัลกอริทึมอื่นๆ เพื่อดูความเหมาะสมของการนำอัลกอริทึมหิ่งห้อยกับการเป็นเครื่องมือในการนำมาใช้แก้ปัญหา TSP

5.1 สรุปผลการดำเนินงาน

อัลกอริทึมหิ่งห้อย เป็นหนึ่งในวิธีเมตาฮิวริสติก ที่มีความเหมาะสมในการนำมาใช้แก้ปัญหา TSP เนื่องจากสามารถให้คำตอบที่เป็นค่าผลลัพธ์ที่ดีที่สุด (Optimal Solution) สำหรับปัญหา TSP ขนาด 16 เมือง และ 22 เมือง อัลกอริทึมหิ่งห้อย ได้ให้คำตอบที่มีค่าเท่ากับค่าผลลัพธ์ที่ดีที่สุด และสำหรับปัญหา TSP ขนาด 52 เมือง อัลกอริทึมหิ่งห้อยได้ให้คำตอบที่มีประสิทธิภาพกว่า วิธีฮิวริสติก และวิธีการเชิงพันธุกรรม (Genetic Algorithm) เนื่องจากสามารถให้คำตอบได้ใกล้เคียงกับค่าผลลัพธ์ที่ดีที่สุด มากที่สุดนั่นเอง

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

ปัญหาและอุปสรรคที่พบในการดำเนินงาน มีดังนี้

1. พารามิเตอร์ที่ต้องกำหนดขึ้นใน FA มีด้วยกันหลายค่า แต่ละค่าจะส่งผลต่อคุณภาพของคำตอบที่ได้ โดยเฉพาะการไม่ทราบค่าสัมประสิทธิ์การดูดซับแสง (γ) ที่แน่ชัด อาจทำให้คุณภาพของคำตอบที่ได้นั้น ไม่ใช่คำตอบที่ดีที่สุด

2. เมื่อขนาดของปัญหามีขนาดใหญ่ขึ้น เช่น ปัญหาขนาด 52 เมือง ระยะเวลาที่ใช้ในการประมวลผลจะใช้เวลานานขึ้น

3. ที่ค่าสัมประสิทธิ์การดูดซับแสง (γ) เท่ากับ 0.01 จำนวนหิ่งห้อยที่เกิดการเคลื่อนที่แบบสุ่ม มีจำนวนเพิ่มมากขึ้น ทำให้อัตราการพัฒนา ปรับปรุงคำตอบ เป็นไปด้วยวิธีการกลยุทธ์เชิงพันธุกรรม (Evolution Strategy) มากขึ้น

5.3 ข้อเสนอแนะ

จากการทำการทดลองทำให้ทราบว่า การกำหนดจำนวนหิ่งห้อย (n) และการกำหนดจำนวนครั้งของการเกิดการกลายพันธุ์แบบอินเวอร์ชัน (m) ส่งผลต่อเวลาในการประมวลผลเป็นอย่างมาก เนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ทำซ้ำ ดัดแปลง หรือนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่มีจำนวนหิ่งห้อยมากๆ จะทำให้เกิดการหาจำนวนคำตอบที่เป็นไปได้ออกมามากขึ้น ส่วนจำนวนครั้งของการกลายพันธุ์แบบอินเวอร์ชัน หากกำหนดให้มีปริมาณมาก จะทำให้อัลกอริทึมต้องทำการสร้างชุดคำตอบเพิ่มขึ้น ซึ่งจะทำให้เพิ่มเวลาในการประมวลผล นอกจากนี้ การกำหนดค่าสัมประสิทธิ์การดูดซับแสง (γ) ยังส่งผลต่อจำนวนหิ่งห้อย ที่เกิดการเคลื่อนที่แบบสุ่มด้วย ยิ่งกำหนดค่าสัมประสิทธิ์การดูดซับแสงให้มีค่ามากขึ้นเท่าใด จำนวนหิ่งห้อยที่ทำการเคลื่อนที่แบบสุ่มยิ่งมีมากขึ้นเท่านั้น ฉะนั้นเมื่อนำอัลกอริทึมไปประยุกต์ใช้กับปัญหา TSP หรือปัญหาอื่นๆที่ใกล้เคียง ก็ควรคำนึงถึงค่าสัมประสิทธิ์การดูดซับแสงนี้ด้วย เพื่อไม่ให้หิ่งห้อยทุกตัวในปริภูมิ เกิดการเคลื่อนที่แบบสุ่มทั้งหมด ดังนั้นหากต้องการลดเวลาในการประมวลผล และปรับปรุงประสิทธิภาพของอัลกอริทึม ก็อาจนำข้อสรุปเหล่านี้ไปใช้เป็นแนวทางในการปรับปรุงต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Pickover, C. A. (2009). *The math book: from Pythagoras to the 57th dimension, 250 milestones in the history of mathematics*. Sterling Publishing Company, Inc..

[2] Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). *Handbooks in operations research and management science*, *12*, 1-68.

[3] Biggs, N., Lloyd, E. K., & Wilson, R. J. (1976). *Graph Theory, 1736-1936*. Oxford University Press.

[4] J. Dalgety. "The Icosian Game." (Jul 6, 2009) puzzlemuseum.com/month/picm02/200207icosian.htm.

[5] Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2011). *The traveling salesman problem: a computational study*. Princeton university press.

[6] Arora, S., Lund, C., Motwani, R., Sudan, M., & Szegedy, M. (1998). Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, *45*(3), 501-555.

[7] optimization.mccormick.northwestern.edu/index.php/Traveling_salesman_problems

[8] Balas, E., & Toth, P. (1983). *Branch and bound methods for the traveling salesman problem* (No. MSRR-488). CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENCES RESEARCH GROUP.

[9] Padberg, M., & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, *33*(1), 60-100.

[10] Martí, R., & Reinelt, G. (2011). *The linear ordering problem: exact and heuristic methods in combinatorial optimization* (Vol. 175). Springer Science & Business Media.

[11] Gendreau, M., Hertz, A., Laporte, G., & Stan, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, *46*(3), 330-335.

[12] Kizilateş, G., & Nuriyeva, F. (2013). On the nearest neighbor algorithms for the traveling salesman problem. In *Advances in Computational Science, Engineering and Information Technology* (pp. 111-118). Springer International Publishing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[13] Kaveh, A. (2017). *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Iran University of Science and Technology, Tehran, Iran (Islamic Republic of)

[14] Geng, X., Chen, Z., Yang, W., Shi, D., & Zhao, K. (2011). Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, *11*(4), 3680-3689.

[15] Wang, K. P., Huang, L., Zhou, C. G., & Pang, W. (2003, November). Particle swarm optimization for traveling salesman problem. In *Machine Learning and Cybernetics, 2003 International Conference on* (Vol. 3, pp. 1583-1585). IEEE.

[16] Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *biosystems*, *43*(2), 73-81.

[17] Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. (1985, July). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications* (pp. 160-165).

[18] Yang, X. S. (2010). *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons.

[19] Karaboga, D., & Gorkemli, B. (2011, June). A combinatorial artificial bee colony algorithm for traveling salesman problem. In *Innovations in intelligent systems and applications (inista), 2011 international symposium on* (pp. 50-53). IEEE.

[20] Ouaarab, A., Ahiod, B., & Yang, X. S. (2014). Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications*, *24*(7-8), 1659-1669.

[21] Jati, G. (2011). Evolutionary discrete firefly algorithm for travelling salesman problem. *Adaptive and intelligent systems*, 393-403.

[22] Yang, X. S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. (Eds.). (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes.

[23] Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209-218). Springer London.

[24] Senthilnath, J., Omkar, S. N., & Mani, V. (2011). Clustering using firefly algorithm: performance study. *Swarm and Evolutionary Computation*, *1*(3), 164-171.

[25] Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Springer Berlin Heidelberg.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[26] Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

[27] Contreras-Bolton, C., & Parada, V. (2015). Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. *PloS one*, *10*(9), e0137724.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
รหัสของโปรแกรม MATLAB (MATLAB Code)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ฟังก์ชัน pick_unique_firefly

```
function [FF_New] = pick_unique_firefly(FF_New)

for i = 1:length(FF_New)
%
All_City(i,:) = FF_New(i).City;
%
end
%
[~,loc] = unique(All_City,'rows');
FF_New = FF_New(loc);
%
end
```

2. ฟังก์ชัน calculate_brightness

```
function [FF] = calculate_brightness(FF,D)

for i = 1:length(FF)
    %
    City = FF(i).City;
    %
    City = [City City(1)]; %Append the first city after the last city (round trip)
    %
    FF(i).Brightness = 1/calculate_total_distance(D,City);
    %
end


end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ฟังก์ชัน move_randomly

```
function [FF_New] = move_randomly(FF,i,m)
%
City = FF(i).City;
num_city = length(City);
%
for k = 1:3*m %Just to make sure that I sample enough permutation.
    loc = randperm(num_city);
    x = loc(1);
    y = loc(2);
    %
    if x > y  %Switch order so that x = y
        t = x;
        x = y;
        y = t;
    end
    %
    B1 = 1:x-1;
    B2 = x:y;
    B3 = y+1:num_city;
    %
    CI = [City(B1) fliplr(City(B2)) City(B3)];
    CIM = rearrange_city_array(CI);
    All_City(k,:) = CIM;
    FF_New(k).City = CIM;
    FF_New(k).Prev_FF = i;
end
%
[~,loc] = unique(All_City,'rows');
%
if length(loc) > m
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    FF_New = FF_New(loc(1:m));
else
    FF_New = FF_New(loc);
    %fprintf('Warning: in Move_Randomly, m = %i, but can only find %i unique inversions.
\n',m,length(loc))
end
%
end

function CM = rearrange_city_array(City)
%This function makes the city array begins with 1.
num_city = length(City);
%
C = [City City];
%
loc = find(C == 1);
%
loc = loc(1);
%
CM = C(loc:loc+num_city-1);
%
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ฟังก์ชัน rank_FF

```
function [FF_New] = rank_FF(FF_New)
%
BR = [FF_New.Brightness];
%
[~,loc] = sort(BR,'descend');
%
FF_New = FF_New(loc);
end
```

5. ฟังก์ชัน move_firefly_toward_attractive_FF

```
function [All_C_Unique,Edge] = move_firefly_toward_attractive_FF(FF,i)
%
j = FF(i).Most_Attractive_FF;
%
Edge = choose_edge(FF,i,j); %Edge is the randomly chosen edge that exists in firefly 'j'
but not in firefly 'i'.
%
[x,y] = find_location_xy(FF,i,Edge); %x and y are locations of city from the Edge chosen
%
[city_bound_x,city_bound_y] = expand_chuck_city(FF,i,j,x,y);
%
[All_C_Unique] = find_solution_firefly(FF,i,city_bound_x,city_bound_y,Edge);
%
end

function [edge_chosen] = choose_edge(FF,i,j)
%
edge_array = FF(i).EdgeDiff{j};
%
n_edge = size(edge_array,1); %Count the number of row.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%
loc = randperm(n_edge);
loc = loc(1);
%
edge_chosen = edge_array(loc,:);
%
end

function [x,y] = find_location_xy(FF,i,Edge)
%
City = FF(i).City;
%
EL = Edge(1);
EH = Edge(2);
%
loc_EL = find(City == EL);
loc_EH = find(City == EH);
%
if loc_EL < loc_EH
    x = loc_EL;
    y = loc_EH;
else
    x = loc_EH;
    y = loc_EL;
end
%
end

function [city_bound_x,city_bound_y] = expand_chuck_city(FF,i,j,x,y)
%
City = FF(i).City;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
City = [City City(1)]; %Append the first city after the last.
%
EJ = FF(j).Edge;
%--------------------
continue_loop = true;
xlow = x;
while continue_loop && xlow > 1
    %
    EI = sort([City(xlow) City(xlow-1)]); %Sort the number
    %
    if test_edge_match_j(EI,EJ)
        xlow = xlow - 1;
    else
        continue_loop = false;
    end
    %
end
%--------------------
continue_loop = true;
xhigh = x;
while continue_loop && xhigh < y
    %
    EI = sort([City(xhigh) City(xhigh+1)]); %Sort the number
    %
    if test_edge_match_j(EI,EJ)
        xhigh = xhigh + 1;
    else
        continue_loop = false;
    end
    %
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%--------------------
continue_loop = true;
ylow = y;
while continue_loop && ylow > xhigh
    %
    EI = sort([City(ylow) City(ylow-1)]); %Sort the number
    %
    if test_edge_match_j(EI,EJ)
        ylow = ylow - 1;
    else
        continue_loop = false;
    end
    %
end
%--------------------
continue_loop = true;
yhigh = y;
while continue_loop && yhigh < length(City)
    %
    EI = sort([City(yhigh) City(yhigh+1)]); %Sort the number
    %
    if test_edge_match_j(EI,EJ)
        yhigh = yhigh + 1;
    else
        continue_loop = false;
    end
    %
end
%
city_bound_x = [xlow xhigh];
city_bound_y = [ylow yhigh];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%
end

function [found_match_edge] = test_edge_match_j(edgei,EJ)
%
found_match_edge = false;
for j = 1:size(EJ,1) %Number of rows (or edges)
    edgej = EJ(j,:);
    if test_edge_match(edgei,edgej)
        found_match_edge = true;
    end
end
end

function [match] = test_edge_match(edgei,edgej)
a = length(find(edgei == edgej));
%
if a == 2
    match = true;
else
    match = false;
end
end

function [All_C_Unique] =
find_solution_firefly(FF,i,city_bound_x,city_bound_y,Edge_Target)
%
City = FF(i).City;
City = [City City(1)];
%
B1 = 1:city_bound_x(1)-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B2 = city_bound_x(1):city_bound_x(2);
B3 = city_bound_x(2)+1:city_bound_y(1)-1;
B4 = city_bound_y(1):city_bound_y(2);
B5 = city_bound_y(2)+1:length(City);
%-------------------------------------------------------------------
%Switch sx before sy
%-------------------------------------------------------------------
C1(1).City = [City(B1) City(B3) City(B2)         City(B4)         City(B5)];
C1(2).City = [City(B1) City(B3) fliplr(City(B2)) City(B4)         City(B5)];
C1(3).City = [City(B1) City(B3) City(B2)         fliplr(City(B4)) City(B5)];
C1(4).City = [City(B1) City(B3) fliplr(City(B2)) fliplr(City(B4)) City(B5)];
C1(1).City = C1(1).City(1:end-1);
C1(2).City = C1(2).City(1:end-1);
if city_bound_y(2) == length(City) %This is the case where the right bound for y is at the
appended location in the city array.
    C1(3).City = C1(3).City(2:end);
    C1(4).City = C1(4).City(2:end);
else
    C1(3).City = C1(3).City(1:end-1);
    C1(4).City = C1(4).City(1:end-1);
end
for k = 1:length(C1)
    C1(k).Edge = calculate_edge(C1(k).City);
    C1(k).Valid_Solution = test_edge_match_j(Edge_Target,C1(k).Edge);
end
%-------------------------------------------------------------------
%Switch sx after sy
%-------------------------------------------------------------------
C2(1).City = [City(B1) City(B3) (City(B4)) (City(B2)) City(B5)];
C2(2).City = [City(B1) City(B3) fliplr(City(B4)) (City(B2)) City(B5)];
C2(3).City = [City(B1) City(B3) (City(B4)) fliplr(City(B2)) City(B5)];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
C2(4).City = [City(B1) City(B3) fliplr(City(B4)) fliplr(City(B2)) City(B5)];
if city_bound_y(2) == length(City) %This is the case where the right bound for y is at the
appended location in the city array.
    C2(1).City = C2(1).City(2:end);
    C2(2).City = C2(2).City(2:end);
    C2(3).City = C2(3).City(2:end);
    C2(4).City = C2(4).City(2:end);
else
    C2(1).City = C2(1).City(1:end-1);
    C2(2).City = C2(2).City(1:end-1);
    C2(3).City = C2(3).City(1:end-1);
    C2(4).City = C2(4).City(1:end-1);
end
for k = 1:length(C2)
    C2(k).Edge = calculate_edge(C2(k).City);
    C2(k).Valid_Solution = test_edge_match_j(Edge_Target,C2(k).Edge);
end
%-----------------------------------------------------------
%Switch sy before sx
%-----------------------------------------------------------
C3(1).City = [City(B1) (City(B4)) (City(B2)) City(B3) City(B5)];
C3(2).City = [City(B1) fliplr(City(B4)) (City(B2)) City(B3) City(B5)];
C3(3).City = [City(B1) (City(B4)) fliplr(City(B2)) City(B3) City(B5)];
C3(4).City = [City(B1) fliplr(City(B4)) fliplr(City(B2)) City(B3) City(B5)];
if city_bound_y(2) == length(City) %This is the case where the right bound for y is at the
appended location in the city array.
    C3(1).City = C3(1).City(2:end);
    C3(2).City = C3(2).City(2:end);
    C3(3).City = C3(3).City(2:end);
    C3(4).City = C3(4).City(2:end);
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    C3(1).City = C3(1).City(1:end-1);
    C3(2).City = C3(2).City(1:end-1);
    C3(3).City = C3(3).City(1:end-1);
    C3(4).City = C3(4).City(1:end-1);
end
for k = 1:length(C3)
    C3(k).Edge = calculate_edge(C3(k).City);
    C3(k).Valid_Solution = test_edge_match_j(Edge_Target,C3(k).Edge);
end
%-----------------------------------------------------------------
%Switch sy after sx
%-----------------------------------------------------------------
C4(1).City = [City(B1) (City(B2)) (City(B4)) City(B3) City(B5)];
C4(2).City = [City(B1) fliplr(City(B2)) (City(B4)) City(B3) City(B5)];
C4(3).City = [City(B1) (City(B2)) fliplr(City(B4)) City(B3) City(B5)];
C4(4).City = [City(B1) fliplr(City(B2)) fliplr(City(B4)) City(B3) City(B5)];
if city_bound_y(2) == length(City) %This is the case where the right bound for y is at the
appended location in the city array.
    C4(1).City = C4(1).City(2:end);
    C4(2).City = C4(2).City(2:end);
    C4(3).City = C4(3).City(2:end);
    C4(4).City = C4(4).City(2:end);
else
    C4(1).City = C4(1).City(1:end-1);
    C4(2).City = C4(2).City(1:end-1);
    C4(3).City = C4(3).City(1:end-1);
    C4(4).City = C4(4).City(1:end-1);
end
for k = 1:length(C3)
    C4(k).Edge = calculate_edge(C4(k).City);
    C4(k).Valid_Solution = test_edge_match_j(Edge_Target,C4(k).Edge);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
%-------------------------------------------------------------------
%Analyze all results
%-------------------------------------------------------------------
index = 1;
loc = find([C1.Valid_Solution]);
for k = 1:length(loc)
   All_C(index).City = C1(loc(k)).City;
   %All_C(index).Edge = C1(loc(k)).Edge;
   index = index + 1;
end
loc = find([C2.Valid_Solution]);
for k = 1:length(loc)
   All_C(index).City = C2(loc(k)).City;
   %All_C(index).Edge = C2(loc(k)).Edge;
   index = index + 1;
end
loc = find([C3.Valid_Solution]);
for k = 1:length(loc)
   All_C(index).City = C3(loc(k)).City;
   %All_C(index).Edge = C3(loc(k)).Edge;
   index = index + 1;
end
loc = find([C4.Valid_Solution]);
for k = 1:length(loc)
   All_C(index).City = C4(loc(k)).City;
   %All_C(index).Edge = C4(loc(k)).Edge;
   index = index + 1;
end
%-------------------------------------------------------------------
%Analyze all results
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------------------
for k = 1:length(All_C)
    All_C(k).Prev_FF = i;
    All_C(k).City = rearrange_city_array(All_C(k).City);
    All_City(k,:) = All_C(k).City;
end
%-------------------------------------------------------------------
%Finding Unique Solutions
%-------------------------------------------------------------------
[~,loc] = unique(All_City,'rows');
All_C_Unique = All_C(loc);
%
end

function [edge_array] = calculate_edge(City)
%
nc = length(City);
City = [City City(1)];
%
for i = 1:nc
    edge_sort = sort([City(i) City(i+1)]);
    edge_array(i,:) = edge_sort;
end
edge_array = sortrows(edge_array);
%
end

function CM = rearrange_city_array(City)
%This function makes the city array begins with 1.
num_city = length(City);
%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
C = [City City];
%
loc = find(C == 1);
%
loc = loc(1);
%
CM = C(loc:loc+num_city-1);
%
end
```

6. ฟังก์ชัน calculate_attractiveness

```
function [FF,Att] = calculate_attractiveness(FF,gamma)
%
Att = zeros(length(FF),length(FF));
%
for i = 1:length(FF)
    for j = 1:length(FF)
        Att(i,j) = calc_attrac(FF,i,j,gamma);
    end
    FF(i).Att = Att(i,:);
end
end

function [beta_r] = calc_attrac(FF,i,j,gamma)
%
beta0 = max([FF(i).Brightness FF(j).Brightness]);
r = FF(i).rDist(j);
%
beta_r = beta0 * exp(-gamma*r^2);
%
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ฟังก์ชัน calculate_distance_between_fireflies

```
function [FF,rDist] = calculate_distance_between_fireflies(FF,EdgeDiff)
%
num_FF = length(FF);
num_city = length(FF(1).City);
%
rDist = zeros(num_FF,num_FF);
%
for i = 1:num_FF
    for j = 1:num_FF
        rDist(i,j) = size(EdgeDiff{i,j},1)/num_city * 10;
    end
    %
    FF(i).rDist = rDist(i,:);
end
%
end
```

8. ฟังก์ชัน calculate_edges_between_fireflies

```
function [FF,EdgeDiff] = calculate_edges_between_fireflies(FF)
%Initialize cell array
EdgeDiff{length(FF),length(FF)} = [];
%
for i = 1:length(FF)
    for j = 1:length(FF)
        %
        EI = FF(i).Edge;
        EJ = FF(j).Edge;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        %
        [ED] = find_edge_diff(EI,EJ);
        %
        EdgeDiff{i,j} = ED;
        %
    end
    FF(i).EdgeDiff = EdgeDiff(i,:);
end
end

function [ED] = find_edge_diff(EI,EJ)
%EI and EJ are 2D array containing edges
%ED contain the edges present in firefly "j" but no in firefly "i".
ED = [];
edge_index = 1;
for j = 1:size(EJ,1) %Number of rows (or edges)
    edgej = EJ(j,:);
    found_match_edge = false;
    for i = 1:size(EI,1) %Number of rows (or edges)
        edgei = EI(i,:);
        if test_edge_match(edgei,edgej)
            found_match_edge = true;
        end
    end
    %
    if ~found_match_edge
        ED(edge_index,:) = edgej;
        edge_index = edge_index + 1;
    end
end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function [match] = test_edge_match(edgei,edgej)
a = length(find(edgei == edgej));
%
if a == 2
    match = true;
else
    match = false;
end
end
```

9. ฟังก์ชัน calculate_total_distance

```
function [distance] = calculate_total_distance(D,C)
%
distance = 0;
C = [C C(1)];
%
for i = 1:length(C)-1
    distance = distance + D(C(i),C(i+1));
end
%
end
```

10. ฟังก์ชัน find_most_attractive_firefly

```
function [FF] = find_most_attractive_firefly(FF)
%
for i = 1:length(FF)
    Att = FF(i).Att;
    [~,loc] = max(Att);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    %
    loc_FF = pick_randomly(loc); %In case there are more than 1 most attractive FF. We
have to choose one FF randomly.
    %
    if loc_FF == i %No attractive neighbor
        FF(i).Move_Randomly = true;
        FF(i).Most_Attractive_FF = NaN;
    else %Might find attrative neighbor
        if abs(FF(i).Att(loc_FF) - FF(i).Att(i)) < 1e-12 %Sometimes the firefly has the same
brightness as the neighbor and in this case it should just move randomly.
            FF(i).Move_Randomly = true;
            FF(i).Most_Attractive_FF = NaN;
        else
            FF(i).Move_Randomly = false;
            FF(i).Most_Attractive_FF = loc_FF;
        end
    end
end
end

function [loc_FF] = pick_randomly(loc)
    n_loc = length(loc);
    rp = randperm(n_loc);
    loc = loc(rp);
    loc_FF = loc(1);
end
```

11. ฟังก์ชัน calculate_edge_fireflies

```
function [FF] = calculate_edge_fireflies(FF)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nc = length(FF(1).City); %Number of city
%
for ff_index = 1:length(FF)
    %
    City = FF(ff_index).City;
    City = [City City(1)]; %Append the first city to the last city
    %
    for i = 1:nc
        edge_sort = sort([City(i) City(i+1)]);
        edge_array(i,:) = edge_sort;
    end
    edge_array = sortrows(edge_array);
    %
    FF(ff_index).Edge = edge_array;
end

end
```

12. ฟังก์ชัน initialize_fireflies

```
%% Setting up
clear; close all; clc;
path(path,[pwd filesep 'functions'])
%% Simulation Parameter
it_max = 100;
%problem_type = 'Debug';
problem_type = 'ulysses16';
%problem_type = 'berlin52';
%problem_type = 'ulysses22';
%% Model Parameter
% n = number of firefly
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% m = number of inversion mutation
% gamma = light absorption coefficient
% D = distance matrix;
[D,num_city,n,m,gamma] = get_input(problem_type);
%% Initialize firefly
%FF is a structure
[FF] = initialize_fireflies(problem_type,n,num_city);  %Generate initial solution
%% Algorithm
it = 1;
continue_loop = true;
Data = [];
while continue_loop
    [FF] = calculate_edge_fireflies(FF);
    [FF] = calculate_brightness(FF,D);
    [FF,EdgeDiff] = calculate_edges_between_fireflies(FF);
    [FF,rDist] = calculate_distance_between_fireflies(FF,EdgeDiff);
    [FF,Att] = calculate_attractiveness(FF,gamma);
    [FF] = find_most_attractive_firefly(FF);
    FF_New = [];
    for i = 1:n
        if FF(i).Move_Randomly
            [Temp] = move_randomly(FF,i,m);
            FF_New = [FF_New Temp];
        else
            [Temp,~] = move_firefly_toward_attractive_FF(FF,i); %Temp = temporary struct
to store fireflies
            FF_New = [FF_New Temp];
        end
    end
    [FF_New] = pick_unique_firefly(FF_New);
    [FF_New] = calculate_brightness(FF_New,D);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    [FF_New] = rank_FF(FF_New);
    Data(it).Number_of_New_Solution = length(FF_New);
    Data(it).Best_Route = FF_New(1).City;
    Data(it).Best_Distance = calculate_total_distance(D,FF_New(1).City);
    Data(it).Number_of_Random_FF = length(find([FF(:).Move_Randomly]));
    %
    FF = FF_New(1:n);
    %
    it = it + 1;
    if it>it_max
        continue_loop = false;
    end
    if mod(it,1) == 0
        fprintf('It = %i \n',it-1)
    end
end
```

13. ฟังก์ชัน get_input

```
function [D,num_city,n,m,gamma] = get_input(problem_type)
%
if strcmp(problem_type,'Debug')
    x = [1 10 2 4 5];
    y = [2 5  1 3 5];
    D = generate_distance_matrix(x,y);
    num_city = 5;
    n = 3;
    m = 20;
    gamma = 0.001;
elseif strcmp(problem_type,'ulysses16')
    filename = 'TSP 16.xlsx';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    [x,y] = read_excel(filename);
    D = generate_distance_matrix(x,y);
    num_city = length(x);
    n = 10;
    m = 20;
    gamma = 0.02;
elseif strcmp(problem_type,'berlin52')
    filename = 'TSP 52.xlsx';
    [x,y] = read_excel(filename);
    D = generate_distance_matrix(x,y);
    num_city = length(x);
    n = 5;
    m = 1000;
    gamma = 0.001;
elseif strcmp(problem_type,'ulysses22')
    filename = 'TSP 22.xlsx';
    [x,y] = read_excel(filename);
    D = generate_distance_matrix(x,y);
    num_city = length(x);
    n = 10;
    m = 20;
    gamma = 0.001;
else
    error('Unknown Problem')
end
%
end

function DM = generate_distance_matrix(x,y)
n = length(x);
for i = 1:n
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for j = 1:n
        DM(i,j) =  sqrt( (x(i)-x(j))^2 + (y(i)-y(j))^2);
    end
end
end

function [x,y] = read_excel(filename)
num = xlsread(filename);
x = num(:,2);
y = num(:,3);
end
```

14. ฟังก์ชันที่ใช้สำหรับในการรันโปรแกรม driver_test

```
%% Setting up
clear; close all; clc;
path(path,[pwd filesep 'functions'])
%% Simulation Parameter
it_max = 100;
%problem_type = 'Debug';
problem_type = 'ulysses16';
%problem_type = 'berlin52';
%problem_type = 'ulysses22';
%% Model Parameter
% n = number of firefly
% m = number of inversion mutation
% gamma = light absorption coefficient
% D = distance matrix;
[D,num_city,n,m,gamma] = get_input(problem_type);
%% Initialize firefly
%FF is a structure
[FF] = initialize_fireflies(problem_type,n,num_city);  %Generate initial solution
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%% Algorithm
it = 1;
continue_loop = true;
Data = [];
while continue_loop
    [FF] = calculate_edge_fireflies(FF);
    [FF] = calculate_brightness(FF,D);
    [FF,EdgeDiff] = calculate_edges_between_fireflies(FF);
    [FF,rDist] = calculate_distance_between_fireflies(FF,EdgeDiff);
    [FF,Att] = calculate_attractiveness(FF,gamma);
    [FF] = find_most_attractive_firefly(FF);
    FF_New = [];
    for i = 1:n
        if FF(i).Move_Randomly
            [Temp] = move_randomly(FF,i,m);
            FF_New = [FF_New Temp];
        else
            [Temp,~] = move_firefly_toward_attractive_FF(FF,i); %Temp = temporary struct
to store fireflies
            FF_New = [FF_New Temp];
        end
    end
    [FF_New] = pick_unique_firefly(FF_New);
    [FF_New] = calculate_brightness(FF_New,D);
    [FF_New] = rank_FF(FF_New);
    Data(it).Number_of_New_Solution = length(FF_New);
    Data(it).Best_Route = FF_New(1).City;
    Data(it).Best_Distance = calculate_total_distance(D,FF_New(1).City);
    Data(it).Number_of_Random_FF = length(find([FF(:).Move_Randomly]));
    %
    FF = FF_New(1:n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    %
    it = it + 1;
    if it>it_max
        continue_loop = false;
    end
    if mod(it,1) == 0
        fprintf('It = %i \n',it-1)
    end
end
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	27	74.8701	2
2	5	10	0.0001	100	52	73.9876	2
3	5	10	0.0001	100	43	75.0899	2
4	5	10	0.0001	100	15	75.2000	2
5	5	10	0.0001	100	32	73.9998	2
6	10	10	0.0001	100	20	75.5624	2
7	10	10	0.0001	100	26	75.0692	2
8	10	10	0.0001	100	14	75.0899	1
9	10	10	0.0001	100	32	74.8270	2
10	10	10	0.0001	100	32	73.9876	4
11	20	10	0.0001	100	19	74.1989	2
12	20	10	0.0001	100	14	73.9876	1
13	20	10	0.0001	100	24	74.0965	2
14	20	10	0.0001	100	11	75.0899	1
15	20	10	0.0001	100	20	74.8701	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	100	0.001	500	9	73.9998	4
2	5	100	0.001	500	6	73.9998	1
3	5	100	0.001	500	10	73.9998	5
4	5	100	0.001	500	11	74.8701	5
5	5	100	0.001	500	9	73.9998	4
6	10	100	0.001	500	10	73.9876	8
7	10	100	0.001	500	8	73.9876	2
8	10	100	0.001	500	9	73.9876	6
9	10	100	0.001	500	9	73.9998	2
10	10	100	0.001	500	10	73.9998	10
11	20	100	0.001	500	11	73.9876	8
12	20	100	0.001	500	11	73.9876	6
13	20	100	0.001	500	10	73.9998	6
14	20	100	0.001	500	8	74.5939	3
15	20	100	0.001	500	7	73.9998	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	500	0.01	800	9	73.9876	5
2	5	500	0.01	800	7	74.0013	5
3	5	500	0.01	800	9	73.9876	5
4	5	500	0.01	800	9	73.9998	5
5	5	500	0.01	800	11	73.9876	5
6	10	500	0.01	800	12	73.9876	10
7	10	500	0.01	800	11	73.9876	10
8	10	500	0.01	800	7	73.9998	10
9	10	500	0.01	800	11	74.5939	10
10	10	500	0.01	800	8	73.9998	10
11	20	500	0.01	800	7	73.9876	20
12	20	500	0.01	800	6	74.0013	20
13	20	500	0.01	800	9	73.9998	20
14	20	500	0.01	800	8	73.9876	20
15	20	500	0.01	800	7	74.5939	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	27	74.8701	2
2	5	10	0.0001	100	52	73.9876	2
3	5	10	0.0001	100	43	75.0899	2
4	5	10	0.0001	100	15	75.2000	2
5	5	10	0.0001	100	32	73.9998	2
6	5	100	0.0001	100	9	74.1087	1
7	5	100	0.0001	100	12	74.1087	2
8	5	100	0.0001	100	9	74.8270	1
9	5	100	0.0001	100	11	75.2000	1
10	5	100	0.0001	100	10	74.8701	2
11	5	500	0.0001	100	8	75.2000	2
12	5	500	0.0001	100	11	73.9876	2
13	5	500	0.0001	100	8	74.1989	1
14	5	500	0.0001	100	10	74.0013	4
15	5	500	0.0001	100	12	73.9998	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	10	0.001	500	16	73.9876	1
2	10	10	0.001	500	31	73.9998	5
3	10	10	0.001	500	15	73.9876	3
4	10	10	0.001	500	19	75.0899	1
5	10	10	0.001	500	27	73.9876	5
6	10	100	0.001	500	11	73.9876	5
7	10	100	0.001	500	11	74.5939	7
8	10	100	0.001	500	11	73.9998	5
9	10	100	0.001	500	12	73.9876	8
10	10	100	0.001	500	8	73.9998	4
11	10	500	0.001	500	10	73.9998	6
12	10	500	0.001	500	8	73.9998	6
13	10	500	0.001	500	9	73.9876	8
14	10	500	0.001	500	10	73.9998	8
15	10	500	0.001	500	10	74.1087	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	10	0.01	800	148	73.9876	20
2	20	10	0.01	800	20	73.9876	20
3	20	10	0.01	800	20	73.9876	20
4	20	10	0.01	800	24	73.9876	20
5	20	10	0.01	800	23	73.9876	20
6	20	100	0.01	800	8	73.9876	20
7	20	100	0.01	800	8	73.9998	20
8	20	100	0.01	800	10	73.9876	20
9	20	100	0.01	800	9	73.9876	20
10	20	100	0.01	800	9	73.9876	20
11	20	500	0.01	800	7	73.9876	20
12	20	500	0.01	800	9	73.9876	20
13	20	500	0.01	800	8	73.9998	20
14	20	500	0.01	800	9	73.9876	20
15	20	500	0.01	800	10	73.9876	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	98	9693.8	1
2	5	10	0.0001	100	95	10161.0	1
3	5	10	0.0001	100	96	9963.7	2
4	5	10	0.0001	100	94	9254.8	2
5	5	10	0.0001	100	99	9852.0	1
6	5	10	0.001	100	96	10800.0	2
7	5	10	0.001	100	99	9523.9	1
8	5	10	0.001	100	98	10196.0	2
9	5	10	0.001	100	99	9964.8	1
10	5	10	0.001	100	100	10065.0	2
11	5	10	0.01	100	100	9660.8	1
12	5	10	0.01	100	97	10192.0	1
13	5	10	0.01	100	100	9749.5	1
14	5	10	0.01	100	98	9824.6	2
15	5	10	0.01	100	91	9513.2	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.0001	500	9	73.9876	1
2	10	100	0.0001	500	11	74.1087	1
3	10	100	0.0001	500	7	75.2000	1
4	10	100	0.0001	500	10	74.1989	1
5	10	100	0.0001	500	10	73.9998	1
6	10	100	0.001	500	11	73.9998	5
7	10	100	0.001	500	11	73.9998	5
8	10	100	0.001	500	11	74.1087	3
9	10	100	0.001	500	8	74.5939	5
10	10	100	0.001	500	10	73.9998	3
11	10	100	0.01	500	9	73.9876	10
12	10	100	0.01	500	8	73.9998	10
13	10	100	0.01	500	9	73.9876	10
14	10	100	0.01	500	9	73.9998	10
15	10	100	0.01	500	12	73.9998	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.0001	800	9	74.5939	2
2	20	500	0.0001	800	9	73.9998	2
3	20	500	0.0001	800	10	74.1989	2
4	20	500	0.0001	800	8	73.9998	2
5	20	500	0.0001	800	10	74.1989	2
6	20	500	0.001	800	10	73.9998	6
7	20	500	0.001	800	10	73.9998	14
8	20	500	0.001	800	8	73.9998	13
9	20	500	0.001	800	8	74.8701	2
10	20	500	0.001	800	8	74.0013	2
11	20	500	0.01	800	7	73.9876	20
12	20	500	0.01	800	6	73.9998	16
13	20	500	0.01	800	8	73.9876	20
14	20	500	0.01	800	9	73.9876	20
15	20	500	0.01	800	9	73.9876	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	27	74.87010	2
2	5	10	0.0001	100	52	73.98760	2
3	5	10	0.0001	100	43	75.08990	2
4	5	10	0.0001	100	15	75.20000	2
5	5	10	0.0001	100	32	73.99980	2
6	5	10	0.0001	500	20	74.10870	1
7	5	10	0.0001	500	29	74.96630	2
8	5	10	0.0001	500	29	73.98760	2
9	5	10	0.0001	500	17	73.98760	1
10	5	10	0.0001	500	63	73.99980	2
11	5	10	0.0001	800	18	74.90620	2
12	5	10	0.0001	800	25	74.46020	2
13	5	10	0.0001	800	20	74.59390	2
14	5	10	0.0001	800	17	74.10870	2
15	5	10	0.0001	800	32	74.19890	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.001	100	9	73.9998	3
2	10	100	0.001	100	8	74.8701	5
3	10	100	0.001	100	9	73.9998	3
4	10	100	0.001	100	8	74.5939	6
5	10	100	0.001	100	11	73.9998	10
6	10	100	0.001	500	11	73.9998	5
7	10	100	0.001	500	11	73.9998	5
8	10	100	0.001	500	11	74.1087	3
9	10	100	0.001	500	8	74.5939	5
10	10	100	0.001	500	10	73.9998	5
11	10	100	0.001	800	7	73.9998	2
12	10	100	0.001	800	8	73.9998	1
13	10	100	0.001	800	10	74.1087	5
14	10	100	0.001	800	5	73.9876	1
15	10	100	0.001	800	11	73.9998	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.1 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses16

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.01	100	8	73.9876	20
2	20	500	0.01	100	9	73.9998	20
3	20	500	0.01	100	9	73.9876	20
4	20	500	0.01	100	10	73.9876	20
5	20	500	0.01	100	8	73.9876	20
6	20	500	0.01	500	8	73.9998	20
7	20	500	0.01	500	7	73.9876	20
8	20	500	0.01	500	7	73.9876	20
9	20	500	0.01	500	9	73.9876	20
10	20	500	0.01	500	8	74.8701	18
11	20	500	0.01	800	7	73.9876	20
12	20	500	0.01	800	6	73.9998	16
13	20	500	0.01	800	8	73.9876	20
14	20	500	0.01	800	9	73.9876	20
15	20	500	0.01	800	9	73.9876	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	60	76.8593	1
2	5	10	0.0001	100	39	76.6493	2
3	5	10	0.0001	100	44	75.9343	2
4	5	10	0.0001	100	32	76.2815	2
5	5	10	0.0001	100	90	76.4845	2
6	10	10	0.0001	100	40	77.2782	4
7	10	10	0.0001	100	66	75.7948	2
8	10	10	0.0001	100	64	75.8645	1
9	10	10	0.0001	100	51	77.8999	2
10	10	10	0.0001	100	95	76.2563	2
11	20	10	0.0001	100	39	76.3261	2
12	20	10	0.0001	100	27	76.1915	2
13	20	10	0.0001	100	27	76.1857	2
14	20	10	0.0001	100	33	76.3194	2
15	20	10	0.0001	100	42	75.3984	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	100	0.001	500	17	76.1270	1
2	5	100	0.001	500	20	76.0949	2
3	5	100	0.001	500	18	77.1971	3
4	5	100	0.001	500	21	75.5088	2
5	5	100	0.001	500	45	76.8119	2
6	10	100	0.001	500	15	76.0279	4
7	10	100	0.001	500	20	75.7948	2
8	10	100	0.001	500	18	75.3097	1
9	10	100	0.001	500	18	75.7948	2
10	10	100	0.001	500	19	75.7948	2
11	20	100	0.001	500	22	75.3097	1
12	20	100	0.001	500	18	75.3097	2
13	20	100	0.001	500	16	76.6274	6
14	20	100	0.001	500	18	75.7948	5
15	20	100	0.001	500	17	75.7948	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	500	0.01	800	14	75.6666	5
2	5	500	0.01	800	14	76.1028	5
3	5	500	0.01	800	15	75.3097	5
4	5	500	0.01	800	13	75.7948	5
5	5	500	0.01	800	16	76.1046	5
6	10	500	0.01	800	15	75.5088	10
7	10	500	0.01	800	14	76.1046	10
8	10	500	0.01	800	15	75.5088	10
9	10	500	0.01	800	17	76.1028	10
10	10	500	0.01	800	13	76.0279	10
11	20	500	0.01	800	13	75.3097	18
12	20	500	0.01	800	13	76.0279	20
13	20	500	0.01	800	13	75.3097	20
14	20	500	0.01	800	15	75.3097	19
15	20	500	0.01	800	12	75.7948	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	60	76.8583	1
2	5	10	0.0001	100	39	76.0949	2
3	5	10	0.0001	100	44	77.1971	3
4	5	10	0.0001	100	32	75.5088	2
5	5	10	0.0001	100	90	76.8119	2
6	5	100	0.0001	100	17	77.2782	3
7	5	100	0.0001	100	17	78.0323	1
8	5	100	0.0001	100	19	76.9006	2
9	5	100	0.0001	100	16	77.1204	1
10	5	100	0.0001	100	23	76.7005	2
11	5	500	0.0001	100	19	76.8119	2
12	5	500	0.0001	100	16	76.3194	2
13	5	500	0.0001	100	15	77.2782	4
14	5	500	0.0001	100	17	75.3097	4
15	5	500	0.0001	100	16	77.0736	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	10	0.001	500	71	77.1797	2
2	10	10	0.001	500	169	76.0279	4
3	10	10	0.001	500	72	75.3097	2
4	10	10	0.001	500	70	75.3097	1
5	10	10	0.001	500	468	77.0997	4
6	10	100	0.001	500	14	76.1028	1
7	10	100	0.001	500	62	77.0997	6
8	10	100	0.001	500	18	75.5088	2
9	10	100	0.001	500	17	76.0949	1
10	10	100	0.001	500	15	76.0182	2
11	10	500	0.001	500	15	75.5088	2
12	10	500	0.001	500	14	78.3613	2
13	10	500	0.001	500	12	76.0279	2
14	10	500	0.001	500	14	75.9104	4
15	10	500	0.001	500	11	76.3926	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	10	0.01	800	61	75.3097	16
2	20	10	0.01	800	276	75.3984	8
3	20	10	0.01	800	760	75.3097	11
4	20	10	0.01	800	49	75.3097	13
5	20	10	0.01	800	345	75.3097	13
6	20	100	0.01	800	13	75.3097	18
7	20	100	0.01	800	172	75.4010	17
8	20	100	0.01	800	268	75.4010	20
9	20	100	0.01	800	23	75.3097	20
10	20	100	0.01	800	26	75.4010	19
11	20	500	0.01	800	13	75.3097	18
12	20	500	0.01	800	13	76.0279	20
13	20	500	0.01	800	13	75.3097	20
14	20	500	0.01	800	15	75.3097	19
15	20	500	0.01	800	12	75.7948	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	60	76.8583	1
2	5	10	0.0001	100	39	76.6493	2
3	5	10	0.0001	100	44	75.9343	2
4	5	10	0.0001	100	32	76.2815	2
5	5	10	0.0001	100	90	76.4845	2
6	5	10	0.001	100	38	76.0182	1
7	5	10	0.001	100	30	77.0126	2
8	5	10	0.001	100	28	77.0997	2
9	5	10	0.001	100	64	75.6666	3
10	5	10	0.001	100	85	76.6001	3
11	5	10	0.01	100	99	76.4423	5
12	5	10	0.01	100	41	75.8645	5
13	5	10	0.01	100	35	76.5765	5
14	5	10	0.01	100	87	76.9120	3
15	5	10	0.01	100	43	76.9531	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.0001	500	16	75.9343	1
2	10	100	0.0001	500	20	76.0949	1
3	10	100	0.0001	500	24	75.3097	2
4	10	100	0.0001	500	20	76.4317	1
5	10	100	0.0001	500	17	76.0279	1
6	10	100	0.001	500	14	76.1028	1
7	10	100	0.001	500	62	77.0997	6
8	10	100	0.001	500	18	75.5088	2
9	10	100	0.001	500	17	76.0949	1
10	10	100	0.001	500	15	76.0182	2
11	10	100	0.01	500	15	75.5088	10
12	10	100	0.01	500	19	75.5088	10
13	10	100	0.01	500	19	75.3097	10
14	10	100	0.01	500	309	75.5088	10
15	10	100	0.01	500	120	75.3097	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.0001	800	15	76.0182	2
2	20	500	0.0001	800	12	75.9104	1
3	20	500	0.0001	800	17	75.9104	4
4	20	500	0.0001	800	18	76.0182	1
5	20	500	0.0001	800	13	76.4272	1
6	20	500	0.001	800	14	75.3097	4
7	20	500	0.001	800	16	76.3261	4
8	20	500	0.001	800	16	75.3097	3
9	20	500	0.001	800	13	76.3261	4
10	20	500	0.001	800	14	76.4317	12
11	20	500	0.01	800	13	75.3097	18
12	20	500	0.01	800	13	76.0279	20
13	20	500	0.01	800	13	75.3097	20
14	20	500	0.01	800	15	75.3097	19
15	20	500	0.01	800	12	75.7948	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	60	76.8583	1
2	5	10	0.0001	100	39	76.6493	2
3	5	10	0.0001	100	44	75.9343	2
4	5	10	0.0001	100	32	76.2815	2
5	5	10	0.0001	100	90	76.4845	2
6	5	10	0.0001	500	50	77.1204	3
7	5	10	0.0001	500	54	75.7948	2
8	5	10	0.0001	500	82	76.1270	2
9	5	10	0.0001	500	48	79.9363	2
10	5	10	0.0001	500	163	75.3097	2
11	5	10	0.0001	800	293	77.0997	2
12	5	10	0.0001	800	42	80.1208	2
13	5	10	0.0001	800	162	79.3776	2
14	5	10	0.0001	800	51	77.1204	1
15	5	10	0.0001	800	38	77.1204	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.001	100	16	76.1028	3
2	10	100	0.001	100	25	76.5895	2
3	10	100	0.001	100	15	76.3261	2
4	10	100	0.001	100	13	77.2782	3
5	10	100	0.001	100	17	75.9104	3
6	10	100	0.001	500	15	76.0279	4
7	10	100	0.001	500	20	75.7948	2
8	10	100	0.001	500	18	75.3097	1
9	10	100	0.001	500	18	75.7948	2
10	10	100	0.001	500	19	75.7948	2
11	10	100	0.001	800	19	77.0997	7
12	10	100	0.001	800	19	76.9006	6
13	10	100	0.001	800	19	75.7948	2
14	10	100	0.001	800	18	75.3097	2
15	10	100	0.001	800	18	76.1334	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.2 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา ulysses22

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.01	100	14	76.1270	20
2	20	500	0.01	100	14	75.3097	20
3	20	500	0.01	100	14	75.3097	20
4	20	500	0.01	100	12	75.7948	20
5	20	500	0.01	100	11	75.3097	19
6	20	500	0.01	500	14	75.7948	20
7	20	500	0.01	500	13	75.3097	20
8	20	500	0.01	500	12	76.1028	16
9	20	500	0.01	500	13	75.3097	18
10	20	500	0.01	500	20	75.3097	20
11	20	500	0.01	800	13	75.3097	18
12	20	500	0.01	800	13	76.0279	20
13	20	500	0.01	800	13	75.3097	20
14	20	500	0.01	800	15	75.3097	19
15	20	500	0.01	800	12	75.7948	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	98	9693.8	1
2	5	10	0.0001	100	95	10161.0	1
3	5	10	0.0001	100	96	9963.7	2
4	5	10	0.0001	100	100	9229.8	2
5	5	10	0.0001	100	99	9852.0	1
6	10	10	0.0001	100	91	9954.6	2
7	10	10	0.0001	100	88	9309.9	2
8	10	10	0.0001	100	95	9622.3	2
9	10	10	0.0001	100	100	9554.6	2
10	10	10	0.0001	100	99	9588.1	2
11	20	10	0.0001	100	97	9121.8	2
12	20	10	0.0001	100	96	9078.0	2
13	20	10	0.0001	100	93	10936.0	2
14	20	10	0.0001	100	97	8846.1	2
15	20	10	0.0001	100	100	9697.1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	100	0.001	500	307	8123.0	2
2	5	100	0.001	500	56	9038.8	2
3	5	100	0.001	500	174	7816.3	1
4	5	100	0.001	500	78	8449.3	2
5	5	100	0.001	500	378	7794.3	1
6	10	100	0.001	500	144	8527.9	2
7	10	100	0.001	500	121	8713.0	4
8	10	100	0.001	500	67	8356.6	2
9	10	100	0.001	500	105	8344.7	2
10	10	100	0.001	500	96	8447.5	2
11	20	100	0.001	500	82	8406.9	1
12	20	100	0.001	500	347	9052.1	4
13	20	100	0.001	500	95	7881.7	1
14	20	100	0.001	500	63	8414.7	2
15	20	100	0.001	500	471	8832.2	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	500	0.01	800	54	8059.7	4
2	5	500	0.01	800	44	8338.0	5
3	5	500	0.01	800	89	8294.9	5
4	5	500	0.01	800	70	8150.0	3
5	5	500	0.01	800	51	7983.7	4
6	10	500	0.01	800	84	8159.3	1
7	10	500	0.01	800	47	8294.9	10
8	10	500	0.01	800	43	8304.0	3
9	10	500	0.01	800	353	7854.7	6
10	10	500	0.01	800	47	8021.0	6
11	20	500	0.01	800	42	8187.9	11
12	20	500	0.01	800	103	7858.4	12
13	20	500	0.01	800	108	8532.6	8
14	20	500	0.01	800	42	8430.2	20
15	20	500	0.01	800	53	8369.5	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	98	9693.8	1
2	5	10	0.0001	100	95	10161.0	1
3	5	10	0.0001	100	96	9963.7	2
4	5	10	0.0001	100	100	9229.8	2
5	5	10	0.0001	100	99	9852.0	1
6	5	100	0.0001	100	85	9295.4	2
7	5	100	0.0001	100	96	8471.0	1
8	5	100	0.0001	100	79	9043.5	1
9	5	100	0.0001	100	97	8296.8	2
10	5	100	0.0001	100	78	8499.7	1
11	5	500	0.0001	100	44	8164.2	1
12	5	500	0.0001	100	56	8207.8	1
13	5	500	0.0001	100	55	8277.7	1
14	5	500	0.0001	100	72	8039.9	1
15	5	500	0.0001	100	45	8602.9	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	10	0.001	500	467	8876.0	2
2	10	10	0.001	500	331	8663.1	2
3	10	10	0.001	500	327	8375.5	2
4	10	10	0.001	500	315	9375.1	2
5	10	10	0.001	500	295	8605.2	2
6	10	100	0.001	500	144	8527.9	2
7	10	100	0.001	500	121	8713.0	4
8	10	100	0.001	500	67	8356.6	2
9	10	100	0.001	500	105	8344.7	2
10	10	100	0.001	500	96	8447.5	2
11	10	500	0.001	500	56	8029.4	2
12	10	500	0.001	500	53	8077.7	2
13	10	500	0.001	500	161	7799.7	1
14	10	500	0.001	500	59	7690.2	2
15	10	500	0.001	500	45	8138.9	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	10	0.01	800	341	8207.2	7
2	20	10	0.01	800	290	8154.2	4
3	20	10	0.01	800	647	8113.5	6
4	20	10	0.01	800	642	8111.6	6
5	20	10	0.01	800	638	8118.5	6
6	20	100	0.01	800	103	7886.6	1
7	20	100	0.01	800	55	8201.4	10
8	20	100	0.01	800	263	8212.2	9
9	20	100	0.01	800	699	7745.6	4
10	20	100	0.01	800	140	8158.0	7
11	20	500	0.01	800	42	8187.9	11
12	20	500	0.01	800	103	7858.4	12
13	20	500	0.01	800	108	8532.6	8
14	20	500	0.01	800	42	8430.2	20
15	20	500	0.01	800	53	8369.5	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	98	9693.8	1
2	5	10	0.0001	100	95	10161.0	1
3	5	10	0.0001	100	96	9963.7	2
4	5	10	0.0001	100	94	9254.8	2
5	5	10	0.0001	100	99	9852.0	1
6	5	10	0.001	100	96	10800.0	2
7	5	10	0.001	100	99	9523.9	1
8	5	10	0.001	100	98	10196.0	2
9	5	10	0.001	100	99	9964.8	1
10	5	10	0.001	100	100	10065.0	2
11	5	10	0.01	100	100	9660.8	1
12	5	10	0.01	100	97	10192.0	1
13	5	10	0.01	100	100	9749.5	1
14	5	10	0.01	100	98	9824.6	2
15	5	10	0.01	100	91	9513.2	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.0001	500	352	9050.3	2
2	10	100	0.0001	500	142	8332.1	2
3	10	100	0.0001	500	95	8873.9	2
4	10	100	0.0001	500	164	8621.5	4
5	10	100	0.0001	500	92	8880.5	2
6	10	100	0.001	500	144	8527.9	2
7	10	100	0.001	500	121	8713.0	4
8	10	100	0.001	500	67	8356.6	2
9	10	100	0.001	500	105	8344.7	2
10	10	100	0.001	500	96	8447.5	2
11	10	100	0.01	500	70	7688.3	2
12	10	100	0.01	500	142	8209.0	4
13	10	100	0.01	500	169	8342.0	3
14	10	100	0.01	500	94	8375.2	4
15	10	100	0.01	500	342	7924.1	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.0001	800	41	8285.2	1
2	20	500	0.0001	800	51	7913.7	2
3	20	500	0.0001	800	50	8337.8	1
4	20	500	0.0001	800	51	7977.2	1
5	20	500	0.0001	800	82	8314.8	1
6	20	500	0.001	800	57	7643.4	2
7	20	500	0.001	800	69	8309.9	4
8	20	500	0.001	800	48	8777.4	2
9	20	500	0.001	800	54	8159.0	1
10	20	500	0.001	800	46	8203.6	1
11	20	500	0.01	800	42	8187.9	11
12	20	500	0.01	800	103	7858.4	12
13	20	500	0.01	800	108	8532.6	8
14	20	500	0.01	800	42	8430.2	20
15	20	500	0.01	800	53	8369.5	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	5	10	0.0001	100	98	9693.8	1
2	5	10	0.0001	100	95	10161.0	1
3	5	10	0.0001	100	96	9963.7	2
4	5	10	0.0001	100	100	9229.8	2
5	5	10	0.0001	100	99	9852.0	1
6	5	10	0.0001	500	484	8213.8	2
7	5	10	0.0001	500	491	8386.6	2
8	5	10	0.0001	500	207	8773.5	2
9	5	10	0.0001	500	358	8484.2	2
10	5	10	0.0001	500	456	8714.4	2
11	5	10	0.0001	800	663	9135.4	2
12	5	10	0.0001	800	451	8091.0	2
13	5	10	0.0001	800	324	8969.7	2
14	5	10	0.0001	800	773	9083.6	2
15	5	10	0.0001	800	622	8621.3	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	10	100	0.001	100	66	8797.6	2
2	10	100	0.001	100	65	8873.6	2
3	10	100	0.001	100	70	8737.1	3
4	10	100	0.001	100	63	8568.1	4
5	10	100	0.001	100	65	8312.0	2
6	10	100	0.001	500	144	8527.9	2
7	10	100	0.001	500	121	8713.0	4
8	10	100	0.001	500	67	8356.6	2
9	10	100	0.001	500	105	8344.7	2
10	10	100	0.001	500	96	8447.5	2
11	10	100	0.001	800	62	8510.5	4
12	10	100	0.001	800	325	8066.1	4
13	10	100	0.001	800	66	8299.6	2
14	10	100	0.001	800	103	8286.5	1
15	10	100	0.001	800	171	8620.8	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ผข.3 (ต่อ) ผลลัพธ์ของการทดลองกับตัวอย่างปัญหา berlin52

ครั้งที่	n	m	γ	t	รอบที่เจอผลลัพธ์ที่ดีที่สุดครั้งแรก	ผลลัพธ์	จำนวนหิ่งห้อยที่เคลื่อนที่แบบสุ่ม
1	20	500	0.01	100	48	8071.8	11
2	20	500	0.01	100	49	7765.4	13
3	20	500	0.01	100	45	7710.7	6
4	20	500	0.01	100	42	8293.4	14
5	20	500	0.01	100	46	7599.7	7
6	20	500	0.01	500	52	7945.6	13
7	20	500	0.01	500	61	7901.0	14
8	20	500	0.01	500	163	7941.6	7
9	20	500	0.01	500	38	8572.9	19
10	20	500	0.01	500	52	7849.7	9
11	20	500	0.01	800	42	8187.9	11
12	20	500	0.01	800	103	7858.4	12
13	20	500	0.01	800	108	8532.6	8
14	20	500	0.01	800	42	8430.2	20
15	20	500	0.01	800	53	8369.5	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้