การพัฒนาระบบจัดการงานสินไหมการประกันภัยรถยนต์ด้วยวิธีระบบอาณานิคมมด

DEVELOPMENT OF AUTO INSURANCE CLAIM MANAGEMENT SYSTEM USING ANT COLONY SYSTEM

เบญจมาภรณ์ ศรีอัมพร

BENJAMAPORN SRIAMPORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาคณิตศาสตร์ประยุกต์
ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2565

KMITL-2022-SC-M-001-083

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPMENT OF AUTO INSURANCE CLAIM MANAGEMENT SYSTEM USING ANT COLONY SYSTEM

BENJAMAPORN SRIAMPORN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE MASTER OF SCIENCE IN APPLIED MATHEMATICS

DEPARTMENT OF MATHEMATICS SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2022

KMITL-2022-SC-M-001-083

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2022
SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การพัฒนาระบบจัดการงานสินไหมการประกันภัยรถยนต์ด้วยวิธีระบบอาณานิคมมด
ชื่อนักศึกษา	นางสาวเบญจมาภรณ์ ศรีอัมพร
รหัสประจำตัว	62605021
ปริญญา	วิทยาศาสตรมหาบัณฑิต (คณิตศาสตร์ประยุกต์)
ภาควิชา	คณิตศาสตร์
พ.ศ.	2565
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผู้ช่วยศาสตราจารย์. ดร. บุษยมาส พิมพ์พรรณชาติ

บทคัดย่อ

ในปัจจุบันการประกันภัยรถยนต์มีบทบาทมากขึ้นจากรายงานภาวะธุรกิจประกันภัยไทย และเมื่อเกิดอุบัติเหตุเกี่ยวกับรถยนต์แล้วบริษัทประกันภัยได้รับแจ้งเหตุ บริษัทฯจะจัดส่งพนักงานออกตรวจสอบที่เกิดเหตุ ซึ่งหนึ่งในขั้นตอนการแจ้งเคลมเพื่อเรียกร้องค่าสินไหมทดแทน คือหากผู้เสียหายต้องการซ่อมรถที่สถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัย ผู้เสียหายต้องนำใบเสนอราคาค่าซ่อมที่ให้สถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัยประเมินราคามาติดต่อที่บริษัทประกันภัย เพื่อให้บริษัทฯจัดส่งพนักงานคุมราคาออกอนุมัติราคาให้ที่สถานบริการนั้น ๆ จากประเด็นดังกล่าวการจัดส่งพนักงานออกนอกสถานที่นั้นมีค่าใช้จ่ายในการดำเนินงาน ดังนั้นเพื่อเป็นแนวทางในการตัดสินใจให้เกิดการทำงานอย่างมีประสิทธิภาพในการเดินทางไปปฏิบัติงานของพนักงานคุมราคา ผู้วิจัยจึงสนใจนำมาพัฒนาเป็นระบบการจัดการงานสินไหมของการประกันภัยรถยนต์ ด้วยภาษาไพทอน (python) โดยให้ความสำคัญกับปัญหาการจัดเส้นทางการเดินรถเพื่อหาวิธีการจัดเส้นทางการเดินรถที่ทำให้ระยะทางรวมมีค่าน้อยที่สุด ด้วยการเปรียบเทียบ 3 วิธีในการจัดการปัญหาเส้นทางการเดินรถ ซึ่งกำหนดให้มีการค้นหาคำตอบจำนวน 100 คำตอบกับตัวอย่างกลุ่มข้อมูลสถานบริการซ่อมรถยนต์ในพื้นที่จังหวัดกรุงเทพมหานคร เพื่อเปรียบเทียบค่าเฉลี่ยของระยะทางรวมที่ได้ จากการทดสอบพบว่าเมื่อเปรียบเทียบค่าเฉลี่ยของระยะทางรวมทั้งสามวิธีจะเห็นว่าทั้งวิธีระบบอาณานิคมมดและวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง เป็นวิธีที่ให้ผลลัพธ์ดีกว่าวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตราทั้งสิ้น โดยวิธีที่ให้ผลลัพธ์ที่ดีที่สุดคือวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง ระบบดังกล่าวเมื่อถูกนำไปใช้จะช่วยให้พนักงานคุมราคาได้แนวทางในการตัดสินใจเพื่อให้เกิดการทำงานอย่างมีประสิทธิภาพรวมไปถึงสามารถช่วยวางแผนในการปฏิบัติงานในแต่ละวันได้

คำสำคัญ : การสลับสองตำแหน่ง, ปัญหาการจัดเส้นทางการเดินรถ, ระบบอาณานิคมมด, สินไหมทดแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Development of Auto Insurance Claim Management System using Ant Colony System
Student Name	Miss Benjamaporn Sriamporn
Student ID	62605021
Degree	Master of Science (Applied Mathematics)
Department	Mathematics
Year	2022
Thesis Advisor	Asst. Prof. Dr. Busayamas Pimpunchat

Abstract

At present, car insurance plays an increasingly important role from the Thai insurance business report. And when there is an accident involving a car, the insurance company has been notified of the incident. Insurance companies will send the surveyor to investigate the scene. which is one of the procedures for notifying a claim to make a claim. That is, the insured prefers to repair the damage with either non-contracted auto repair services garages or an authorized car dealer's garage, the insurance company will assign the claim officer to these garages to approve the repair cost proposed by them. From the above point, the dispatch of employees out of the premises has operating costs. For the purpose of as a guideline for decision making to work effectively for cost control surveying process. The researcher was interested in developing a claim management system for car insurance using python language. In this study, an important problem is to find a route to minimize the total distance. By there are three methods to compare the manage Vehicle Routing Problem. This required 100 iterations to be searched with ten sample of the local auto repair garage dataset in Bangkok to compare the average of the total distance. From the study, it was found that when comparing the average distances from all three methods, it was seen that both the traditional ant colony system methods and the modified ant colony system method with two-Opt that yields better results than Dykstra-based decision-making criteria. By the way, the best results are the modified ant colony system method with two-Opt. However, when such a system is implemented, will guide the surveyor decisions to work more efficiently, as well as help them to plan the day-to-day work.

Keywords: 2-Opt local search, Vehicle Routing Problem, Ant Colony System, claims.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผศ.ดร.บุษยมาส พิมพ์พรรณชาติ อาจารย์ที่ปรึกษาที่ได้ให้คำเสนอแนะการใช้โปรแกรม แนวคิดทางการศึกษาค้นคว้าหาความรู้ และตลอดจนแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอดจนวิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์ ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง และขอบพระคุณคณาจารย์ทุกท่านที่ได้กรุณาประสิทธิ์ประสาทวิชาความรู้อันถือเป็นพื้นฐานที่สำคัญยิ่งในการทำวิจัยครั้งนี้ ขอบคุณทางคณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มอบทุนการศึกษาให้ผู้วิจัย ขอบพระคุณ คุณพ่อ คุณแม่ ที่ให้กำลังใจในการศึกษา และให้การสนับสนุนในการทำวิจัยตลอดมา คุณค่า และประโยชน์อันพึงมีส่งเป็นผลจากงานวิจัยนี้ ขอมอบแด่ คุณพ่อ คุณแม่ ญาติพี่น้อง ครู อาจารย์ และผู้มีพระคุณทุกท่าน

เบญจมาภรณ์ ศรีอัมพร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 วิธีอาณานิคมมด (Ant Colony Optimization; ACO)	4
2.2 วิธีระบบอาณานิคมมดสำหรับแก้ปัญหาการจัดเส้นทางการเดินรถ	7
2.2.1 ขั้นตอนวิธีระบบหลายอาณานิคมมดสำหรับแก้ปัญหาการจัดเส้นทางการเดินรถ	7
2.3 วิธีการค้นหาเฉพาะที่ (Local Search)	15
2.4 อัลกอริทึมของไดก์สตรา (Dijkstra’s Algorithm)	17
บทที่ 3 วิธีการดำเนินงานวิจัย	18
3.1 ศึกษาและเก็บรวบรวมข้อมูล	18
3.2 ออกแบบระบบและกระบวนการค้นหาคำตอบด้วยวิธีต่างๆ	19
3.2.1 ออกแบบระบบ	19
3.2.2 กระบวนการค้นหาคำตอบด้วยวิธีต่าง ๆ	20
3.3 พัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนัก	25
บทที่ 4 ผลการวิจัยและการอภิปรายผล	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.1 ผลลัพธ์การเปรียบเทียบ	26
4.2 ผลการพัฒนาระบบ	29
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	34
5.1 สรุปผลการวิจัย	34
5.2 ข้อเสนอแนะ	34
เอกสารอ้างอิง	35
ภาคผนวก	37
ภาคผนวก ก	38
ข้อมูลและโค้ดในการพัฒนาระบบ	38
ภาคผนวก ข	52
เปเปอร์ที่ได้รับการตีพิมพ์	52
ประวัติผู้เขียน	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	ระยะทางการเดินทางระหว่างแต่ละเมือง	9
2.2	คำตอบที่ได้จากการเริ่มต้นที่แต่ละเมืองและใช้วิธีการเลือกเมืองที่ใกล้ที่สุดเป็นลำดับถัดไป (Nearest Neighborhood (NN)) ในการสร้างคำตอบ	10
2.3	แผนการผลิตของสินค้าชนิดหนึ่งในตัวอย่างที่ 2	13
2.4	แผนการผลิตใหม่หลังจากดำเนินการด้วยวิธีสลับสองตำแหน่ง	14
3.1	รายชื่อเขตทั้งหมดในกรุงเทพมหานคร (เรียงตามรหัสเขตการปกครองที่ใช้ในราชการ)	19
3.2	ตัวอย่าง Distance Matrix (ระยะทางเป็น กิโลเมตร)	19
3.2	การออกแบบระบบ	20
4.1	ค่าพารามิเตอร์ต่างๆ	27
4.2	ข้อมูลระยะทางของปัญหาที่สุ่มมาทดสอบในระบบ	27
4.3	ผลลัพธ์ด้วยวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา	28
4.4	ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับต่ำ (Low threshold setting)	29
4.5	ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับกลาง (Medium threshold setting)	29
4.6	ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับสูง (High threshold setting)	29
ก.1	ข้อมูลตำแหน่งอู่ซ่อมรถยนต์ที่ใช้ในการทดสอบระบบ	39
ก.2	ข้อมูลระยะทางระหว่างอู่ซ่อมรถยนต์ที่ใช้ในการทดสอบระบบ	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	ตัวอย่างเส้นทางการเดินทางเพื่อหาอาหารของมด F คือ แหล่งอาหาร และ N คือรังมด	3
2.2	ตัวอย่างปัญหาการจัดเส้นทางการเดินรถ เส้นทางที่สามารถเดินทางได้ (ซ้าย) และเมื่อจัดเส้นทางแล้ว (ขวา)	4
2.3	กระบวนการของการหาค่าเหมาะสมด้วยวิธีการอาณานิคมมด [7]	6
2.4	เส้นทางการเดินก่อนทำวิธีสลับสองตำแหน่ง	12
2.5	เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง	12
2.6	เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง	12
2.7	ค่า Y_t ก่อนทำวิธีสลับสองตำแหน่ง	13
2.8	เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง	13
2.9	การปรับปรุงคำตอบด้วยวิธี One-Move	16
2.10	การปรับปรุงคำตอบด้วยวิธี Crossover-Move	16
2.11	ขั้นตอนวิธีหาวิถีสั้นที่สุดด้วยอัลกอริทึมของไดก์สตรา	17
3.1	ตัวอย่างตำแหน่งอู่ซ่อมรถยนต์ในจังหวัดกรุงเทพมหานคร	18
3.2	ตัวอย่างวิธีสลับสองตำแหน่ง	22
4.1	หน้าโปรแกรมระบบการจัดการงานสินไหมในกรณีความเสียหายหนัก	30
4.2	แผนภูมิระบบการจัดการสินไหมกรณีความเสียหายหนักด้วยวิธีระบบอาณานิคมมด ที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง	30
4.3	หน้าต่างโปรแกรม I.C.M System (1)	31
4.4	หน้าต่างการตั้งค่า	31
4.5	หน้าต่างโปรแกรม I.C.M System (2)	32
4.6	หน้าต่างโปรแกรม I.C.M System หลังประมวลผลเสร็จ	33
4.7	หน้าต่างสำหรับแสดงแผนที่เส้นทางการเดินทาง	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
ก.1	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 1 library และการดึงข้อมูล	41
ก.2	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 2 อัลกอริทึมสำหรับประมวลผล	42
ก.3	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 3 อัลกอริทึมสำหรับประมวลผล (ต่อ)	43
ก.4	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 4 การรับค่าจากหน้าต่างโปรแกรม	44
ก.5	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 5 การสร้างหน้าต่างโปรแกรม	45
ก.6	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 6 การสร้างหน้าต่างโปรแกรมและแสดงผลลัพธ์	46
ก.7	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 7 หน้าต่างแสดงผลลัพธ์	47
ก.8	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 8 การสร้างหน้าต่างตั้งค่าตัวแปร	48
ก.9	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 9 การแสดงผลลัพธ์แบบตาราง	49
ก.10	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 10 การแสดงผลลัพธ์แบบตาราง (ต่อ)	50
ก.11	โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 11 การสร้างปุ่มและช่องรับค่าของหน้าต่างโปรแกรม	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

การประกันภัยรถยนต์มีบทบาทมากขึ้นในปัจจุบัน จากรายงานภาวะธุรกิจประกันภัยไทย ประจำไตรมาส 1 ปี 2564 พบว่าประกันรถยนต์ภาคสมัครใจปรับตัวเพิ่มขึ้นมากกว่าช่วงเดียวกันของปีก่อน 2.68% [1] แสดงให้เห็นว่าการทำประกันภัยรถยนต์มีความสำคัญมากขึ้น และเมื่อเกิดอุบัติเหตุเกี่ยวกับรถยนต์แบบมีคู่กรณี ที่เรียกว่า “เคลมสด”รถประกัน หรือรถคู่กรณีมีความเสียหายมาก จนไม่สามารถขับเคลื่อนได้ เมื่อบริษัทประกันภัยได้รับแจ้งเหตุดังกล่าวบริษัทฯ จะจัดส่งพนักงานออกตรวจสอบ ณ ที่เกิดเหตุ โดยผู้ขับขี่รถประกันหรือรถคู่กรณีจะได้รับใบหลักฐานในการติดต่อค่าสินไหมจากพนักงานตรวจสอบอุบัติเหตุ หลังจากนั้นหากผู้เสียหายต้องการซ่อมรถกับอู่ หรือสถานบริการในเครือของบริษัทประกันภัย บริษัทจะดำเนินการประเมินราคาและส่งรถเข้าสถานบริการเพื่อดำเนินการซ่อมได้เลย แต่หากผู้เสียหายต้องการซ่อมห้าง คือผู้เอาประกันหรือเจ้าของรถต้องการซ่อมรถที่อู่ หรือสถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัย ผู้เอาประกันต้องนำใบเสนอราคาค่าซ่อมที่ให้อู่ หรือสถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัยประเมินราคามาติดต่อที่บริษัทประกันภัย เพื่อให้บริษัทฯ จัดส่งพนักงานคุมราคาออกอนุมัติราคาให้ที่สถานบริการนั้นๆ [2] โดยทั้งหมดที่กล่าวมานั้นเป็นหนึ่งในขั้นตอนการแจ้งเคลม ทั้งนี้บริษัทประกันภัยได้จัดเก็บข้อมูลผู้เอาประกัน ผู้ขับขี่ รายละเอียดการเกิดอุบัติเหตุต่าง ๆ รวมถึงบันทึกรายการค่าสินไหม ผ่านใบรายการความเสียหายจากขั้นตอนข้างต้น ผู้วิจัยจึงเล็งเห็นว่าข้อมูลดังกล่าวมีความสำคัญที่สามารถนำไปวิเคราะห์ในด้านอื่นๆเพิ่มเติมได้ ซึ่งมองการพิจารณาสินไหมเป็น 2 ส่วน คือ ส่วนงานพิจารณาสินไหมความเสียหายที่ต่ำกว่า 100,000 บาท และส่วนงานพิจารณาสินไหมความเสียหายเกินกว่า 100,000 บาท ส่วนงานพิจารณาสินไหมความเสียหายเกินกว่า 100,000 บาท เป็นสินไหมกรณีความเสียหายหนักเนื่องจากบริษัทฯ ต้องจัดส่งพนักงานคุมราคาออกอนุมัติราคาที่สถานบริการนั้น ๆ ซึ่งการจัดส่งพนักงานออกนอกสถานที่นั้นมีค่าใช้จ่ายในการดำเนินงานผู้วิจัยจึงสนใจนำมาสร้างระบบการจัดการงานสินไหมในกรณีความเสียหายหนักของการประกันภัยรถยนต์

งานวิจัยนี้จึงได้พัฒนาระบบในการจัดการงานและหาเส้นทางที่เหมาะสมในการเดินทางไปยังสถานบริการซ่อมรถยนต์ให้เหมาะสมกับปริมาณใน 1วัน เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ จากปัญหาที่กล่าวมาข้างต้น ผู้วิจัยจึงให้ความสำคัญกับปัญหาการจัดเส้นทางการเดินรถเพื่อหาวิธีการจัดเส้นทางการเดินรถที่ทำให้ระยะทางรวมมีค่าน้อยที่สุด ซึ่งปัญหานี้สามารถแก้ได้โดยใช้ แบบจำลองเชิงคณิตศาสตร์ (Mathematical Programming Model) แต่ยังคงประสบปัญหาที่ต้องใช้ เวลาในการคำนวณนานโดยเฉพาะกับปัญหาที่มีขนาดใหญ่ ดังนั้นในทางปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงนิยมแก้ปัญหาด้วยวิธีฮิวริสติกส์ (Heuristic Method) ซึ่งเป็นวิธีที่ใช้สามัญสำนึกและ สามารถหาคำตอบในเวลาอันรวดเร็วถึงแม้อาจไม่ได้คำตอบที่ดีที่สุดก็ตาม ตัวอย่างของวิธีฮิวริสติกส์ได้แก่ ขั้นตอนวิธีการเชิงพันธุกรรม (Genetic Algorithm; GA) ขั้นตอนวิธีการเลียนแบบการอบอ่อน (Simulated Annealing Algorithm; SA) และขั้นตอนวิธีการอาณานิคมมด (Ant Colony Optimization; ACO) เป็นต้น โดยอุดม จันทร์จรัสสุข และ กิตติโรจน์ สันติฐายี. [3] กล่าวว่าขั้นตอนอาณานิคมมดเป็นวิธีที่เลียนแบบพฤติกรรมการหาอาหารของมด ซึ่งได้ถูกนำมาใช้ในการแก้ปัญหาเชิงการจัด (Combinatorial Optimization Problem) ได้หลากหลายและได้รับความนิยมในปัจจุบัน เนื่องจากสามารถให้ผลลัพธ์ที่ดี

ดังนั้น ในงานวิจัยนี้ ผู้วิจัยจึงได้นำเสนอวิธีระบบอาณานิคมมด (Ant Colony System; ACS) โดยนำมาใช้ในการแก้ปัญหาการจัดเส้นทางการเดินรถพร้อมทั้งปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง (2-Opt)

1.2 วัตถุประสงค์ของงานวิจัย

พัฒนาระบบจัดการงานสินไหมในการประกันภัยรถยนต์ด้วยการนำวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง มาประยุกต์ใช้ในการแก้ปัญหาการจัดเส้นทางการเดินรถด้วยภาษาไพทอน (python)

1.3 ขอบเขตของงานวิจัย

ข้อมูลสถานบริการซ่อมรถยนต์ในจังหวัดกรุงเทพมหานครด้วยการสุ่มเก็บตำแหน่งของข้อมูล (latitude, longitude) อู่ซ่อมรถยนต์รวมถึงระยะทางระหว่างตำแหน่งทั้งหมด (กิโลเมตร) จำนวน 21 ข้อมูลที่รวมจุดเริ่มต้นซึ่งกำหนดให้เป็น สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อใช้เป็นข้อมูลพื้นฐานในการนำมาทดสอบกับระบบจัดการงานสินไหมในการประกันภัยรถยนต์ด้วยวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

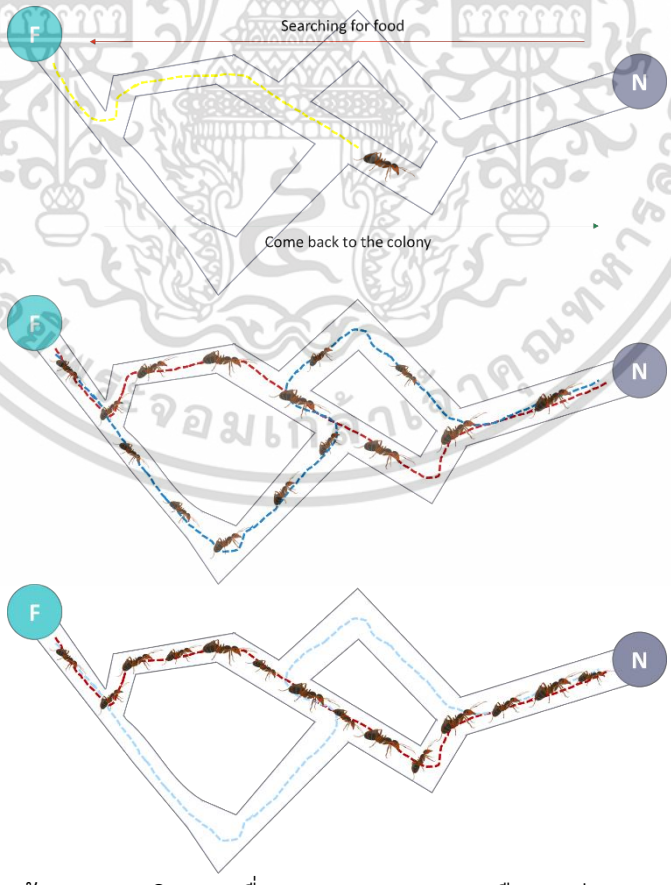
1.) ได้ระบบจัดการงานสินไหมในการประกันภัยรถยนต์ด้วยการนำวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง ที่สามารถแนะนำเส้นทางเพื่อเป็นแนวทางประกอบการตัดสินใจ และวางแผนในการเดินทางของพนักงาน

2.) เพื่อเป็นแนวทางในการตัดสินใจให้เกิดการทำงานอย่างมีประสิทธิภาพในการเดินทางไปปฏิบัติงานด้วยรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

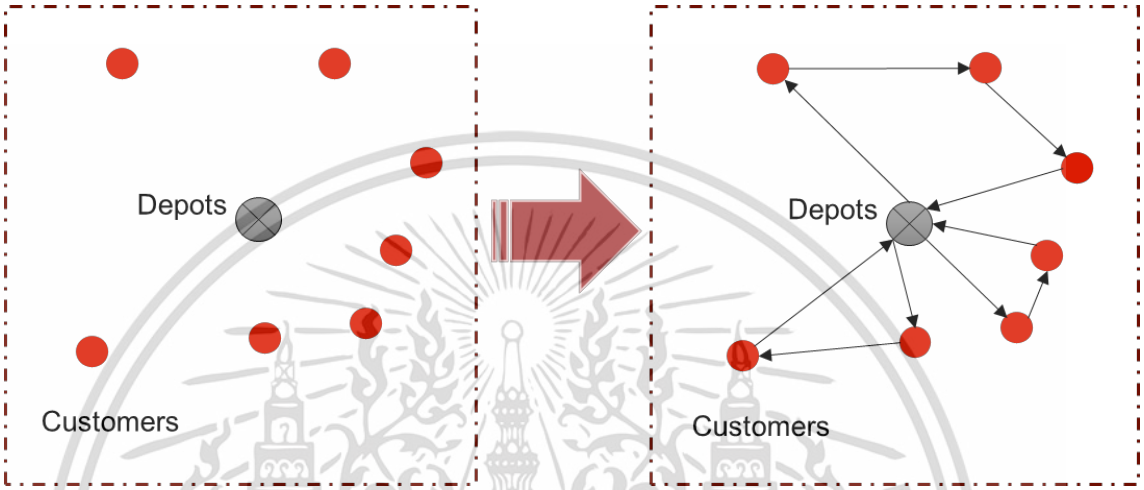
ในบทนี้จะกล่าวถึงขั้นตอนวิธีระบบอาณานิคมมดรวมไปถึงขั้นตอนการปรับปรุงคำตอบด้วยวิธีต่าง ๆ และงานวิจัยทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้ขั้นตอนวิธีดังกล่าวในการแก้ปัญหาการจัดเส้นทางการเดินรถ ซึ่งวิธีการระบบมดหรืออาณานิคมมดเป็นวิธีการเลียนแบบพฤติกรรมจริงของมดที่ออกไปหาอาหาร โดยมดจะใช้วิธีการเดินทางแบบสุ่มจากรังไปสู่แหล่งอาหาร และกลับมาที่รังอีกครั้งหลังจากได้อาหารแล้ว ซึ่งในระหว่างเดินทางมดจะปล่อยสารเคมีชนิดหนึ่งที่เรียกว่า ฟีโรโมน (Pheromone) เพื่อให้มดตัวอื่น ๆ สามารถเดินทางตามกลิ่นหรือร่องรอยของฟีโรโมนนั้น ๆ ไปยังแหล่งอาหารได้ เมื่อมดตัวอื่น ๆไปยังแหล่งอาหารก็จะมีการทิ้งสารฟีโรโมนเพิ่มเติมบนเส้นทางนั้น ๆที่เดินผ่าน ดังนั้นเส้นทางที่สั้นกว่าจะมีความเข้มข้นของฟีโรโมนที่สูงกว่า เนื่องจากมีการเพิ่มสารฟีโรโมนที่บ่อยกว่า และเมื่อเวลาผ่านไปฟีโรโมนบนเส้นทางที่ยาวกว่าก็จะค่อย ๆระเหยหรือจางหายไปทำให้เหลือเส้นทางที่มีระยะทางสั้นที่สุด ซึ่งตัวอย่างเส้นทางการเดินทางเพื่อหาอาหารของมดสามารถเห็นได้ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างเส้นทางการเดินทางเพื่อหาอาหารของมด F คือ แหล่งอาหาร และ N คือรังมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาการจัดเส้นทางการเดินรถ (Vehicle Routing Problem; VRP) เป็นปัญหาที่เกี่ยวกับการจัดเส้นทางการขนส่งสินค้าจากศูนย์กระจายสินค้าไปยังลูกค้าที่กระจายตามจุดต่าง ๆ [4] ภายใต้เงื่อนไขหรือข้อจำกัดในการขนส่ง เช่น ความจุของรถบรรทุก กรอบเวลาในการส่งสินค้า เป็นต้น โดยมีวัตถุประสงค์เพื่อหาเส้นทางการเดินรถที่สั้นที่สุด หรือมีต้นทุนการขนส่งที่น้อยที่สุด โดยตัวอย่างปัญหาการจัดเส้นทางการเดินรถสามารถเห็นได้ดังรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างปัญหาการจัดเส้นทางการเดินรถ เส้นทางที่สามารถเดินทางได้ (ซ้าย) และเมื่อจัดเส้นทางแล้ว (ขวา)

2.1 วิธีอาณานิคมมด (Ant Colony Optimization; ACO)

ขั้นตอนวิธีอาณานิคมมด (Ant Colony Optimization; ACO) ได้ถูกนำเสนอโดย Colorni และ Dorigo [5] ในการแก้ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem; TSP) โดยใช้มดในการหาเส้นทางที่สั้นที่สุดสามารถสังเกตได้จากรูปที่ 2.3 ซึ่งกระบวนการหาคำตอบจะอาศัยการทำซ้ำโดยเส้นทางที่ดีจากรอบการทำซ้ำก่อนหน้าจะถูกเพิ่มฟีโรโมนให้มีความเข้มมากขึ้นเพื่อเป็นข้อมูลช่วยให้มดตัวอื่น ๆ หาเส้นทางได้ดีขึ้นในรอบถัดไป ในขณะเดียวกันก็จะลดฟีโรโมนบนเส้นทางทั้งหมด เพื่อหลีกเลี่ยงโอกาสในการได้เส้นทางที่ไม่ดี โดย [6] มีวิธีการขั้นพื้นฐานของวิธีการอาณานิคมมดโดยทั่วไปประกอบด้วยขั้นตอนง่าย ๆ ดังนี้

1. เริ่มต้นตั้งค่าพารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องกับวิธีอาณานิคมมดทั้งหมด
2. สร้างคำตอบเริ่มต้นจากพารามิเตอร์เริ่มต้นที่ตั้งไว้
3. ปรับค่าพารามิเตอร์ต่างๆ ในข้อที่ 1
4. วนซ้ำข้อ 2-3 จนกระทั่งหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ที่ต้องตั้งค่าไว้ก่อนการเริ่มกระบวนการมี ดังนี้

1. จำนวนรอบที่ต้องการวนซ้ำ (Iteration) จำนวนรอบในการวนซ้ำหาคำตอบแต่ละรอบประกอบด้วยมด 1 ฝูงในการหาคำตอบ

2. จำนวนประชากรมด (Ant Population) ใน 1 ฝูง จำนวนมดใน 1 ฝูงถ้าในปัญหา TSP หมายถึง ใน 1 รอบของการเลือกเส้นทางจะประกอบไปด้วยจำนวนกี่คำตอบหรือกี่เส้นทาง จำนวนเส้นทางจะมีค่าเท่ากับจำนวนประชากรทั้งหมดใน 1 ฝูงมด

3. ค่าฟีโรโมน (Pheromone) ทำหน้าที่เหมือนกับฟีโรโมนของมด และมีหน้าที่เหมือนกันคือทำหน้าที่เพิ่มความน่าสนใจของเส้นทางที่มดตัวต่อ ๆ ไปจะเดินตามเส้นทางของมดตัวที่เดินไปก่อนหน้า

4. ค่าความน่าสนใจ (Attractiveness : $\omega(i,j)$) เป็นผลคูณของค่าฟีโรโมนของการเชื่อมเส้นทางระหว่างเมืองหนึ่งกับอีกเมืองหนึ่งกับระยะทางระหว่างเส้นทางเชื่อมสองเมืองนั้น ในปัญหา TSP ค่าความน่าสนใจนี้จะมี 2 พารามิเตอร์ย่อยคือ แอลฟา (α) และบีตา (β) ซึ่งแอลฟาและบีตาจะเป็นค่าคงที่ค่าหนึ่งที่ระบุถึงการให้น้ำหนักกับค่าฟีโรโมนกับค่าระยะทางระหว่างการเชื่อมของเมืองสองเมือง โดยทั่วไปแอลฟาและบีตาจะถูกตั้งค่าเป็นค่าจำนวนเต็มที่มีค่าระหว่าง 1-5 สมการสำหรับคำนวณหาค่าความน่าสนใจจะแสดงได้ดังสมการที่ 2.1

$$\omega(i,j) = \tau_{i,j}^{\alpha} \times \pi_{i,j}^{\beta} \qquad (2.1)$$

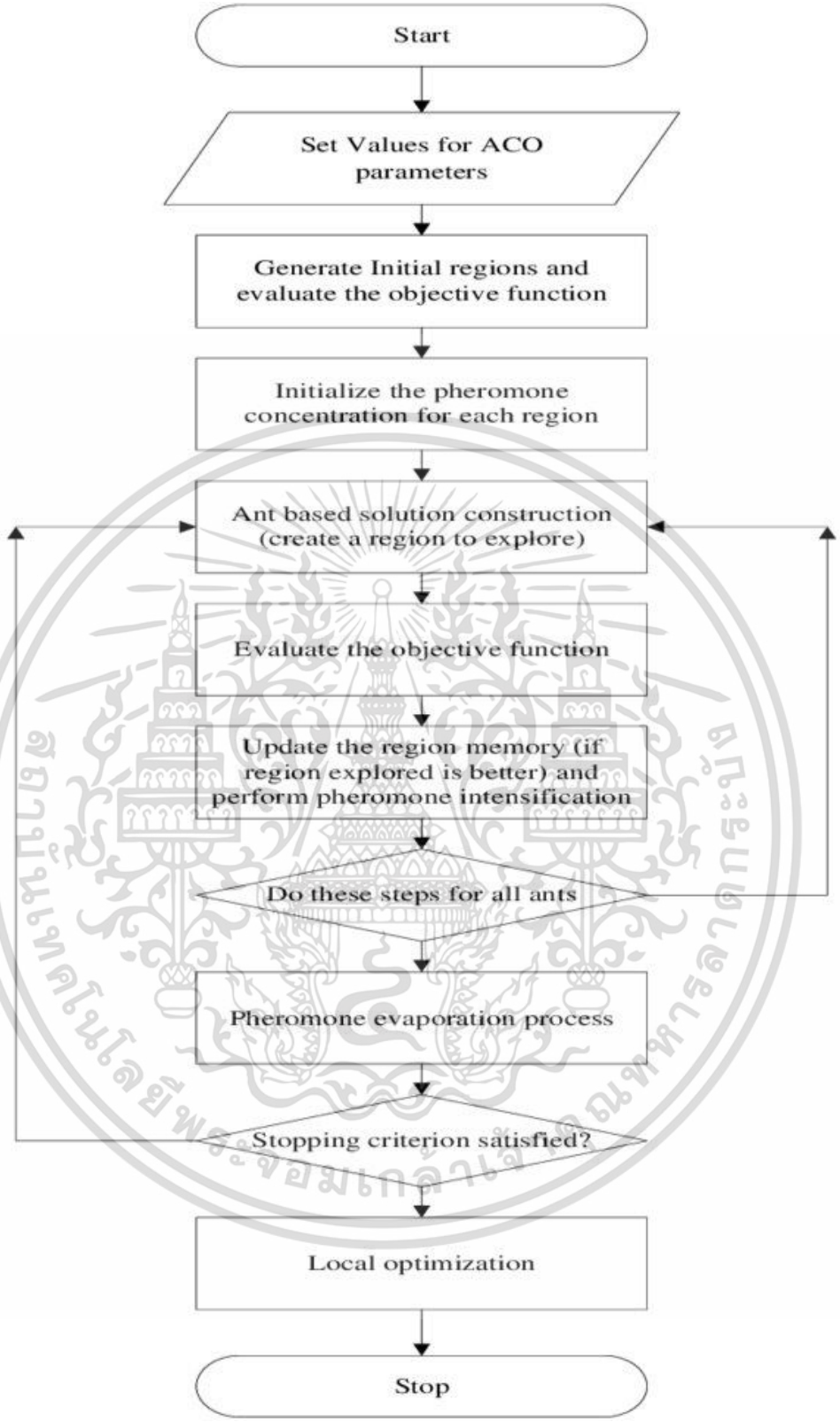
โดย $\omega(i,j)$ คือค่าความน่าสนใจของการเดินทางจากเมือง i ไปเมือง j

$\tau_{i,j}^{\alpha}$ คือค่าฟีโรโมนของการเดินทางระหว่างเมือง i ไปเมือง j ที่ให้น้ำหนัก(ยกกำลัง)ด้วยค่าน้ำหนักของฟีโรโมน

$\pi_{i,j}^{\beta}$ คือค่าคงที่ค่าหนึ่งในปัญหา TSP เป็นส่วนกลับของระยะห่างระหว่างเมืองสองเมือง ค่านี้จะให้น้ำหนักเป็นบีตา

$\pi = \dfrac{1}{d_{i,j}}$ ถ้า $d_{i,j}$ คือระยะห่างระหว่างเมือง i ไปเมือง j

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 กระบวนการของการหาค่าเหมาะสมด้วยวิธีการอาณานิคมมด [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 วิธีระบบอาณานิคมมดสำหรับแก้ปัญหาการจัดเส้นทางการเดินรถ

ปัญหาการจัดเส้นทางการเดินรถเป็นปัญหาเชิงการจัดที่มีความซับซ้อนในระดับ NP-Hard [8] และยากต่อการหาคำตอบที่เหมาะที่สุด (Optimal Solution) การแก้ปัญหาสามารถทำได้หลายวิธี ขึ้นอยู่กับรูปแบบของปัญหาที่ทำการศึกษา หรือลักษณะของวิธีที่ใช้แก้ปัญหา โดย Zanakis et al. [9] ได้แนะนำให้ใช้วิธีฮิวริสติกส์ในการหาคำตอบแทนการใช้วิธีแม่นตรงสำหรับแก้ปัญหาที่มีขนาดใหญ่ และต้องการคำตอบในเวลาอันสั้น Bullnheimer et al. [10] ได้ประยุกต์ใช้วิธีระบบมด (Ant System; AS) ในการแก้ปัญหาการจัดเส้นทางการเดินรถ แล้วการเปรียบเทียบประสิทธิภาพกับวิธีการจัดเส้นทาง ทั้งหมด 3 วิธีคือ 1. วิธีค้นหาแบบทาบู(Tabu Search) 2. วิธีการจำลองการอบอ่อน (Simulated Annealing) และ 3.วิธีโครงข่ายประสาทเทียม (Neural Network) [1] ได้สรุปว่า จากผลการเปรียบเทียบพบว่าวิธีระบบมดสามารถให้คำตอบที่ดีโดยมีค่าเฉลี่ยของเปอร์เซ็นต์ความแตกต่างจากคำตอบที่ เหมาะที่สุดเท่ากับ 4.43 เปอร์เซ็นต์ [3] กล่าวว่าGambardella et al. [11] ได้ประยุกต์ใช้วิธีระบบอาณานิคมมดหลายกลุ่ม (Multiple Ant Colony System; MACS) สำหรับแก้ปัญหาการกำหนดเส้นทางยานพาหนะแบบมีกรอบเวลา (Vehicle Routing Problem with Time Window; VRPTW) จำนวน 17 ปัญหา ด้วยฟังก์ชันวัตถุประสงค์สองประการคือ การลดจำนวนทัวร์ (หรือยานพาหนะ) และ การลดเวลาการเดินทางทั้งหมดให้เหลือน้อยที่สุดโดยที่จำนวนทัวร์ลดลง โดยได้แบ่งมดออกเป็นสองกลุ่มสำหรับหาคำตอบ มดกลุ่มแรกใช้ในการหาจำนวน รถบรรทุกที่น้อยสุดที่ใช้ในการส่งสินค้าและ ใช้มดกลุ่มที่สองในการหาระยะทางการเดินรถที่น้อยที่สุด ซึ่งผลที่ได้ พบว่าขั้นตอนวิธีที่นำเสนอให้คำตอบที่ดีกว่างานวิจัยของShaw [12] Rochart et al. [13] และ Taillard et al.[14] จำนวน 14 ปัญหา

2.2.1 ขั้นตอนวิธีระบบหลายอาณานิคมมดสำหรับแก้ปัญหาการจัดเส้นทางการเดินรถ

ระบบหลายอาณานิคมมดที่นำเสนอมีการใช้มดหลายกลุ่มในการช่วยหาคำตอบโดยเก็บค่าฟีโรโมนเฉพาะที่ (Local Pheromone) แยกตามกลุ่มมด และใช้ฟีโรโมนส่วนกลาง (Global Pheromone) ในการแบ่งปันข้อมูลระหว่างกลุ่ม ระบบอาณานิคมมดใช้วิธีค้นหาเพื่อนบ้านใกล้สุด (Nearest neighbor search) ในการหาคำตอบเริ่มต้น เพื่อช่วยให้มดสามารถหาคำตอบที่ดีได้เร็วขึ้น การหา คำตอบของระบบอาณานิคมมดจะแบ่งมดออกเป็น หลายกลุ่ม โดยมดในแต่ละกลุ่มจะหาคำตอบที่เป็นไปได้ จากค่าความน่าจะเป็นซึ่งคำนวณจากค่าของฟีโรโมนและ ค่าทัศนวิสัยของมด ในแต่ละรอบของการค้นหาคำตอบ คำตอบของมดที่ดีที่สุดในแต่ละกลุ่มจะถูกนำไปปรับปรุง ด้วยวิธีสลับสองตำแหน่ง แล้วนำคำตอบที่ได้ไปปรับค่าฟีโรโมนเฉพาะที่ ส่วนคำตอบที่ดีที่สุดจากมดทุก กลุ่มจะถูกนำไปปรับค่าฟีโรโมนส่วนกลาง การหาคำตอบ จะทำซ้ำจนกว่าจะครบรอบที่กำหนดไว้แล้วนำคำตอบที่ดี ที่สุดไปใช้จัดเส้นทางการเดินรถ รายละเอียดของขั้นตอน การแก้ปัญหาการจัดเส้นทางการเดินรถด้วยระบบหลายอาณานิคมมดได้แสดงไว้ดังนี้ [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 กำหนดค่าพารามิเตอร์และค่าเริ่มต้นของฟีโรโมน ส่วนกลางและฟีโรโมนเฉพาะที่
ขั้นตอนที่ 2 หาคำตอบเริ่มต้นด้วยวิธีค้นหาเพื่อนบ้านใกล้สุด
ขั้นตอนที่ 3 เมื่อยังไม่ถึงจำนวนรอบที่กำหนด

- สำหรับมดแต่ละกลุ่มทำขั้นตอนย่อยต่อไปนี้
    1. ค้นหาเส้นทางด้วยวิธีอาณานิคมมด
    2. นำคำตอบที่ดีที่สุดไปปรับปรุงด้วยวิธีสลับสองตำแหน่ง
    3. นำคำตอบที่ปรับปรุงแล้วไปปรับค่าฟีโรโมนเฉพาะที่
- คัดเลือกคำตอบที่ดีที่สุดจากมดทุกกลุ่มเพื่อนำไปปรับค่าฟีโรโมนส่วนกลาง

ขั้นตอนที่4 ส่งค่าคำตอบที่ดีที่สุด

2.2.1.1 การกำหนดค่าฟีโรโมน

การกำหนดค่าฟีโรโมน เป็นสิ่งที่สำคัญสำหรับใช้เป็นข้อมูลในการหาคำตอบ โดยค่าฟีโรโมนที่มดใช้หาคำตอบ จะมาจากค่าฟีโรโมนเฉพาะที่ และค่าฟีโรโมนส่วนกลาง ซึ่งสามารถคำนวณได้จากสมการที่ (2.2) ค่าฟีโรโมนจะถูกเก็บไว้ในเมทริกซ์ขนาด $(n+1)\times(n+1)$ โดยที่ n คือจำนวนลูกค้าทั้งหมด เรียกเมทริกซ์นี้ว่า ฟีโรโมนเมทริกซ์ ซึ่งสามารถเขียนแทนด้วย $[\tau_{ij}]$ โดยที่ $i, j \in \{0,...,n\}$ เมื่อ 0 คือดัชนีของศูนย์กระจายสินค้า ในช่วงเริ่มต้นค่าฟีโรโมนส่วนกลาง และค่าฟีโรโมนเฉพาะที่ แต่ละค่าจะถูกกำหนดให้มีค่าเท่ากันทั้งหมด

$$\tau_{i,j} = \lambda\tau^{g}_{i,j} + (1-\lambda)\tau^{l}_{i,j} \qquad ;\forall l \in \{1,...,L\} \tag{2.2}$$

โดย τ^{g}_{ij} คือค่าฟีโรโมนส่วนกลางระหว่างลูกค้า i กับลูกค้า j

τ^{l}_{ij} คือค่าฟีโรโมนเฉพาะที่ระหว่างลูกค้า i กับลูกค้า j ของมดกลุ่มที่ l

L คือจำนวนกลุ่มของมดทั้งหมด

λ คือสัดส่วนระหว่างการใช้ค่าฟีโรโมนส่วนกลาง และค่าฟีโรโมนเฉพาะที่ ซึ่งมีค่าอยู่ระหว่าง [0,1]

2.2.1.2 การหาคำตอบเริ่มต้นด้วยวิธีค้นหาเพื่อนบ้านใกล้สุด

การหาคำตอบเริ่มต้นด้วยวิธีค้นหาเพื่อนบ้านใกล้สุดเป็นวิธีการค้นหาเส้นทางโดยพิจารณาจากระยะทางที่น้อยที่สุดของลูกค้าข้างเคียงจากตำแหน่งลูกค้าปัจจุบัน โดยการค้นหาเส้นทางของรถแต่ละคันจะเริ่มจากศูนย์กระจายสินค้าไปส่งสินค้าให้กับลูกค้าทีละรายจนกว่าปริมาณสินค้าที่เหลือมีไม่เพียงพอ หรือเมื่อลูกค้าได้รับสินค้าครบทุกราย แล้วจึงกลับมาที่ศูนย์กระจายสินค้า โดยลูกค้ารายหนึ่งจะรับสินค้าได้จากรถบรรทุกเพียงคันเดียวเท่านั้น เช่น สมมุติให้มีจำนวนเมืองที่จะจัดเส้นทางการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดินทางจำนวน 5 เมือง ระยะการเดินทางจากเมืองหนึ่งไปอีกเมืองหนึ่งแสดงในตารางที่ 2.1 ซึ่งมีลำดับขั้นหรือรหัสเทียมดังนี้ [6]

1. เริ่มต้นจากเมืองใดก็ได้ (กรณีที่ไม่ได้ระบุเมืองเบื้องต้น (Home Town) ของเส้นทาง) อาจจะใช้วิธีการวงกลมรูเล็ตในการเลือกเมืองแรก
2. เลือกเมืองในลำดับถัดไปจากเมืองที่มีระยะทางใกล้ที่สุดจากเมืองที่แล้ว
3. ทำซ้ำตอนที่ 2 จนทุกเมืองถูกเลือก และเมืองสุดท้ายถูกเลือก พนักงานจะเดินทางกลับไปที่เมืองที่เลือกไว้ในข้อที่ 1

ตารางที่ 2.1 ระยะทางการเดินทางระหว่างแต่ละเมือง

i \ j	1	2	3	4	5
1	0	18	17	20	21
2	18	0	21	16	18
3	16	98	0	19	20
4	17	16	20	0	1
5	18	18	19	16	0

ตัวอย่างที่ 1 จากตารางที่ 2.1 ให้ประยุกต์ใช้รหัสเทียมหรือลำดับขั้นซึ่งมีรหัสเทียมดังนี้

1. เลือกเมืองที่ 1 เป็นเมืองเริ่มต้น(ได้จากการสุ่มด้วยวิธีวงกลมรูเล็ต)
2. เลือกเมืองในลำดับถัดไป

ลำดับที่ 2 รอบที่ 1

เมืองที่อยู่ใกล้ที่สุดจากเมืองที่ 1 คือเมืองที่ 3 เพราะมีระยะทาง 17 ซึ่งถ้าเดินทางไปเมืองที่ 2 จะมีระยะทาง 18 เดินทางไปเมืองที่ 4 และ 5 จะมีระยะทาง 20 และ 21 หน่วยระยะทางตามลำดับ ดังนั้นเส้นทางปัจจุบันเป็น 1-3 โดยมีระยะทางรวม 17 หน่วยระยะทาง

ลำดับที่ 2 รอบที่ 2

เมืองปัจจุบันคือเมืองที่ 3 เมืองที่ใกล้ที่สุดของเมืองที่ 3 คือเมืองที่ 1 แต่เรายังกลับไปไม่ได้ เพราะจะทำให้เส้นทางผ่านแค่ 2 เมือง คือเมืองที่ 1 และเมืองที่ 3 ดังนั้นเราจึงใช้ลำดับใกล้ที่สุดถัดจากเมืองที่ 1 นั่นคือ เมืองที่ 4 ซึ่งมีระยะทาง 19 หน่วย เส้นทางปัจจุบันคือ 1-3-4 ซึ่งมีระยะทางรวม 17+19= 36 หน่วยระยะทาง

ลำดับที่ 2 รอบที่ 3

เมืองปัจจุบันคือเมืองที่ 4 เมืองที่ใกล้ที่สุดของเมืองที่ 4 คือ เมืองที่ 2 ซึ่งมีระยะทาง 16 หน่วยเส้นทางปัจจุบันคือ 1-3-4-2 ซึ่งมีระยะทางรวมคือ 17+19+16 = 52 หน่วยระยะทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับที่ 2 รอบที่ 4

เมืองปัจจุบันคือเมืองที่ 2 เมืองที่ใกล้ที่สุดของเมืองที่ 2 คือ เมืองที่ 5 ซึ่งมีระยะทาง 18 หน่วยเส้นทางปัจจุบันคือ 1-3-4-2-5 ซึ่งมีระยะทางรวมคือ 17+19+16+18 = 70 หน่วยระยะทาง

และเมื่อเดินทางผ่านทุกเมืองแล้ว ก็จะกลับไปที่เมืองเริ่มต้นคือเมืองที่ 1 ซึ่งจะได้เส้นทางสุดท้ายเป็นดังนี้ 1-3-4-2-5-1 ซึ่งมีระยะทางรวมคือ 17+19+16+18+18 = 88 หน่วยระยะทาง

นั่นคือคำตอบที่ได้จากการเริ่มต้นด้วยเมืองที่ 1 หากเราเริ่มต้นที่เมืองที่ 2 จะได้คำตอบเป็น 2-4-5-1-3-2 ซึ่งมีระยะทางรวมเป็น 84 หน่วยระยะทาง ซึ่งสั้นกว่าการเริ่มต้นด้วยเมืองที่ 1 วิธีการนี้หากจะดำเนินการให้ครบเป็นฮิวริสติกที่สมบูรณ์ มักจะดำเนินการด้วยการเปลี่ยนจุดเริ่มต้นไปเรื่อย ๆ ถือว่าเป็นคำตอบของฮิวริสติก วิธีการนี้เป็นการพัฒนาฮิวริสติกอย่างง่าย ผลการเริ่มต้นจากเมืองต่าง ๆ และใช้ลำดับขั้นหรือรหัสเทียม แสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 คำตอบที่ได้จากการเริ่มต้นที่แต่ละเมืองและใช้วิธีการเลือกเมืองที่ใกล้ที่สุดเป็นลำดับถัดไป (Nearest Neighborhood (NN)) ในการสร้างคำตอบ

เมืองเริ่มต้น	เส้นทาง	ระยะทาง
1	1-3-4-2-5-1	88
2	2-4-5-1-3-2	84
3	3-1-2-4-5-3	86
4	4-2-1-3-5-4	87
5	5-4-2-1-3-5	87

จากตารางที่ 2.2 การเริ่มต้นที่เมืองที่ 2 จะมีระยะทางที่สั้นที่สุดคือ 84 หน่วยระยะทาง ซึ่งถือว่าเป็นคำตอบของฮิวริสติกอย่างง่ายนี้

2.2.1.3 การเลือกเส้นทางจากความน่าจะเป็น

การเลือกเส้นทางการเดินรถจะใช้ค่าความน่าจะเป็น ซึ่งคำนวณจากปัจจัยที่สำคัญสองอย่างคือ ค่าความเข้มข้นของฟีโรโมน $(\tau_{i,j})$ และค่าทัศนวิสัยของมด $(\eta_{i,j})$ โดยค่าทัศนวิสัยของมดสามารถคำนวณได้จากสมการที่ (2.3)

$$\eta_{i,j} = \frac{1}{d_{i,j} + \frac{d_{j,0}}{b} + 1} \quad ;\forall i, j \in \{0,...,n\} \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย d_{ij} คือระยะทางจากลูกค้า i ไปยังลูกค้า j

$d_{j,0}$ คือระยะทางจากลูกค้า j ไปยังศูนย์กระจายสินค้า

b คือค่าคงที่ของการให้ความสำคัญกับลูกค้าที่อยู่ใกล้ศูนย์กระจายสินค้า

n คือจำนวนลูกค้าทั้งหมด

ในการเลือกลำดับลูกค้าถัดไป ค่าความน่าจะเป็นที่รถบรรทุกคันหนึ่งจะเลือกไปหาลูกค้า j จากตำแหน่งของลูกค้า i สามารถคำนวณได้จากสมการที่ (2.4) [3]

$$p_{i,j} = \begin{cases} \max_{j \in S}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta} & if \;\; q \le q_0 \\ \text{(exploitation)} & \\ \dfrac{[\tau_{i,j}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{z \in S}[\tau_{i,z}]^{\alpha}[\eta_{i,z}]^{\beta}} & ; j \in S \;\; if \;\; q > q_0 \\ \text{(exploration)} & \end{cases} \tag{2.4}$$

โดย α คือพารามิเตอร์ที่กำหนดอิทธิพลของการใช้ค่าความเข้มข้นของฟีโรโมน

β คือพารามิเตอร์ที่กำหนดอิทธิพลของค่าทัศนวิสัยของมด

S คือเซตของลูกค้าที่ยังไม่ได้รับสินค้า

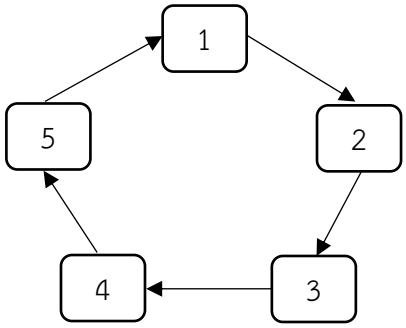
q_0 คือพารามิเตอร์ที่ใช้ควบคุมการเลือกค่าความน่าจะเป็นจากสมการ exploitation และ สมการ exploration

q คือตัวแปรสุ่มที่ใช้เลือกค่าความน่าจะเป็น ซึ่งมีค่าอยู่ระหว่าง [0,1]

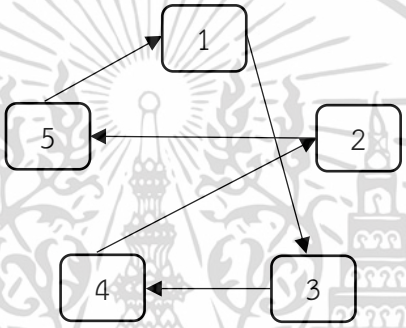
2.2.1.4 การปรับปรุงคำตอบโดยใช้วิธีการค้นหาเฉพาะที่

วิธีการค้นหาเฉพาะที่เป็นวิธีที่ใช้สำหรับปรับปรุงคุณภาพของคำตอบ ซึ่งงานวิจัยนี้ได้ใช้วิธีสลับสองตำแหน่ง (2-Opt) ในการปรับปรุงเส้นทางที่ดีที่สุดที่หาได้ของมดในแต่ละกลุ่ม โดยวิธีสลับสองตำแหน่งจะเลือกสลับสองเส้นเชื่อม (edge) ที่ไม่อยู่ติดกันบนเส้นทางเดิม เพื่อให้ได้เส้นทางใหม่ที่สั้นกว่า การสลับจะสลับทุกเส้นเชื่อมที่เป็นไปได้ของรถบรรทุกแต่ละคันจนกว่าจะไม่สามารถสลับได้อีก เพื่อหาเส้นทางที่ให้ระยะทางสั้นที่สุด เช่น หากเส้นทางการเดินทางเป็น 1-2-3-4-5-1 การใช้วิธีสลับสองตำแหน่งนี้จะเลือกคู่ของเมืองมาสองคู่ เช่น ในที่นี้เลือกคู่ 1-2 และคู่ 4-5 ในการสลับ โดยหลักการคือห้ามคู่ 1-2 และ 4-5 เดินทางต่อกันอีก จะได้เส้นทางใหม่ดังนี้ 1-3-4-2-5-1 หรือสามารถแทนเป็นรูปที่ 2.4 และ 2.5 [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

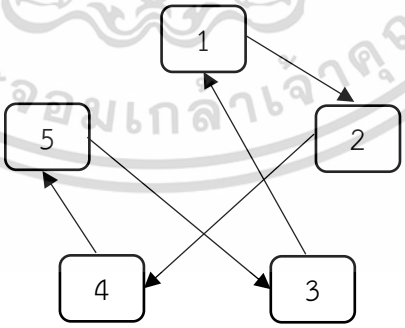


รูปที่ 2.4 เส้นทางการเดินก่อนทำวิธีสลับสองตำแหน่ง



รูปที่ 2.5 เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง

จากรูปที่ 2.4 หากคู่ที่ถูกเลือกให้ทำวิธีสลับสองตำแหน่งเป็นคู่ 2-3 และคู่ที่ 5-1 โดยหลักการมีไว้ว่าห้ามคู่ 2-3 และคู่ 5-1 เดินทางต่อกันอีก จะได้เส้นทางใหม่ดังนี้ 1-2-4-5-3-1 หรือสามารถแทนเป็นรูปภาพที่ 2.6 [6]



รูปที่ 2.6 เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง

วิธีการสลับสองตำแหน่ง นิยมใช้กับปัญหาด้านการขนส่งหรือการเดินทาง แต่อาจจะนำไปประยุกต์ใช้ได้กับปัญหาอื่น ๆ เช่น ในตัวอย่างที่ 2 จะแสดงการประยุกต์ใช้ วิธีสลับสองตำแหน่งกับปัญหาการหาขนาดการผลิตที่เหมาะสม

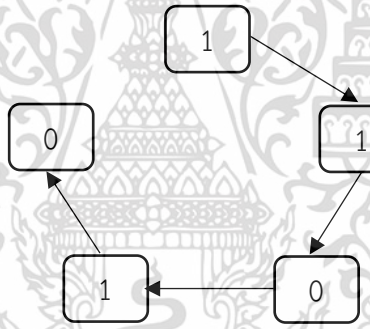
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 สมมุติว่ามีแผนการผลิตของสินค้าชนิดหนึ่งแสดงในตารางที่ 2.3

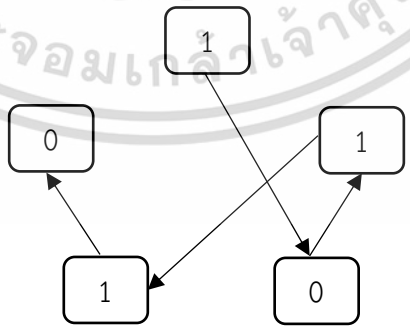
ตารางที่ 2.3 แผนการผลิตของสินค้าชนิดหนึ่งในตัวอย่างที่ 2

คาบเวลา	1	2	3	4	5
ความต้องการ	100	150	200	180	180
Y_t	1	1	0	1	0
จำนวนผลิต	100	350	0	360	0
จำนวนชิ้นในการเก็บ	0	200	0	180	0

จากตารางที่ 2.3 ผลิตสินค้าในคาบเวลาที่ 1,2 และ 4 จำนวน 100,350 และ 360 ชิ้น ตามลำดับ และมีการเก็บสินค้าในคาบเวลาที่ 2 จำนวน 200 ชิ้น ในคาบที่ 4 จำนวน 180 ชิ้น สามารถนำค่า Y_t มาจัดเป็นลำดับดังนี้ 1-1-0-1-0 และเลือกคู่ที่ต้องการสลับคือ คู่ลำดับที่ 1-2 (1-1) และ 3-4 (0-1) ซึ่งจะได้ค่าการผลิต-ไม่ผลิตเป็น 1-0-1-1-0 รายละเอียดการสลับตำแหน่งสามารถแสดงได้ดังรูปที่ 2.7 และ รูปที่ 2.8 [6]



รูปที่ 2.7 ค่า Y_t ก่อนทำวิธีสลับสองตำแหน่ง



รูปที่ 2.8 เส้นทางการเดินหลังทำวิธีสลับสองตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.7 ค่า Y_t ในลำดับต่าง ๆ คือ 1-1-0-1-0 ซึ่งไม่มีการเดินทางกลับจากตำแหน่งที่ 5 ไปตำแหน่งที่ 1 ไม่มีการผลิตวนซ้ำเหมือนการเดินทางกลับไปที่จุดเริ่มต้นในปัญหา TPS เพราะการหาขนาดการผลิตเมื่อวางแผนจบที่คาบเวลาที่ 5 หรือตำแหน่งที่ 5 ก็จะเป็นการสิ้นสุดการวางแผนการผลิตเมื่อดำเนินการด้วยวิธีสลับสองตำแหน่ง คู่ที่ 1-2 (1-1) และ 3-4 (0-1) จะเรียงลำดับ Y_t ใหม่เป็น 1-0-1-1-0 เนื่องจากต้องห้ามการเดินทางเชื่อมกันระหว่างตำแหน่ง 1-2 และ 3-4 จากนั้นได้ตารางการผลิตใหม่ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 แผนการผลิตใหม่หลังจากดำเนินการด้วยวิธีสลับสองตำแหน่ง

คาบเวลา	1	2	3	4	5
ความต้องการ	100	150	200	180	180
Y_t	1	0	1	1	0
จำนวนผลิต	250	0	200	360	0
จำนวนชิ้นในการเก็บ	150	0	0	180	0

จากตารางที่ 2.4 มีการผลิตในคาบเวลาที่ 1, 3 และ 4 ด้วยจำนวน 250, 200 และ 360 ชิ้น ตามลำดับ และเมื่อผลิตมากกว่าความต้องการ ในคาบที่ 1 โดยผลิตสินค้าจำนวน 250 ชิ้น และจะถูกขายหรือใช้ไปจำนวน 100 ชิ้น ดังนั้นเมื่อสิ้นคาบเวลาที่ 1 จะเหลือสินค้าที่ต้องเก็บไว้เพื่อใช้ในคาบต่อ ๆ ไปจำนวน 150 ชิ้น เช่นเดียวกับในคาบเวลาที่ 4 ที่ต้องเหลือสินค้า 180 ชิ้น เพื่อเก็บไว้ใช้ในคาบที่ 5 จากตารางที่ 2 จะมีจำนวนครั้งที่สั่งผลิต 3 ครั้ง คือคาบเวลาที่ 1,3 และ 4 คิดเป็นค่าใช้จ่ายในการสั่งผลิต $3\times 45,000 = 135,000$ บาท (45,000 บาท คือค่าใช้จ่ายในการสั่งซื้อหรือสั่งผลิตที่โจทย์กำหนด / ครั้ง) และจำนวนที่จัดเก็บคือ $150+180=330$ ชิ้น ซึ่งต้องเสียค่าใช้จ่ายในการจัดเก็บ $330\times 210 = 69,300$ บาท (210 บาท คือค่าใช้จ่ายในการจัดเก็บโจทย์กำหนด) รวมค่าใช้จ่าย $204,300$ บาท

2.2.1.5 การปรับค่าฟีโรโมน

การปรับค่าฟีโรโมนเป็นการเพิ่มโอกาสให้มดสามารถหาเส้นทางที่ดีกว่าเดิม โดยงานวิจัยนี้ใช้วิธีที่เหมือนกันในการปรับค่าฟีโรโมนเฉพาะที่และค่าฟีโรโมนส่วนกลาง การปรับค่าฟีโรโมนสามารถแบ่งออกเป็นสองขั้นตอนคือการลดค่าฟีโรโมน และการเพิ่มค่าฟีโรโมน การลดค่าฟีโรโมนจะกระทำกับทุกค่าของคู่อันดับ $[i,j]$ โดยฟีโรโมนเฉพาะที่และฟีโรโมนส่วนกลางทุก ๆ ค่าจะถูกลดเป็นสัดส่วนที่เท่ากัน (กำหนดโดยอัตราการระเหย ρ ซึ่งมีค่าอยู่ระหว่าง 0 ถึง 1) ซึ่งค่าฟีโรโมนใหม่สามารถคำนวณได้จากสมการที่ (2.5) [3] โดย k คือดัชนีของฟีโรโมนเฉพาะที่ $(1,...,L)$ หรือฟีโรโมนส่วนกลาง (g)

$$\tau_{i,j}^{k} = (1-\rho)\tau_{i,j}^{k} \quad ;\forall i,j \in \{0,...,n\}, \quad \forall k \in \{1,...,L;g\} \tag{2.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มค่าฟีโรโมนจะใช้คำตอบที่ดีที่สุดที่หาได้ของมดแต่ละกลุ่ม (d_k^*) ในการปรับค่าฟีโรโมนเฉพาะที่ของมดกลุ่มนั้น ๆ และใช้คำตอบที่ดีที่สุดจากมดทุกกลุ่มในการเพิ่ม ค่าฟีโรโมนส่วนกลาง โดย k แทนดัชนีของฟีโรโมนเฉพาะที่ $(1,...,L)$ หรือฟีโรโมนส่วนกลาง (g) การเพิ่มค่าจะเพิ่มเฉพาะเส้นเชื่อมที่อยู่บนเส้นทางของ d_k^* เท่านั้น โดยเพิ่มด้วยอัตราการเพิ่มของฟีโรโมน $(\Delta;\Delta>0)$ ซึ่งแสดงไว้ในสมการที่ (2.6) [3]

$$\tau^k_{i,d_k^*(i)} = \tau^k_{i,d_k^*(i)} + \Delta \quad ;\forall i \in \{0,...,n\}, \quad \forall k \in \{1,...,L;g\} \tag{2.6}$$

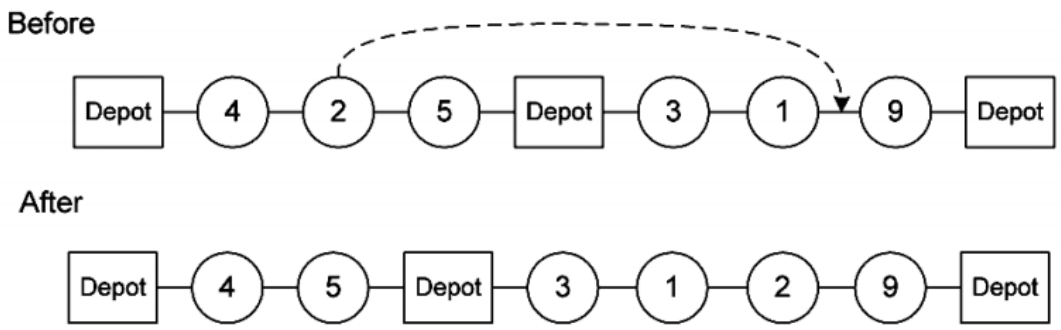
โดย $d_k^*(i)$ คือหมายเลขของลูกค้าที่ถัดไปจากลูกค้า i บนเส้นทาง d_k^*

2.3 วิธีการค้นหาเฉพาะที่ (Local Search)

วิธีการปรับปรุงคำตอบด้วยวิธีการค้นหาเฉพาะที่ (Local Search) ได้ถูกนำมาใช้ร่วมกับวิธีอาณานิคมมด เพื่อให้ได้คำตอบที่ดีขึ้น จิตินนท์และรพีพันธ์.[15] ได้ ประยุกต์ใช้วิธีอาณานิคมมดและการปรับปรุงคุณภาพคำตอบด้วยวิธีการย้ายลูกค้าระหว่างเส้นทาง (Crossover-Move) การสลับสองตำแหน่ง (2-Opt) และการย้ายหนึ่งตำแหน่ง (One-Move) เพื่อแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะ (Vehicle Routing Problem: VRP) ให้กับบริษัทเจียรนัยน้ำดื่ม ซึ่งมีรูปแบบปัญหาเป็นแบบเอ็นพี-ฮาร์ด (NP-HARD) โดยความจุของยานพาหนะมีจำนวนจำกัดและลูกค้าแต่ละรายมีความต้องการสินค้าไม่แน่นอน ซึ่งผลการทดสอบ เมื่อเปรียบเทียบกับการจัดเส้นทางของผู้ประกอบการพบว่า อัลกอริทึมที่นำเสนอให้ผลลัพธ์อยู่ในระดับที่ดี โดยสามารถลดระยะทางจากเดิม 584.25 กิโลเมตรเป็น 441.35 กิโลเมตร หรือคิดเป็น 24.46 เปอร์เซ็นต์ จากระยะทางการเดินรถเดิม

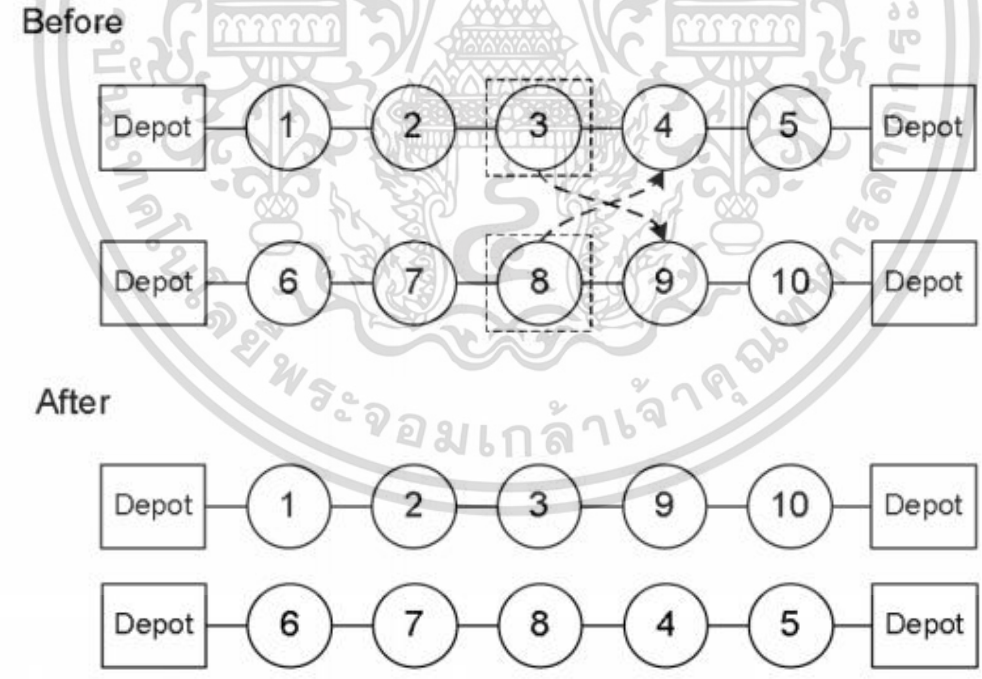
การปรับปรุงคุณภาพคำตอบด้วยวิธีการ One-Move เป็นการย้ายลูกค้าหนึ่งรายจากสับเซตหนึ่งไปยังอีกสับเซตหนึ่ง โดยไม่มีการย้ายแบบสลับหรือย้ายภายในสับเซตตัวเอง ดังแสดงในรูปที่ 2.9 และการย้ายจะต้องไม่ขัดแย้งกับเงื่อนไข ในงานวิจัยนี้ลูกค้าจะ ถูกเลือกและย้ายไปแทรกในเส้นทางอื่นที่ไม่ใช่เส้นทางเดิมที่ลูกค้ารายนั้น ในทุก ๆ ตำแหน่งเท่าที่จะเป็นไปได้แล้วบันทึกค่าระยะทางที่เกิดขึ้นไว้เพื่อเลือกตำแหน่งที่ย้ายแล้วดีที่สุด จากนั้นทำซ้ำการย้ายลูกค้าจนกว่าจะไม่สามารถลดระยะทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การปรับปรุงคำตอบด้วยวิธี One-Move

การปรับปรุงคุณภาพคำตอบด้วยวิธีการ Crossover-Move เป็นการย้ายลูกค้าระหว่างเส้นทางย่อย โดยการเลือกลูกค้ามาสองรายจากสองเส้นทาง แล้วทำการสลับลูกค้าทั้งหมดที่อยู่ถัดจากลูกค้ารายที่เลือกนั้น จากเส้นทางหนึ่งไปยังอีกเส้นทางหนึ่ง ดังนั้นส่วนต้นของเส้นทางแรกจะถูกเชื่อมด้วยส่วนท้ายของเส้นทางที่สองและในขณะเดียวกันส่วนต้นของเส้นทางที่สองจะถูกเชื่อมเข้ากับส่วนท้ายของเส้นทางแรกนั่นเองการย้ายจะย้ายทุก ๆ ตำแหน่งที่เป็นไปได้แล้วบันทึกค่าระยะทางรวมที่เกิดขึ้นไว้เพื่อเลือกตำแหน่งที่ย้ายแล้วดีที่สุด จากนั้นทำซ้ำการย้ายลูกค้าจนกว่าจะไม่สามารถย้ายลูกค้าได้ แสดงดังรูปที่ 2.10 [15]



รูปที่ 2.10 การปรับปรุงคำตอบด้วยวิธี Crossover-Move

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการศึกษางานวิจัยที่เกี่ยวข้อง ผู้วิจัยมีแนวคิดที่จะใช้มดหลายกลุ่มในการช่วยหาเส้นทางเพื่อแก้ปัญหาการจัดเส้นทางการเดินรถ โดยผู้วิจัยคิดว่าการใช้มดหลายกลุ่มจะช่วยเพิ่มโอกาสในการพบคำตอบที่ดีเนื่องจากสร้างความหลากหลายมากขึ้น และยังช่วยป้องกันไม่ให้การค้นหาคำตอบไปติดอยู่ในที่ใดที่หนึ่ง โดยผู้วิจัยจะทำการเก็บค่าฟีโรโมนแยกตามกลุ่มมด และใช้ฟีโรโมนส่วนกลางในการแลกเปลี่ยนข้อมูลระหว่างกลุ่ม จากการศึกษา [11] และได้รวมเอาวิธีการปรับปรุงคำตอบด้วยการสลับสองตำแหน่งเข้าด้วยกัน เพื่อให้สามารถหาคำตอบได้ดียิ่งขึ้น

2.4 อัลกอริทึมของไดก์สตรา (Dijkstra's Algorithm)

เนื่องอัลกอริทึมของไดก์สตราเป็นหนึ่งในวิธีที่นิยมนำมาใช้เพื่อแก้ไขปัญหาวิถีสั้นสุดจากจุดหนึ่งใด ๆ สำหรับข้อมูลที่มีลักษณะเชื่อมกันแบบกราฟ โดยค่าความยาวของเส้นเชื่อมไม่เป็นลบซึ่งขั้นตอนวิธีนี้จะหาระยะทางสั้นที่สุดจากจุดหนึ่งไปยังจุดใด ๆ ในกราฟโดยจะหาเส้นทางที่สั้นที่สุดไปทีละจุดยอดเรื่อย ๆ จนครบตามที่ต้องการ

ขั้นตอนวิธีหาวิถีสั้นที่สุดสำหรับกราฟที่มีการถ่วงน้ำหนักของเส้นเชื่อมที่ไม่เป็นลบนั้นจะแสดงรหัสเทียมดังรูปที่ 2.11 โดยอัลกอริทึมของไดก์สตราจะรักษาลำดับความสำคัญของคิว $\min Q$ ที่ใช้ในการเก็บจุดยอดที่ยังไม่ได้ประมวลผลด้วยค่าประมาณเส้นทางที่สั้นที่สุด $est(v)$ เป็นค่าคีย์ จากนั้นจะแยกจุดยอด u ที่มี $est(u)$ ต่ำสุดจาก $\min Q$ ซ้ำแล้วซ้ำเล่า และคลายทุกเส้นเชื่อมที่เกิดขึ้นจาก u ไปยังจุดยอดใด ๆ ใน $\min Q$ หลังจากแยกจุดยอดหนึ่งจุดจาก $\min Q$ และเมื่อการคลายทั้งหมดผ่านจุดยอดนั้นเสร็จสิ้นแล้ว อัลกอริทึมจะถือว่าจุดยอดนี้เป็นการประมวลผลและจะไม่แตะต้องจุดยอดนี้อีก อัลกอริทึมของไดก์สตราจะหยุดเมื่อ $\min Q$ ว่างเปล่าหรือเมื่อตรวจสอบจุดยอดทุกจุดเพียงครั้งเดียว [16]

Algorithm Dijkstra's shortest-path algorithm

```
Dijkstra(Graph G, Vertex s)
1. Initialize(G, s);
2. Priority_Queue minQ = {all vertices in V};
3. while (minQ ≠ ∅) do
4.    Vertex u = ExtractMin(minQ); // minimum est(u)
5.    for (each v ∈ minQ such that (u, v) ∈ E)
6.       Relax(u, v);
7.    end for
8. end while
```

รูปที่ 2.11 ขั้นตอนวิธีหาวิถีสั้นที่สุดด้วยอัลกอริทึมของไดก์สตรา

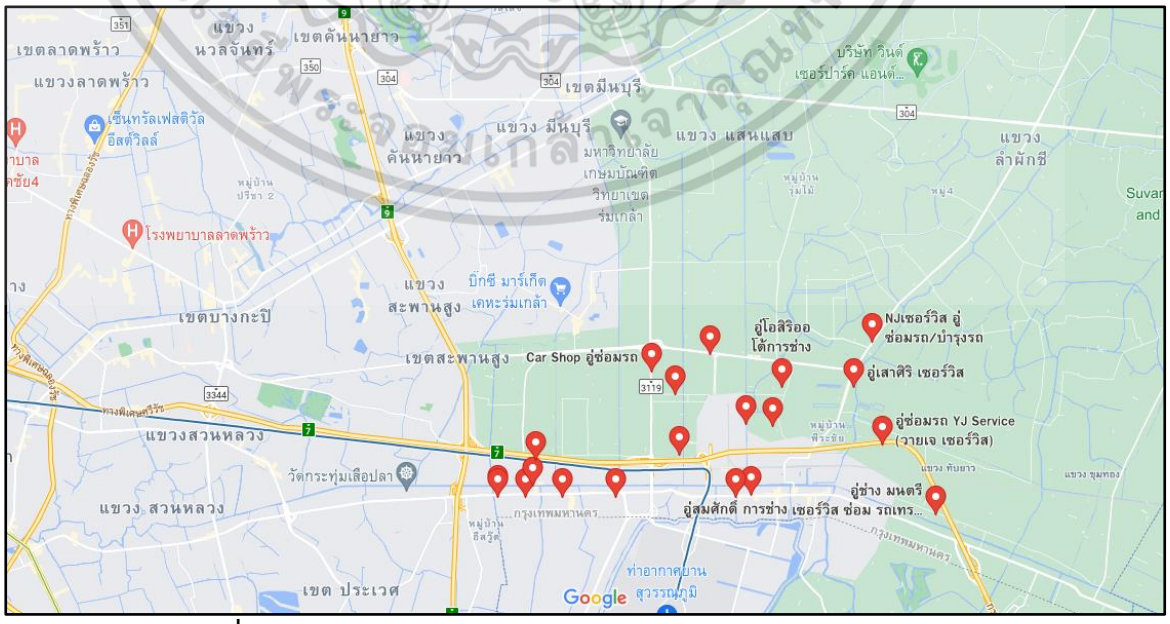
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงานวิจัย

ในบทนี้จะนำทฤษฎีบทต่างๆ มาใช้ในการพัฒนาระบบจัดการงานสินไหมในการประกันภัยรถยนต์ที่นำวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง มาประยุกต์ใช้ในการแก้ปัญหาการจัดเส้นทางการเดินรถจากที่ได้กล่าวมาในบทก่อนหน้า โดยจะอธิบายการเก็บรวบรวมข้อมูล การออกแบบระบบและกระบวนการหาคำตอบด้วยวิธีต่างๆ รวมไปถึงอธิบายการพัฒนาระบบ

3.1 ศึกษาและเก็บรวบรวมข้อมูล

การเก็บรวบรวมข้อมูลเริ่มจากกำหนดให้ kmitl เป็นสำนักงานใหญ่(จุดเริ่มต้น และจุดสุดท้าย) โดยการสุ่มจากสถานบริการซ่อมรถยนต์ที่มีอยู่ใน จังหวัดกรุงเทพมหานคร จำนวน 20 อู่ ถัดมาจะทำการบันทึกจุดที่อยู่ของอู่ด้วยค่า latitude,longitude แล้วกำหนดตำแหน่งลงเว็บ Google Map พร้อมทั้งสร้าง Distance Matrix ดังตารางที่ 3.2 เพื่อเก็บค่าระยะทางจากจุดข้อมูลทั้งหมด เนื่องจากการเดินทางอาจจะมีความแปรปรวนในการจราจร จึงแบ่งช่วงเวลาในการเก็บข้อมูลเป็น 3 ช่วงเวลา (เช้า,กลางวัน,เย็น) โดยคาดคะเนจากการจราจร ได้แก่ 8.00-11.00 น. ,11.00-14.00 น. ,14.00-17.00 น. และกำหนดชื่ออู่ซ่อมรถตามเขตที่อยู่ของอู่ตามด้วย ลำดับที่ของอู่(กำหนดเอง) เช่น อู่สมศักดิ์การช่าง เขตลาดกระบัง ลำดับที่ 1 (11-1) โดยอ้างอิงข้อมูลเขตจาก [กรุงเทพมหานคร – วิกิพีเดีย] ดังตารางที่ 3.1



รูปที่ 3.1 ตัวอย่างตำแหน่งอู่ซ่อมรถยนต์ในจังหวัดกรุงเทพมหานคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 รายชื่อเขตทั้งหมดในกรุงเทพมหานคร (เรียงตามรหัสเขตการปกครองที่ใช้ในราชการ)

1. เขตพระนคร	11. เขตลาดกระบัง	21. เขตบางขุนเทียน	31. เขตบางคอแหลม	41. เขตหลักสี่
2. เขตดุสิต	12. เขตยานนาวา	22. เขตภาษีเจริญ	32. เขตประเวศ	42. เขตสายไหม
3. เขตหนองจอก	13. เขตสัมพันธวงศ์	23. เขตหนองแขม	33. เขตคลองเตย	43. เขตคันนายาว
4. เขตบางรัก	14. เขตพญาไท	24. เขตราษฎร์บูรณะ	34. เขตสวนหลวง	44. เขตสะพานสูง
5. เขตบางเขน	15. เขตธนบุรี	25. เขตบางพลัด	35. เขตจอมทอง	45. เขตวังทองหลาง
6. เขตบางกะปิ	16. เขตบางกอกใหญ่	26. เขตดินแดง	36. เขตดอนเมือง	46. เขตคลองสามวา
7. เขตปทุมวัน	17. เขตห้วยขวาง	27. เขตบึงกุ่ม	37. เขตราชเทวี	47. เขตบางนา
8. เขตป้อมปราบศัตรูพ่าย	18. เขตคลองสาน	28. เขตสาทร	38. เขตลาดพร้าว	48. เขตทวีวัฒนา
9. เขตพระโขนง	19. เขตตลิ่งชัน	29. เขตบางซื่อ	39. เขตวัฒนา	49. เขตทุ่งครุ
10. เขตมีนบุรี	20. เขตบางกอกน้อย	30. เขตจตุจักร	40. เขตบางแค	50. เขตบางบอน

ตารางที่ 3.2 ตัวอย่าง Distance Matrix (ระยะทางเป็น กิโลเมตร)

	11-7	**10-1**	**32-1**	**32-2**	**11-8**	**10-2**	**45-1**	**34-2**	**34-3**	**46-1**
11-7	0	13.1	17.3	23.8	8.10	9.50	21.0	24.3	22.4	17.7
10-1	11.2	0	24.0	32.8	17.0	8.90	25.3	31.2	29.0	16.5
32-1	19.5	23.9	0	8.00	13.8	16.4	12.3	8.60	5.70	22.2
32-2	29.4	33.8	9.0	0	19.5	26.2	22.0	11.1	5.10	32.1
11-8	11.4	17.3	7.7	13.7	0	15.0	26.4	20.6	12.7	24.4
10-2	7.90	7.70	19.0	27.8	13.7	0	17.2	26.2	24.0	13.5
45-1	24.6	27.7	16.9	20.9	24.9	22.3	0	8.90	15.9	20.0
34-2	25.0	33.0	9.60	11.4	19.3	21.9	8.80	0	8.70	25.2
34-3	26.2	30.7	6.40	4.90	20.5	23.2	12.3	8.60	0	29.0
46-1	17.0	14.9	22.7	31.6	24.5	9.40	17.8	23.8	27.8	0

3.2 ออกแบบระบบและกระบวนการค้นหาคำตอบด้วยวิธีต่าง ๆ

3.2.1 ออกแบบระบบ

ใช้แนวคิดพื้นฐาน “Input-Process-Output” ที่จะตอบโจทย์วัตถุประสงค์ (Objectives) ที่เราต้องการโดยเริ่มคิดที่วัตถุประสงค์ (Objectives) เป็นสำคัญ ผลลัพธ์ (Output) ที่เราต้องการใช้มีอะไรเป็นตัวชี้วัด (Indicator) มีกระบวนการ (Process) อะไรบ้าง เพื่อให้ได้ผลลัพธ์ที่เราต้องการ สุดท้ายต้องมองย้อนกลับไปว่าเรามีปัจจัยนำเข้า (Input) อะไรบ้าง เพื่อให้กระบวนการสามารถดำเนินงานได้อย่างสมบูรณ์ ซึ่งกระบวนการดังกล่าวจะแสดงดังตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 การออกแบบระบบ

Input	ข้อมูล ชื่อ และเขตพื้นที่อู่ซ่อมรถยนต์
Process	ใช้อัลกอริทึมเพื่อคำนวณหาระยะทางรวม และแนะนำเส้นทางในการเดินทางโดยพิจารณาจากอัลกอริทึมดังนี้ - ไดก์สตรา สำหรับหาวิถีสั้นที่สุดสำหรับกราฟที่มีการถ่วงน้ำหนักของเส้นเชื่อมที่ไม่เป็นลบ โดยหาระยะทางสั้นที่สุดจากจุดยอดหนึ่งไปยังจุดใด ๆ ในกราฟโดยหาเส้นทางที่สั้นที่สุดไปทีละจุดจนครบตามที่ต้องการ - ระบบอาณานิคมมด คือการใช้มดในการหาเส้นทางที่สั้นที่สุดกระบวนการหาคำตอบจะอาศัยการทำซ้ำโดยเส้นทางที่ดีจากรอบการทำซ้ำก่อนหน้าจะถูกเพิ่มฟีโรโมนให้มีความเข้มมากขึ้นเพื่อเป็นข้อมูลช่วยให้มดตัวอื่น ๆ หาเส้นทางได้ดีขึ้นในรอบ ถัดไป - ระบบอาณานิคมมดที่มีการปรับปรุงด้วยวิธีสลับสองตำแหน่ง คือวิธีเดียวกันกับระบบอาณานิคมมดแต่จะมีการเพิ่มการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งเพื่อช่วยในการค้นหาคำตอบที่ดีขึ้นจากวิธีดังกล่าว
Output	- ระยะทางรวมในการเดินทาง - แนะนำเส้นทางในการเดินทางไปยังอู่ต่างๆ - ตัวอย่างรูปแผนที่ของเส้นทางที่ต้องเดินทาง - เมทริกซ์ข้อมูลระยะทางจากอู่หนึ่งไปยังอู่หนึ่ง
Objective	ระบบจัดการงานสินไหมในกรณีความเสียหายหนักด้วยวิธีที่ให้คำตอบที่ดีที่สุด

3.2.2 กระบวนการค้นหาคำตอบด้วยวิธีต่าง ๆ

กระบวนการค้นหาคำตอบด้วยวิธีต่าง ๆ แบ่งออกเป็น 3 ส่วนหลัก คือ ขั้นตอนวิธีระบบอาณานิคมมด ขั้นตอนวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งและขั้นตอนวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา โดยมีรายละเอียดขั้นตอนวิธีดังนี้

3.2.2.1 ขั้นตอนวิธีระบบอาณานิคมมด มีขั้นตอนดังนี้

ขั้นตอนที่1 การกำหนดค่าพารามิเตอร์ต่าง ๆ

ขั้นตอนที่2 การกำหนดค่าฟีโรโมน

ขั้นตอนที่3 การเลือกเส้นทางจากความน่าจะเป็น

ขั้นตอนที่4 การปรับค่าฟีโรโมน

เนื่องจากขั้นตอนวิธีระบบอาณานิคมมด มีวิธีทำเหมือนกับหัวข้อที่ 3.2.2 โดยในหัวข้อนี้จะไม่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งเหมือนกับหัวข้อที่ 3.2.2 แต่วิธีอื่น ๆ มีกระบวนการเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 ขั้นตอนวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง (2-opt) มีขั้นตอนดังนี้

ขั้นตอนที่1 การกำหนดค่าพารามิเตอร์ต่างๆ ได้แก่

1. จำนวนรอบที่ต้องการวนซ้ำ (Iteration)
2. จำนวนประชากรมด (Ant Population)
3. ค่าฟีโรโมนเริ่มต้น(Pheromone: τ) เริ่มต้นที่ 1 เท่ากันหมด ยกเว้น การเชื่อมเมือง

   เดียวกัน เท่ากับ 0 เนื่องจากไม่สามารถดำเนินการเชื่อมได้
4. อัตราการระเหยของค่าฟีโรโมน (Evaporation Rate: ρ)
5. ค่าพารามิเตอร์ (α) และค่าพารามิเตอร์ (β)

ขั้นตอนที่2 การกำหนดค่าฟีโรโมน

การกำหนดค่าฟีโรโมน เป็นสิ่งที่สำคัญสำหรับใช้เป็นข้อมูลในการหาคำตอบ โดยค่าฟีโรโมนที่มดใช้หาคำตอบ สามารถคำนวณได้จากสมการที่ (3.1) ค่าฟีโรโมนจะถูกเก็บไว้ในเมทริกซ์ขนาด $(n)\times(n)$ โดยที่ n คือจำนวนอู่ซ่อมรถทั้งหมด เรียกเมทริกซ์นี้ว่า ฟีโรโมนเมทริกซ์ ซึ่งสามารถเขียนแทนด้วย $[\tau_{ij}]$ โดยที่ $i, j \in \{1,...,n\}$ เมื่อ 1 คือดัชนีของสำนักงานใหญ่ (จุดเริ่มต้น และจุดสุดท้าย) ในช่วงเริ่มต้นค่าฟีโรโมนแต่ละค่าจะถูกกำหนดให้มีค่าเท่ากันทั้งหมด นั้นคือกำหนดให้สมาชิกทุกตัวใน $\tau_{n\times n}$ มีค่าเท่ากับ 1 ยกเว้นการเชื่อมเมืองเดียวกัน (แนวทแยง) เท่ากับ 0 เนื่องจากไม่สามารถดำเนินการเชื่อมได้

$$\tau_{n\times n} = \begin{bmatrix} \tau_{1,1} & \cdots & \tau_{1,j} \\ \vdots & \ddots & \vdots \\ \tau_{i,1} & \cdots & \tau_{i,j} \end{bmatrix} \quad ;\forall i, j \in \{1,...,n\} \tag{3.1}$$

โดย τ_{ij} คือค่าฟีโรโมนระหว่างลูกค้า i กับลูกค้า j ที่กำหนดขึ้น

ขั้นตอนที่3 การเลือกเส้นทางจากความน่าจะเป็น

การเลือกเส้นทางการเดินรถจะใช้ค่าความน่าจะเป็น ซึ่งคำนวณจากปัจจัยที่สำคัญสองอย่างคือ ค่าความเข้มข้นของฟีโรโมน $(\tau_{i,j})$ และค่าทัศนวิสัยของมด $(\eta_{i,j})$ โดยค่าทัศนวิสัยของมดสามารถคำนวณได้จากสมการที่ (3.2)

$$\eta_{i,j} = \frac{1}{d_{i,j}} \quad ;\forall i, j \in \{1,...,n\} \tag{3.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย $d_{i,j}$ คือระยะทางจากลูกค้า i ไปยังลูกค้า j

n คือจำนวนอู่ซ่อมรถทั้งหมด

ในการเลือกลำดับลูกค้าถัดไปค่าความน่าจะเป็นที่รถยนต์คันหนึ่งจะเลือกไปหาอู่ซ่อมรถจากตำแหน่งของอู่ซ่อมรถ i สามารถคำนวณได้จากสมการที่ (3.3) และ (3.4)

$$p_{i,j} = \frac{[\tau_{i,j}]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{j=1}^{J}[\tau_{i,j}]^{\alpha}[\eta_{i,j}]^{\beta}} \quad ; j \in J \tag{3.3}$$

$$(exploration)$$

$$cumpop_{ij} = \sum_{k=1}^{J} p_{i,k} \quad ;if \;\; q \le cumpop_{ij} \tag{3.4}$$

โดย p_{ij} คือค่าความน่าจะเป็นของการเลือกอู่ซ่อมรถ j เมื่อมดอยู่ที่อู่ซ่อมรถ i

α คือพารามิเตอร์ที่กำหนดอิทธิพลของ การใช้ค่าความเข้มข้นของฟีโรโมน

β คือพารามิเตอร์ที่กำหนดอิทธิพลของ ค่าทัศนวิสัยของมด

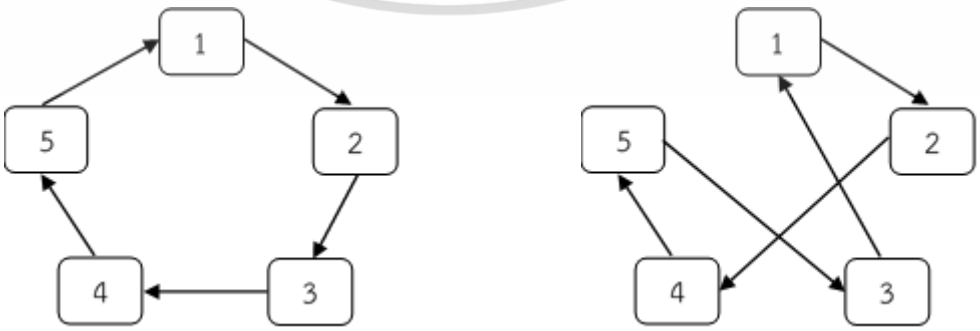
J คือเซตของอู่ซ่อมรถที่ยังไม่ถูกเลือกให้เข้าไปอยู่ในเส้นทางใด ๆ

q คือตัวแปรสุ่มที่ใช้เลือกค่าความน่าจะเป็น ซึ่งมีค่าอยู่ระหว่าง $[0,1)$

$cumpop_{ij}$ คือความน่าจะเป็นสะสมของการเลือกอู่ซ่อมรถ j เมื่อมดอยู่ที่อู่ซ่อมรถ i

ขั้นตอนที่4 การปรับปรุงคำตอบ Local search ด้วยวิธีการสลับสองตำแหน่ง

วิธีการค้นหาเฉพาะที่เป็นวิธีที่ใช้สำหรับปรับปรุง คุณภาพของคำตอบ ซึ่งงานวิจัยนี้ได้ใช้วิธีสลับสองตำแหน่งในการปรับปรุงเส้นทางที่ดีที่สุดที่หาได้ของมดในแต่ละกลุ่ม โดยวิธีสลับสองตำแหน่ง จะเลือกสลับสองเส้นเชื่อม (edge) ที่ไม่อยู่ติดกันบนเส้นทางเดิม เพื่อให้ได้เส้นทางใหม่ที่สั้นกว่า การสลับจะสลับทุกเส้นเชื่อมที่เป็นไปได้ของอู่ซ่อมรถ จนกว่าจะไม่สามารถสลับได้อีก เพื่อหาเส้นทางที่ให้ระยะทางสั้นที่สุด ดังตัวอย่างรูปที่ 3.2



ก่อนสลับ **หลังสลับ**

รูปที่ 3.2 ตัวอย่างวิธีสลับสองตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่5 การปรับค่าฟีโรโมน

การปรับค่าฟีโรโมนเป็นการเพิ่มโอกาส ให้มดสามารถหาเส้นทางที่ดีกว่าเดิม ซึ่งงานวิจัยนี้จะทำการปรับค่าฟีโรโมนได้เมื่อ มดได้รับการปรับปรุงคำตอบที่ดีที่สุดแล้วเท่านั้นจึงจะสามารถทำการปรับฟีโรโมนบนเส้นทางนั้น ๆ ได้ โดยค่าฟีโรโมนจะถูกปรับตามสมการที่ (3.5) ถ้ามดที่สร้างเส้นทางที่ดีที่สุดไม่ได้เดินทางผ่านจุดเชื่อมนั้น และค่าฟีโรโมนจะถูกปรับตามสมการที่ (3.6) ถ้ามดตัวนั้นเดินทางผ่านจุดเชื่อมคำตอบที่ดีที่สุด

$$\tau_{i,j}{}^{n} = (1-\rho)(\tau_{i,j}{}^{n-1}) \tag{3.5}$$

$$\tau_{i,j}{}^{n} = (1-\rho)(\tau_{i,j}{}^{n-1}) + \Delta \tag{3.6}$$

โดย Δ คือส่วนกลับของระยะทางรวมของเส้นทางที่สั้นที่สุดในรอบนั้น ๆ

ρ คืออัตราการระเหย ซึ่งมีค่าอยู่ระหว่าง $(0,1)$

ซึ่งอัลกอริทึมในการค้นหาเส้นทางที่เหมาะสมของวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งแสดงขั้นตอนดังนี้

1. ดึงข้อมูลค่าระยะทาง (เส้นเชื่อม: Edge) ที่ต้องการเก็บไว้ในรูปแบบเมทริกซ์ ขนาด $(n \times n)$

$$X_{n\times n} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,j} \\ \vdots & \ddots & \vdots \\ d_{i,1} & \cdots & d_{i,j} \end{bmatrix}$$

2. กำหนดค่าเริ่มต้นของพารามิเตอร์ $\rho, \alpha, \beta, Ant, maximum_iteration$

3. สร้างเมทริกซ์ค่าทัศนวิสัยของมด $\eta_{n\times n} = \begin{bmatrix} \eta_{1,1} & \cdots & \eta_{1,j} \\ \vdots & \ddots & \vdots \\ \eta_{i,1} & \cdots & \eta_{i,j} \end{bmatrix}$ โดยที่ $\eta_{i,j} = \frac{1}{d_{i,j}}$

4. สร้างเมทริกซ์ค่าความเข้มข้นของฟีโรโมน $\tau_{n\times n} = \begin{bmatrix} \tau_{1,1} & \cdots & \tau_{1,j} \\ \vdots & \ddots & \vdots \\ \tau_{i,1} & \cdots & \tau_{i,j} \end{bmatrix}$

5. for ($iteration$ = 1 to $maximum_iteration$)

6. ค่าความน่าสนใจ (Attractiveness) =

$$\begin{bmatrix} (\tau_{1,1})^{\alpha} \times (\eta_{1,1})^{\alpha} & \cdots & (\tau_{1,j})^{\alpha} \times (\eta_{1,j})^{\alpha} \\ \vdots & \ddots & \vdots \\ (\tau_{i,1})^{\alpha} \times (\eta_{i,1})^{\alpha} & \cdots & (\tau_{i,j})^{\alpha} \times (\eta_{i,j})^{\alpha} \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. $Path_{Ant\times(n+1)}$

อาเรย์ (Array) สำหรับเก็บเส้นทางของมดแต่ละตัว โดยจะเริ่มใหม่ทุกครั้งเมื่อเริ่มรอบถัดไป

8. for (i = 1 to Ant)

9. $Traveled_{1\times(n+1)}$

รายการ (list) สำหรับเก็บจุดถัดไปที่ถูกเลือก โดยจะเริ่มใหม่ทุกครั้งเมื่อเริ่มรอบถัดไป

10. for (j = 1 to n -1)

11. $$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{j=1}^{J}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} \quad ; j \in J$$

12. $$cumpop_{ij} = \sum_{k=1}^{J} p_{i,k}$$

13. if $(q \le cumpop_{ij})$

เป็นค่าที่สุ่มเลือกมีค่าน้อยกว่าหรือเท่ากับความน่าจะเป็นสะสมของจุดนั้นในลำดับแรก

14. $Traveled_{1,(j+1)}$ = จุดที่ถูกเลือก

15. ปรับปรุงเส้นทางด้วยวิธีสลับสองตำแหน่ง

16. if ($Distane\ of\ Path_{new}^{2-opt} \le Distane\ of\ Path_{old}$)

17. $Path_{i} = Path_{new}^{2-opt}$ //บันทึกทับเส้นทางเดิมเมื่อเจอเส้นทางที่ดีกว่า

18. ปรับค่าฟีโรโมน $\tau_{i,j}^{new} = (1-\rho)\tau_{i,j}^{old} + \Delta$ เฉพาะจุดในเส้นทางที่น้อยที่สุดที่มดหาเจอ

19. เลือกเส้นทางจากผลรวมค่าระยะทางที่น้อยที่สุดใน $Path_{Ant\times(n+1)}$ เมื่อจบการทำซ้ำ

3.2.2.3 ขั้นตอนวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา

(Decision criteria base on Dijkstra's algorithm)

อัลกอริทึมสำหรับหาเส้นทางที่เหมาะสม ที่เขียนอยู่บนพื้นฐานของ Dijkstra's Algorithm แสดงขั้นตอนแบบย่อดังนี้

1. $X_{n\times n} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n} \\ \vdots & \ddots & \vdots \\ d_{n,1} & \cdots & d_{n,n} \end{bmatrix}$ เมทริกซ์เก็บค่าระยะทาง (เส้นเชื่อม: Edge) ขนาด $(n \times n)$

2. $y_{1\times k} = [D_1, D_2, ..., D_n]$ เป็น Array สำหรับเก็บเส้นทางที่สั้นที่สุดจากจุดหนึ่ง (Node) ไปยังจุดใดๆ

3. $c = 1$ คือตัวแปรสำหรับกำหนดจุดเริ่มต้น

4. for (i = 1 to b) แถว

5. for (j = 1 to b) หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. if ($X_{c,j}$ น้อยกว่า $X_{c,j+1}$ และ $X_{c,j} \neq 0$) โดย $X_{c,j} = 0$ เมื่ออยู่อู่ปัจจุบัน และเป็นอู่ที่เดินทางไปแล้ว
7. D_i = $X_{c,j+1}$ เก็บค่าระยะทางสั้นที่สุด
8. $c = j$ ตัวเลขจุด(อู่)ถัดไปที่มีระยะทางสั้นที่สุด
9. End if
10. Next j
11. for (k = 1 to n) ให้หลักในเมทริกซ์ของจุดที่เดินทางไปแล้วเป็น 0 ทั้งหมด
12. $X_{k,c}$ = 0
13. Next k
14. Next i
15. D_n = ระยะทางจากจุดสุดท้าย กลับไปยังจุดเริ่ม
16. $D_S = \sum_{k=1}^{n} y_{1,k}$ รวมระยะทางทั้งหมดของเส้นทางที่ได้

โดยกำหนดให้ $x_{c,j}$ คือ ค่าที่น้อยที่สุดและไม่เป็นศูนย์ของสมาชิกในแต่ละแถวของเมทริกซ์ $X_{n\times n}$

D_S คือ ระยะทางรวม

$y_{1\not\in k}$ คือ เมทริกซ์แถวสำหรับเก็บค่าระยะทางน้อยที่สุดของจุดถัดไปโดยที่ไม่ซ้ำจุดเดิม

b คือ จำนวนงานสินไหม

$X_{n\times n}$ คือ เมทริกซ์สำหรับเก็บค่าระยะทางจากจุดหนึ่งไปยังจุดหนึ่ง โดยที่ $n = b+1$ นั่นคือจำนวนมิติเกิดจากการนำจำนวนงานสินไหมรวมกับจุดเริ่มต้น

3.3 พัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนัก

ทำการพัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนักโดยใช้โปรแกรม PyCharm 2021.1.3(Community Edition) ในการเขียนภาษา Python (Python3.9) โดยมีขั้นตอนของระบบการจัดการสินไหมกรณีความเสียหายหนักดังนี้

1. รับงานเข้าสู่ระบบ ด้วยชื่อสถานบริการซ่อมรถยนต์และข้อมูลเบื้องต้นอื่น ๆ

2. อัพเดทงานและเพิ่มเข้าระบบจ่ายงานเพื่อดึงข้อมูล ชื่อสถานบริการซ่อมรถยนต์ latitude, longitude และ ข้อมูลระยะทางของการเดินทางไปยังสถานบริการซ่อมรถยนต์ต่าง ๆ

3. กำหนดค่าพารามิเตอร์ต่าง ๆ ที่จำเป็น ได้แก่ Iteration, Ant Population, Evaporation Rate, α , β

4. เลือกเข้าสู่ขั้นตอนวิธีระบบอาณานิคมมด หรือขั้นตอนวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง หรือขั้นตอนวิธีของไดก์สตรา

5. แสดงผลลัพธ์ระยะทางและเส้นทางที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการวิจัยและการอภิปรายผล

ในบทนี้จะแบ่งการแสดงผลออกเป็น 2 ส่วน ส่วนที่หนึ่งคือผลลัพธ์การเปรียบเทียบซึ่งจะเป็นการอธิบายผลลัพธ์ที่ได้จากวิธีที่นำมาทดสอบและอธิบายผลที่ได้ ถัดมาจะเป็นผลการพัฒนาระบบโดยหน้าโปรแกรมจะถูกพัฒนาโดยภาษาไพทอน

4.1 ผลลัพธ์การเปรียบเทียบ

ผู้วิจัยทดลองเปรียบเทียบ 3 วิธีในการจัดการปัญหาเส้นทางการเดินรถ ซึ่งกำหนดให้มีการค้นหาคำตอบจำนวน 100 คำตอบ ได้แก่ วิธีระบบอาณานิคมมด, วิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง และวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา โดยกำหนดค่าพารามิเตอร์ที่เกี่ยวข้องกับวิธี วิธีระบบอาณานิคมมด และวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง เป็น 3 ระดับ คือ ระดับต่ำ (Low Setting) ระดับกลาง (Medium Setting) และระดับสูง (High Setting) ดังตารางที่ 4.1 ที่ประกอบด้วย ค่าอัตราการระเหยของค่าฟีโรโมน (Evaporation Rate) โดยผู้วิจัยกำหนดให้มีอัตราการระเหยของฟีโรโมนต่อครั้งสูงสุดที่ 9% นั่นหมายความว่า ถ้ากำหนดค่าเป็น 100% แสดงว่าการระเหยต่อครั้งคือ 100% ซึ่งจะทำให้ได้คำตอบที่ไม่หลากหลายและไม่เข้าใกล้ค่าที่ดีที่สุด จึงแบ่งช่วงอัตราการระเหยของฟีโรโมนที่ 3 ระดับคือ 0.01 0.05 และ 0.09 ตามลำดับ ถัดมาในส่วน ค่าพารามิเตอร์ (α) และค่าพารามิเตอร์ (β) ที่จะเป็นค่าคงที่ค่าหนึ่งที่ระบุถึงการให้น้ำหนักกับค่าฟีโรโมนกับค่าระยะทางระหว่างการเชื่อมของเมืองสองเมือง โดยทั่วไปแอลฟาและบีตาจะถูกตั้งค่าเป็นค่าจำนวนเต็มที่มีค่าระหว่าง 1-5 จึงแบ่งช่วงในการทดสอบที่ 3 ระดับคือ 1, 3 และ 5 ตามลำดับ ซึ่งจำนวนมด (Ant Population) กำหนดให้อยู่ในช่วง 20-100 ตัว โดยแบ่งเป็นช่วงนำไปทดสอบดังนี้ 20, 40, 60, 80 และ 100 ตัว เพื่อดูแนวโน้มของการได้ค่าที่ดีที่สุด และจำนวนรอบในการวนซ้ำ (Iteration) 100 รอบ ส่วนวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา คือการพิจารณาเส้นทางจากค่าระยะทางถัดไปที่น้อยที่สุดโดยไม่มีการพิจารณาย้อนกลับจนกว่าจะพิจารณาครบทุกจุด ใช้หน่วยประมวลผล Intel®Core(TM) i7-6500U @2.5GHz RAM 8.00 GB แล้วบันทึกค่าประมาณการของระยะทางรวมที่น้อยที่สุด โดยตารางที่ 4.2 จะแสดงรายละเอียดของปัญหาที่สุ่มมาจำนวน 10 อู่ ซึ่งประกอบด้วย เขตที่อยู่ของอู่ในจังหวัดกรุงเทพมหานคร (01-50) และลำดับที่ของอู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าพารามิเตอร์ต่างๆ

Level / Parameter	Evaporation rate	Alpha (α)	Beta (β)
Low Setting	0.01	1	1
Medium Setting	0.05	3	3
High Setting	0.09	5	5

ตารางที่ 4.2 ข้อมูลระยะทางของปัญหาที่สุ่มมาทดสอบในระบบ

	11-01	07-01	38-01	11-02	11-03	11-04	34-01	11-05	11-06	11-07
11-01	0	35.5	36.9	2.7	6.6	4.6	23.5	9.9	3.1	11.7
07-01	34.5	0	18.3	35.6	38.6	35.1	17.3	34.6	32.2	36.4
38-01	31.4	24	0	32.5	35.4	31.9	19.2	31.4	29.1	33.2
11-02	6.8	39.6	40.9	0	3.9	6.3	27.5	12.5	9.1	14.4
11-03	6.5	39.2	40.5	3.3	0	5.9	27.1	12.2	7.6	14
11-04	4.5	35.7	37	6.3	5.9	0	23.6	8	7.7	9.8
34-01	17.8	18.6	19.7	18.9	21.8	18.3	0	17.8	15.4	19.6
11-05	11.1	34.7	24.7	11.4	13.7	8.2	22.6	0	8	2
11-06	7.9	33	34.3	9	12.9	11.3	20.9	7.4	0	9.2
11-07	12.9	38.6	25.1	15.3	15.5	10	26.5	8.5	11.9	0

จากการทดสอบการจัดเส้นทางการเดินรถด้วยวิธีที่กำหนดพบว่า เมื่อใช้วิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตราได้ระยะทางรวมที่ 127 กิโลเมตร ดังตารางที่ 4.3 นั่นคือในการค้นหาคำตอบทั้งหมด 100 คำตอบยังได้คำตอบ และเส้นทางแบบเดิมแต่ต่างจากวิธีระบบอาณานิคมมด และวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง เนื่องจากการค้นหาคำตอบแต่ละรอบนั้นอาจจะเจอคำตอบที่ดีซึ่งขึ้นอยู่กับการตั้งค่าพารามิเตอร์ต่าง ๆ ดังนั้นผู้วิจัยจึงนำวิธีที่เหลือมาทดสอบหาค่าประสิทธิภาพ (Efficiency) เพิ่มเติมโดยหาว่าคำตอบที่ได้มีโอกาสเป็นคำตอบที่ดีกว่าวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตราที่ระยะทางรวม 127 กิโลเมตรอยู่กี่เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลลัพธ์ด้วยวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา

รอบที่ / ลำดับที่	1	2	3	4	5	6	7	8	9	10	11	ระยะทางรวม
1	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
2	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
3	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
4	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
5	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
6	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
7	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
8	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
9	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
10	11-01	11-02	11-03	11-04	11-06	11-05	11-07	38-01	34-01	07-01	11-01	127
⋮	"										"	⋮
100	11-01	11-02	11-03	11-04	11-05	11-07	38-01	34-01	07-01	11-06	11-01	127

ซึ่งจากการแบ่งการทดสอบตามตารางค่าพารามิเตอร์เป็น 3 ระดับในวิธีระบบอาณานิคมมด และวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งเปรียบเทียบกันนั้นจะได้ผลลัพธ์ดังตารางที่ 4.4, 4.5 และ 4.6 ตามลำดับ โดยตารางที่ 4.4 จะแสดงผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับต่ำ (Low Setting) แสดงให้เห็นว่าที่ค่าประชากรมด (Ant Population) เริ่มที่ 20 และค่อยๆ เพิ่มขึ้นตามการแบ่งการทดสอบนั้น ที่ช่วงแรกที่วิธีระบบอาณานิคมมดจะได้ค่าวัดประสิทธิภาพ (Efficiency) อยู่ 61% เพิ่มขึ้นเป็น 90%, 98% มีการปรับเพิ่มขึ้นเรื่อย ๆ จนกระทั่งที่ 100% ซึ่งจะต่างจากวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง เนื่องจากที่ช่วงแรกค่าวัดประสิทธิภาพ (Efficiency) อยู่ 95% เพิ่มขึ้นเป็น 98%, 99% จนกระทั่ง 100% ส่วนนี้แสดงให้เห็นว่าหากมีการนำวิธีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งมาร่วมในการค้นหาคำตอบด้วยนั้น จะทำให้สามารถหาผลลัพธ์ที่ดีได้เร็วขึ้น โดยจากทั้งสามตารางดังกล่าวแสดงให้เห็นว่าค่าพารามิเตอร์ในกลุ่มที่ 2 มีแนวโน้มของโอกาสที่จะได้คำตอบที่ดี ผู้วิจัยจึงได้เลือกค่าพารามิเตอร์กลุ่มดังกล่าวที่ค่าจำนวนมด (Ant Population) เท่ากับ 80 ตัว ด้วยวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่งเป็นผลลัพธ์ที่ดีที่สุด และเมื่อเปรียบเทียบผลลัพธ์ค่าเฉลี่ยของระยะทางรวมทั้งสามวิธีจะเห็นว่าทั้งวิธีระบบอาณานิคมมด และวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง เป็นวิธีที่ให้ผลลัพธ์ดีกว่าวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตราทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่ 1

ตารางที่ 4.4 ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับต่ำ (Low threshold setting)

Parameters	Iteration = 100, Evaporation rate = 0.01, Alpha = 1, Beta = 1									
Ant Population	20		40		60		80		100	
Algorithm	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt
Average (km)	124.68	119.132	119.396	115.532	117.442	114.2	115.79	113.805	114.674	112.943
Efficiency (%)	61	95	90	100	98	100	99	100	100	100

กลุ่มที่ 2

ตารางที่ 4.5 ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับกลาง (Medium threshold setting)

Parameters	Iteration = 100, Evaporation rate = 0.05, Alpha = 3, Beta = 3									
Ant Population	20		40		60		80		100	
Algorithm	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt
Average (km)	114.407	112.988	112.562	112.4	112.269	112.247	112.196	112.181	112.197	112.157
Efficiency (%)	100	100	100	100	100	100	100	100	100	100

กลุ่มที่ 3

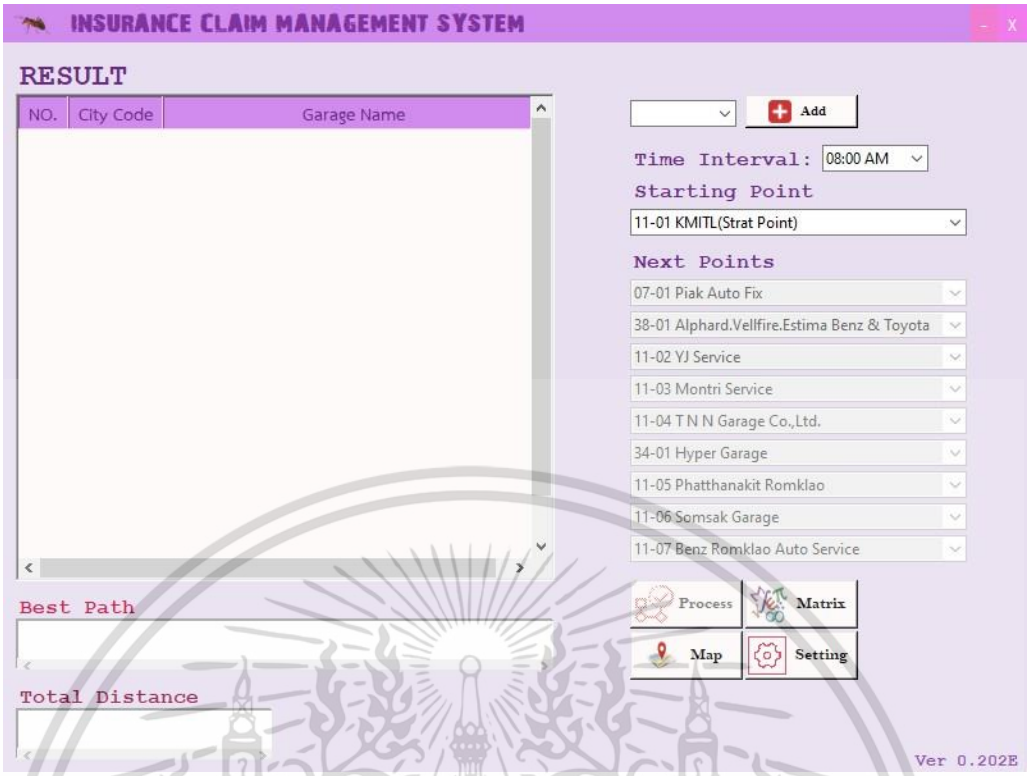
ตารางที่ 4.6 ผลลัพธ์เมื่อกำหนดค่าพารามิเตอร์ระดับสูง (High threshold setting)

Parameters	Iteration = 100, Evaporation rate = 0.09, Alpha = 5, Beta = 5									
Ant Population	20		40		60		80		100	
Algorithm	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt	ACS	ACS+2Opt
Average (km)	114.778	113.426	113.254	113.143	112.926	112.79	112.753	112.779	112.558	112.561
Efficiency (%)	100	100	100	100	100	100	100	100	100	100

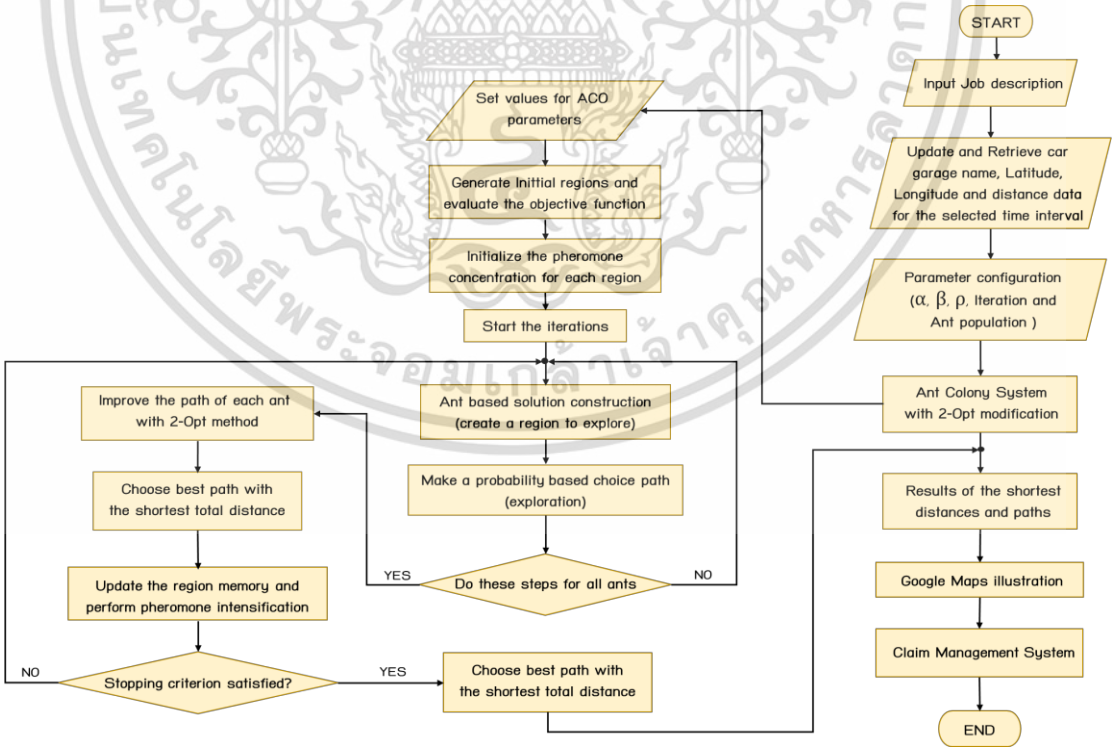
4.2 ผลการพัฒนาระบบ

เมื่อทำการพัฒนาหน้าโปรแกรมสำหรับระบบการจัดการงานสินไหมในกรณีความเสียหายหนักโดยใช้โปรแกรม PyCharm 2021.1.3 (Community Edition) ในการเขียนภาษาไพทอน (Python3.9) จะได้ดังรูปที่ 4.1 โดยกระบวนการภายในได้แสดงขั้นตอนการทำงานของระบบดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 หน้าโปรแกรมระบบการจัดการงานสินไหมในกรณีความเสียหายหนัก



รูปที่ 4.2 แผนภูมิระบบการจัดการสินไหมกรณีความเสียหายหนักด้วยวิธีระบบอาณานิคมมด ที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

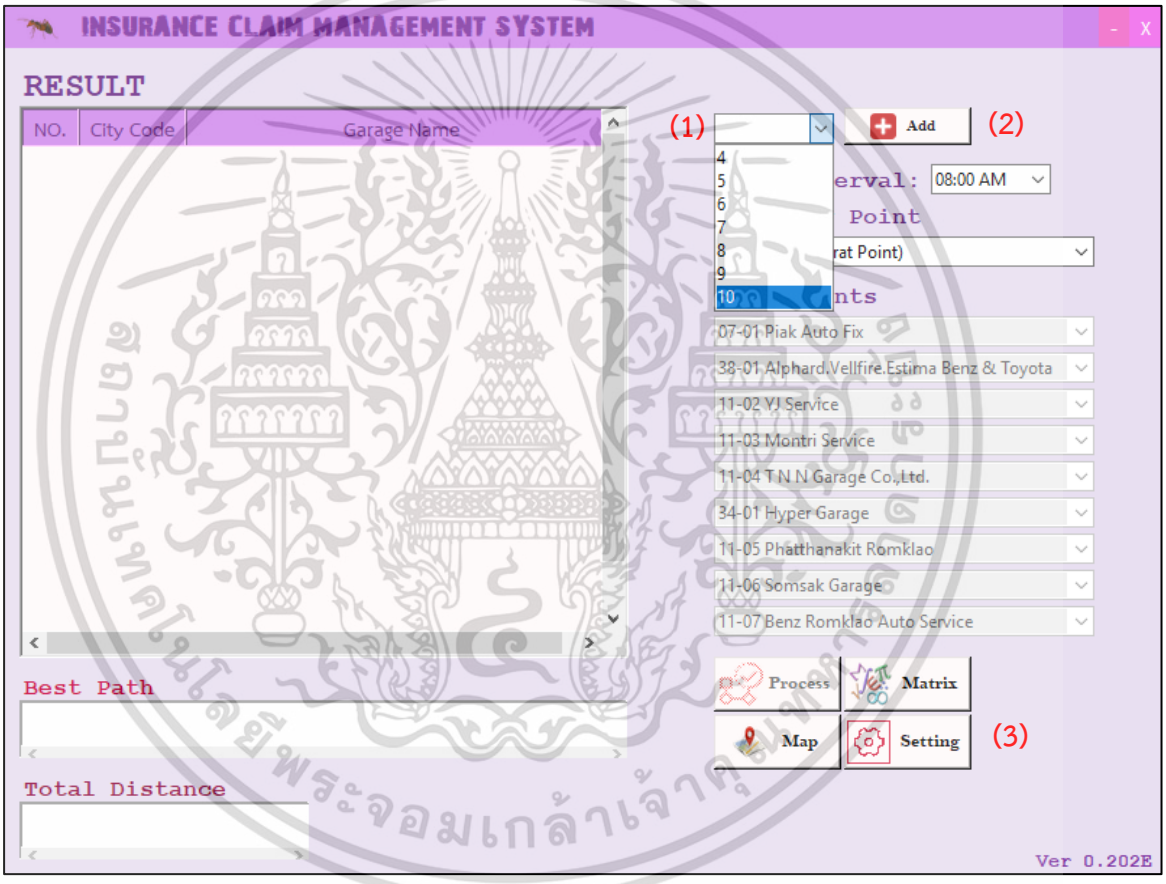
ซึ่งมีขั้นตอนการใช้งานระบบ ดังนี้

เมื่อเปิดตัวโปรแกรมจะแสดงหน้าต่างดังรูปที่ 4.3 หน้าต่างโปรแกรม I.C.M System (1) ซึ่งเป็นโปรแกรมประมวลผลเพื่อหาเส้นทางที่เหมาะสม โดยมีวิธีการใช้งานดังนี้

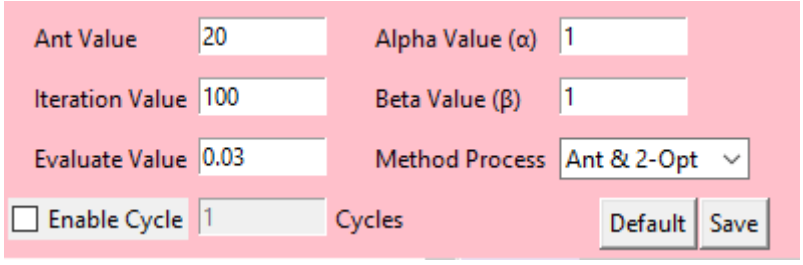
ขั้นตอนที่ (1) เลือกจำนวนจุดทั้งหมดที่ต้องการเดินทาง โดยรวมจุดเริ่มต้นด้วย

ขั้นตอนที่ (2) กดปุ่มเพิ่ม (Add)

ขั้นตอนที่ (3) กดปุ่มตั้งค่า (Setting) เพื่อทำการตั้งค่าตัวแปร จำนวนมด (Ant Population), จำนวนรอบในการวนซ้ำ (Iteration), อัตราการระเหยของค่าฟีโรโมน (Evaporation Rate), ค่าพารามิเตอร์ (α) และค่าพารามิเตอร์ (β) โดยจะปรากฏหน้าต่างดังรูปที่ 4.4 หน้าต่างการตั้งค่า



รูปที่ 4.3 หน้าต่างโปรแกรม I.C.M System (1)



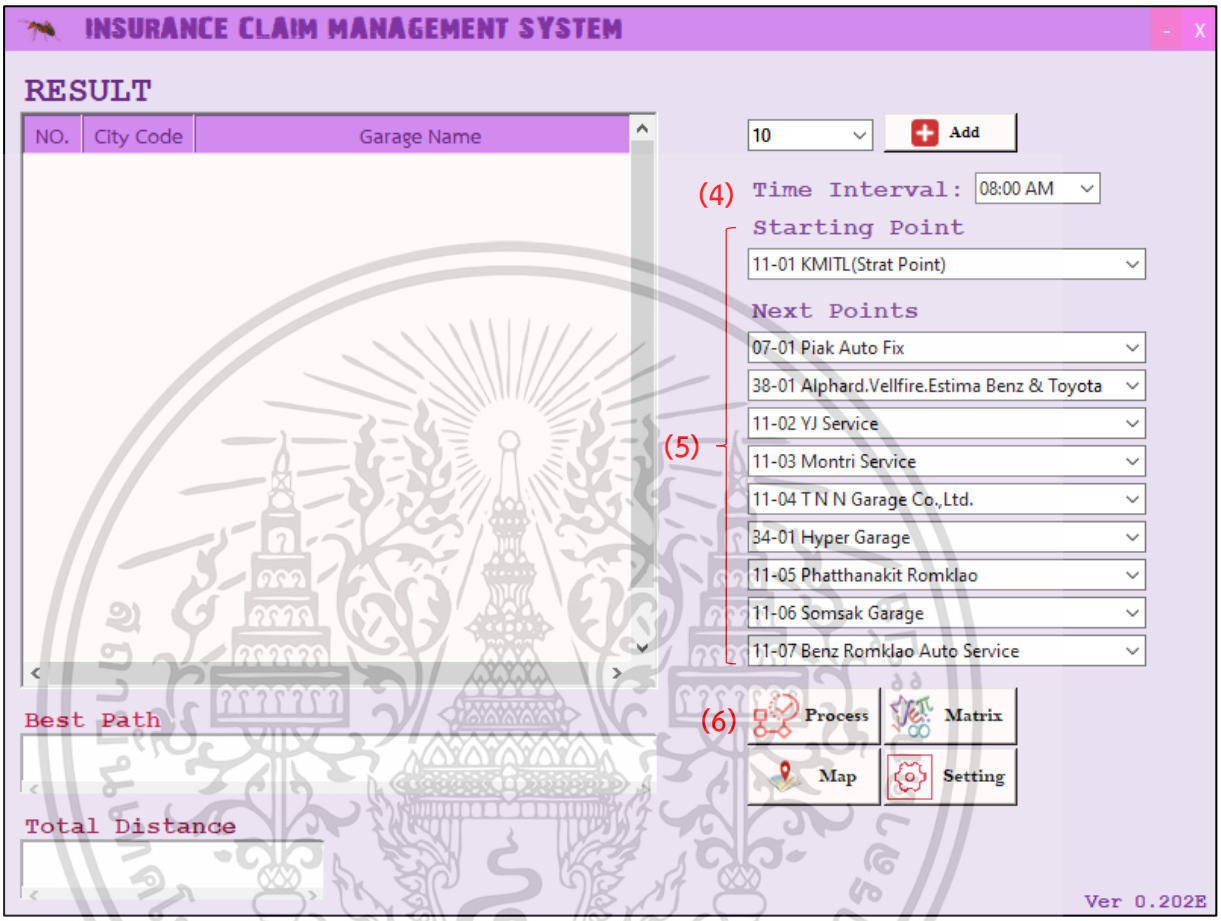
รูปที่ 4.4 หน้าต่างการตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ (4) เลือกช่วงเวลาที่ต้องการใช้ข้อมูล

ขั้นตอนที่ (5) เลือกสถานที่ตามที่ต้องการ

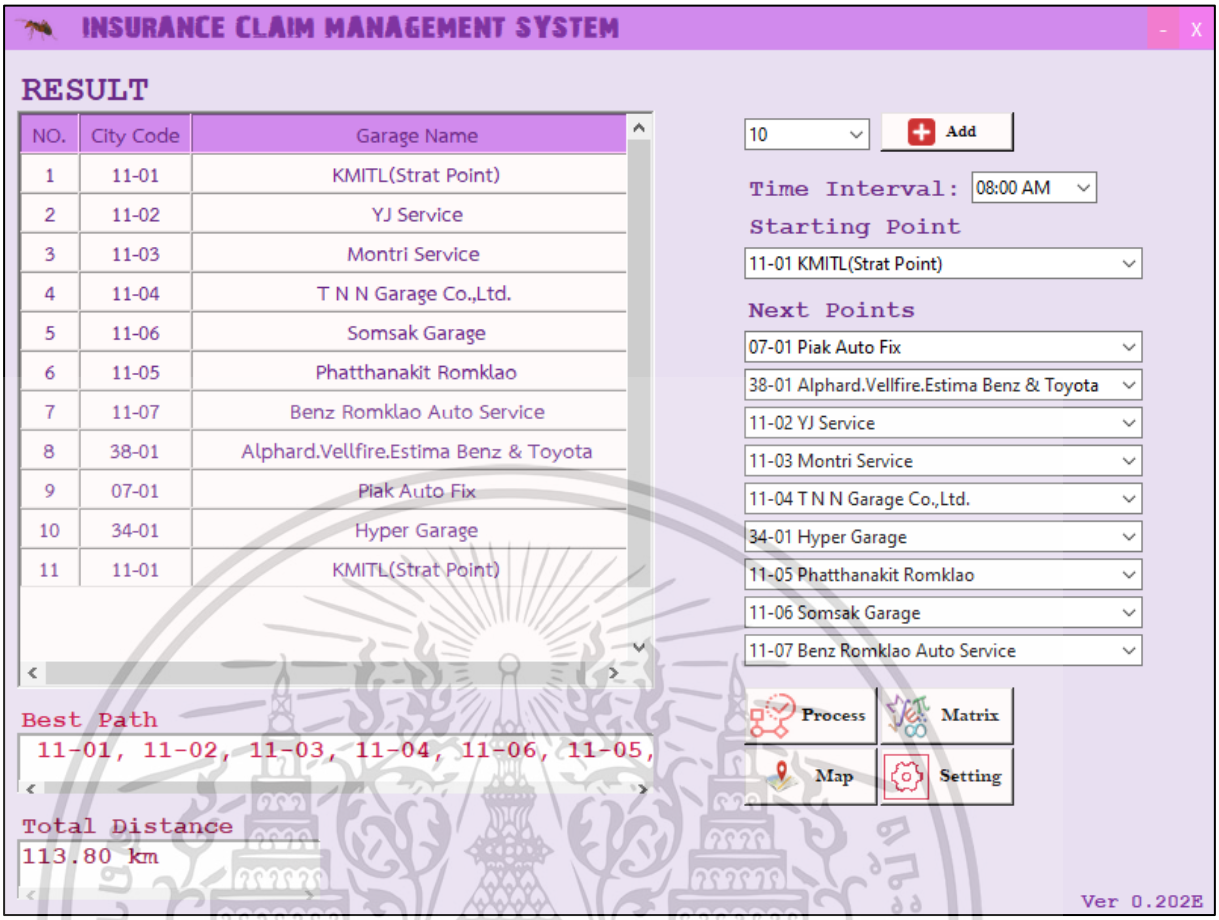
ขั้นตอนที่ (6) กดปุ่มประมวลผล (Process) เพื่อหาเส้นทาง



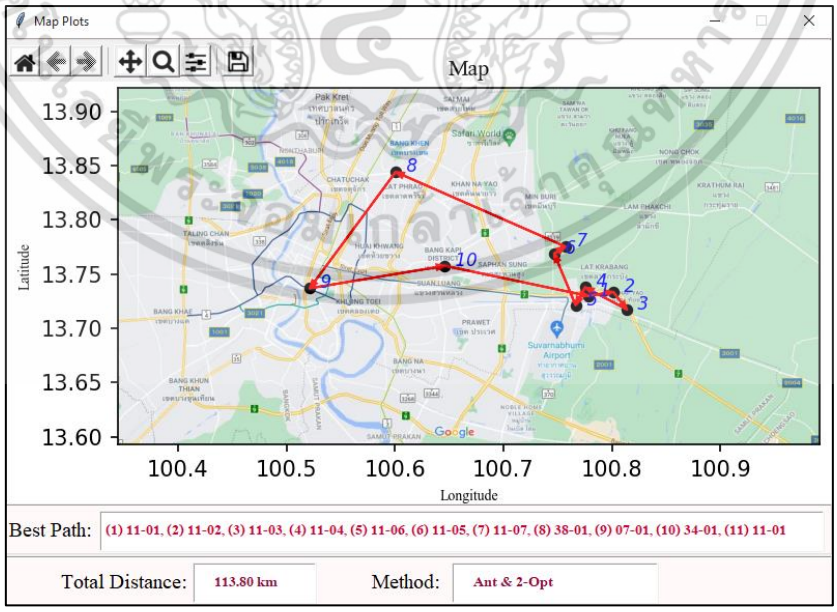
รูปที่ 4.5 หน้าต่างโปรแกรม I.C.M System (2)

ขั้นตอนที่ (7) หลังกดปุ่มประมวลผล (Process) จะแสดงผลเส้นทางดังรูปที่ 4.6 หน้าต่างโปรแกรม I.C.M System หลังประมวลผลเสร็จ และรูปที่ 4.7 Map Plots

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 หน้าต่างโปรแกรม I.C.M System หลังประมวลผลเสร็จ



รูปที่ 4.7 หน้าต่างสำหรับแสดงแผนที่เส้นทางการเดินทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

จากการดำเนินผลการวิจัยเกี่ยวกับการพัฒนาระบบจัดการงานสินไหมการประกันภัยรถยนต์ด้วยวิธีระบบอาณานิคมมดได้ข้อสรุปดังต่อไปนี้

5.1 สรุปผลการวิจัย

งานวิจัยนี้ผู้วิจัยนำเสนอการทดลองเปรียบเทียบ 3 วิธีในการจัดการปัญหาเส้นทางการเดินรถ ซึ่งกำหนดให้มีการค้นหาคำตอบจำนวน 100 คำตอบ ได้แก่ วิธีระบบอาณานิคมมด, วิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง และวิธีเกณฑ์การตัดสินใจบนพื้นฐานไดก์สตรา เพื่อเลือกวิธีที่เหมาะสมให้กับการพัฒนาระบบการจัดการสินไหมการประกันภัยรถยนต์ โดยจากผลการทดสอบพบว่าวิธีระบบอาณานิคมมดที่มีการปรับปรุงคำตอบด้วยวิธีสลับสองตำแหน่ง มีประสิทธิภาพในการให้ผลลัพธ์ที่ดีกว่าสองวิธีที่เหลือ ดังนั้นระบบการจัดการสินไหมการประกันภัยรถยนต์จึงถูกพัฒนาด้วยอัลกอริทึมดังกล่าว และเมื่อระบบถูกนำไปใช้นั่นคือผู้วิจัยหวังว่าระบบที่พัฒนาขึ้นนั้นจะช่วยให้พนักงานคุมราคาได้แนวทางในการตัดสินใจเพื่อให้เกิดการทำงานอย่างมีประสิทธิภาพรวมไปถึงสามารถช่วยวางแผนในการปฏิบัติงานในแต่ละวันได้

5.2 ข้อเสนอแนะ

ค่าพารามิเตอร์ต่าง ๆ ที่ผู้วิจัยทำการทดสอบด้วยข้อมูลที่ผู้วิจัยได้ทำการเก็บข้อมูลมาเองเบื้องต้น โดยหากเป็นข้อมูลในส่วนอื่น ๆ อาจจะมีการใช้ค่าพารามิเตอร์ที่แตกต่างกันออกไปได้ตามแต่ข้อมูลที่ผู้วิจัยนั้นๆ มี ในด้านของโปรแกรมที่ผู้วิจัยได้จัดทำขึ้นนั้นสามารถนำไปพัฒนาต่อด้วยการผูกระบบกับ GPS เพื่อนำทางพนักงานตามเส้นทางต่างๆ แบบ real time เพื่อการแสดงผลลัพธ์ที่ดีขึ้นแต่จะมาพร้อมกับเวลาที่ต้องใช้ในการคำนวณที่มากขึ้นเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] สำนักงานคณะกรรมการกำกับและส่งเสริมการประกอบธุรกิจประกันภัย (คปภ.). 2565. **รายงานภาวะธุรกิจประกันภัยไทย ประจำไตรมาส 1 ปี 2564**. [Online]. เข้าถึงได้จาก : https://www.oic.or.th/th/consumer/education-resources/article.

[2] LMG Insurance Public Company Limited. 2560. **ขั้นตอนการเคลมประกันภัย**. [Online]. เข้าถึงได้จาก : https://www.lmginsurance.co.th/th/Claims/Pages/involve-other-party.aspx#.

[3] อุดม จันทร์จรัสสุข และ กิตติโรจน์ สันติฐายี. 2561. "การเปรียบเทียบวิธีระบบอาณานิคมมดและวิธีระบบมดในการแก้ปัญหาการจัดเส้นทางการเดินรถที่มีความจุรถจำกัด." *วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่.* 25(3) : 199-208.

[4] Laporte, G. 1992. "The Vehicle Routing Problem: An overview of exact and approximate algorithms." *European Journal of Operational Research North-Holland.* 59 : 345-358

[5] Colorni, A., Dorigo, M., and Maniezzo, V. 1991. "Distributed optimization by ant colonies." *European conference on artificial life, Paris, France, Elsevier Publishing.* 134-142.

[6] ผศ. ดร.ระพีพันธ์ ปิตาคะโส. 2554. **วิธีการเมตาฮิวริสติก เพื่อแก้ไขปัญหาการวางแผนการผลิตและการจัดการโลจิสติกส์.** กรุงเทพฯ : ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

[7] Khanna, A., Mishra, A., Tiwari, V., and Gupta, P. N.. 2015."A LITERATURE BASED SURVEY ON SWARM INTELLIGENCE INSPIRED OPTIMIZATION TECHNIQUE." *International Journal of Advanced Technology in Engineering and Science.* 3(1) : 2348-7550.

[8] Mazzeo, S. and Loiseau, I. 2004. "An ant colony algorithm for the capacitated vehicle routing". *Discrete Mathematics.* 18 : 181-186.

[9] Zanakis, S.H. and Evans, J.R.. 1981. "Heuristic "optimization": why, when, and how to use it". *Journal of Heuristics*. 11(5) : 84-91.

[10] Bullnheimer, B., Hartl, R. F. and Strauss, C.. 1997. "Applying the ant system to the vehicle routing problem." *The second International Conference on Metaheuristics - MIC97, Sophia-Antipolis, France.* 285-296.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

[11] Gambardella, L.M., Taillard, E. and Agazzi, G.. 1999. “MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows”. in: Corne, D. Dorigo, M. and Glover, F. (Eds.), New Ideas in Optimization, McGraw-Hill, London. 63-76.

[12] Shaw, P. 1998. “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems.” *International Conference on Principles and Practice of Constraint Programming-CP98.* 1520 : 417-431.

[13] Rochat, Y. and Taillard, E.D. 1995. “Probabilistic Diversification and Intensification in Local Search for Vehicle Routing.” *Journal of Heuristics.* 1 : 147-167.

[14] Taillard, E.D., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J.Y.A. 1997. “Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows.” *Journal of Transportation Science.* 31(2) : 170-186.

[15] ฐิตินนท์ ศรีสุวรรณดี และ รพีพันธ์ ปิตาคะโส. 2555. “การแก้ปัญหาการจัดเส้นทางการขนส่งยานพาหนะด้วยวิธีอาณานิคมมด กระณีศึกษา บริษัทเจียรนัยน้ำดื่มจำกัด.” *วารสารวิจัยมหาวิทยาลัยขอนแก่น.* 17(5) : 706-714.

[16] Wang, L. T., Chang, Y. W. and Cheng, K. T. T.. (2009). **Electronic design automation: synthesis, verification, and test.** United States of America : Morgan Kaufmann.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ข้อมูลและโค้ดในการพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.1 ข้อมูลตำแหน่งอู่ซ่อมรถยนต์ที่ใช้ในการทดสอบระบบ

No.	Garage name	lat , long
1	kmitl(origin point)	13.729908703751873, 100.77822842269966
2	อู่ซ่อมรถยนต์เปี้ยกออโต้ฟิก	13.73803301395852, 100.52166842323673
3	ซ่อมเบนซ์. อัลพาร์ด ลาดปลาเค้า 44	13.844130088200076, 100.60062546927638
4	อู่ซ่อมรถ YJ Service (วายเจ เซอร์วิส)	13.73360588466476, 100.80118645185053
5	อู่ช่าง มนตรี เซอร์วิส	13.721141991439179, 100.8167260387182
6	อู่ TNN ลาดกระบัง	13.73781686294395, 100.7753306963312
7	Hyper Garage อู่ซ่อมรถ พระราม9	13.740112480447753, 100.6338424000351
8	อู่พัฒนกิจ ร่มเกล้า เคาะพ่นสีรถ	13.768895901066603, 100.74679959362065
9	อู่สมศักดิ์ การช่าง	13.721430132161224, 100.76647518175665
10	อู่ซ่อมรถเบนซ์ - เบนซ์ร่มเกล้าออโต้เซอร์วิส	13.775433186073087, 100.75706149097836
11	อู่พัฒนกิจ มีนบุรี เคาะพ่นสีรถ	13.819537583968076, 100.77754080077563
12	อู่ซ่อมรถเจี่ยวุฒิยนต์	13.716280079331046, 100.66965310550651
13	อู่ซ่อมรถชิตณรงค์บริการศรีนครินทร์65	13.674276925730034, 100.64729449084209
14	อู่สุวรรณภูมิเซอร์วิส	13.721521633515001, 100.73805422966755
15	อู่ซ่อมรถ ป.เป็ด	13.802052500094646, 100.71876935098746
16	อู่ซ่อมรถมินิ aodautoshop	13.768957266729872, 100.61257023791161
17	นินจาการาจ อู่ซ่อมรถยนต์	13.741467648940592, 100.61147382385438
18	อู่ซ่อมรถ Honda HKP SPEED	13.703973103258434, 100.63102024768173
19	อู่ซ่อมรถสยามเทคนิค	13.855924719270703, 100.69314077581483
20	นิเทศเซอร์วิส อู่ซ่อมรถเชฟโรเลต	13.836666523901592, 100.66434142807903
21	อู่ช่างพร้อม ซ่อมรถยนต์	13.724208912439229, 100.71833971591452

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ใช้ในการทดสอบระบบ

	kmitl(orig	อู่ซ่อมรถย	ซ่อมเบนซ์.	อู่ซ่อมรถ Y	อู่ช่าง มนต	"อู่ TNN ล	Hyper Ga	อู่พัฒนกิจ	อู่สมศักดิ์ ก	อู่ซ่อมรถเบ	อู่พัฒนกิจ ร	อู่ซ่อมรถเจี๋	อู่ซ่อมรถชิ	อู่สุวรรณภูมิ	อู่ซ่อมรถ ป	อู่ซ่อมรถมิ	นินจาการา	อู่ซ่อมรถ H	อู่ซ่อมรถส	นิเทศเซอร์วิ	อู่ช่างพร้อม
kmitl(origin point)	0	35.5	36.9	2.7	6.6	4.6	23.5	9.9	3.1	11.7	17.5	14.3	26.2	6.5	17.4	27.6	21.9	23.2	27.2	23.3	8.0
อู่ซ่อมรถยนต์เปี้ยกออโต้ฟิก	34.5	0	18.3	35.6	38.6	35.1	17.3	34.6	32.2	36.4	43.2	24.3	19.5	32.7	38.2	17.8	16.7	14.6	32.8	28.3	29.3
ซ่อมเบนซ์. อัลพาร์ด ลาดปลาเค้า 44	31.4	24.0	0	32.5	35.4	31.9	19.2	31.4	29.1	33.2	24.0	24.3	30.3	29.5	21.0	11.9	17.1	24.3	13.6	9.1	26.6
อู่ซ่อมรถ YJ Service (วายเจ เซอร์วิส)	6.8	39.6	40.9	0	3.9	6.3	27.5	12.5	9.1	14.4	20.1	20.5	30.3	12.5	20.0	31.6	26.0	27.2	31.2	27.3	14.0
อู่ช่าง มนตรี เซอร์วิส	6.5	39.2	40.5	3.3	0	5.9	27.1	12.2	7.6	14.0	19.8	20.1	29.9	10.9	19.6	31.2	25.6	26.8	30.8	26.9	12.5
"อู่ TNN ลาดกระบัง (T N N Garage Co.,Ltd.)"	4.5	35.7	37.0	6.3	5.9	0	23.6	8.0	7.7	9.8	15.6	16.6	26.4	9.1	15.4	27.7	22.1	23.3	27.3	23.4	10.6
Hyper Garage อู่ซ่อมรถ พระราม9	17.8	18.6	19.7	18.9	21.8	18.3	0	17.8	15.4	19.6	26.4	7.4	9.1	15.9	21.5	10.4	2.6	6.6	25.2	21.3	12.6
อู่พัฒนกิจ ร่มเกล้า เคาะพ่นสีรถ	11.1	34.7	24.7	11.4	13.7	8.2	22.6	0	8.0	2.0	10.9	15.6	25.4	7.5	7.6	18.7	21.1	22.3	16.7	22.4	9.0
อู่สมศักดิ์ การช่าง	7.9	33.0	34.3	9.0	12.9	11.3	20.9	7.4	0	9.2	18.2	11.7	23.8	3.9	14.8	25.0	19.4	20.6	24.6	20.7	5.5
อู่ซ่อมรถเบนซ์ - เบนซ์ร่มเกล้าออโต้เซอร์วิส	12.9	38.6	25.1	15.3	15.5	10.0	26.5	8.5	11.9	0	11.2	19.5	29.4	11.4	7.9	24.6	25.0	26.2	17.0	17.0	12.9
อู่พัฒนกิจ มีนบุรี เคาะพ่นสีรถ	21.7	43.1	22.9	21.2	24.4	18.9	35.1	14.4	17.8	13.1	0	23.9	33.8	17.3	7.7	27.7	33.0	30.7	14.9	14.9	18.8
อู่ซ่อมรถเจี๋ยวุฒิยนต์	15.5	24.8	26.1	16.6	19.5	16.1	11.1	15.5	10.7	17.3	24.0	0	9.0	7.7	19.0	16.9	9.6	6.4	22.7	18.8	6.9
อู่ซ่อมรถชิตณรงค์บริการศรีนครินทร์65	24.2	19.2	30.2	25.3	28.2	24.7	13.0	22.0	21.8	23.8	32.8	8.0	0	13.7	27.8	20.9	11.4	4.9	31.6	27.7	12.8
อู่สุวรรณภูมิเซอร์วิส	8.0	32.9	34.2	9.0	12.9	8.6	20.8	6.3	3.2	8.1	17.0	13.8	19.5	0	13.7	24.9	19.3	20.5	24.5	20.6	4.8
อู่ซ่อมรถ ป.เป็ด	18.6	35.6	17.5	18.9	21.2	15.7	23.5	12.1	15.5	9.5	8.9	16.4	26.2	15.0	0	22.3	21.9	23.2	9.4	9.4	16.5
อู่ซ่อมรถมินิ aodautoshop	28.3	18.5	12.1	29.4	32.3	22.7	7.1	19.2	25.9	21.0	25.3	12.3	22.0	26.4	17.2	0	8.8	12.3	17.8	13.3	23.1
นินจาการาจ อู่ซ่อมรถยนต์(Ninja Garage Pattanakarn)	22.5	16.1	18.1	23.6	26.5	23.1	2.6	22.5	20.2	24.3	31.2	8.6	11.1	20.6	26.2	8.9	0	8.6	23.8	19.3	17.3
อู่ซ่อมรถ Honda HKP SPEED	20.5	16.0	25.1	21.6	24.6	21.1	10.3	20.6	15.8	22.4	29.0	5.7	5.1	12.7	24.0	15.9	8.7	0	27.8	23.9	11.9
อู่ซ่อมรถสยามเทคนิค	26.3	34.9	17.1	27.4	30.3	26.8	27.4	26.3	23.9	17.7	16.5	22.2	32.1	24.4	13.5	20.0	25.2	29.0	0	6.3	21.5
นิเทศเซอร์วิส อู่ซ่อมรถเซฟโรเลต	22.4	37.5	12.7	23.4	26.4	22.9	25.4	22.4	20.0	24.2	15.0	18.3	28.1	20.5	13.8	17.5	23.9	25.1	8.0	0	17.6
อู่ช่างพร้อม ซ่อมรถยนต์	11.1	29.8	26.7	12.2	16.0	14.4	17.7	9.4	6.4	11.2	20.1	6.8	17.3	3.3	17.3	21.8	16.2	10.9	21.1	17.1	0

ตารางที่ 2 ข้อมูลระยะทางระหว่างอู่ซ่อมรถที่ใช้ในการทดสอบระบบ

โค้ดที่ใช้ในการพัฒนาระบบ ประกอบด้วยรูปที่ 1-11

```python
import tkinter as tk
import numpy as np
from tkinter import ttk
from tkinter import *
import time
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.backends.backend_tkagg import NavigationToolbar2Tk
import re
import os
import openpyxl
from pathlib import Path
from PIL import ImageTk
import PIL.Image
try:
    from configparser import ConfigParser
except ImportError:
    import ConfigParser

with open('garage-data-int.txt', 'rb') as f2:
    latlong = f2.read()
latlong = latlong.decode("utf-8")
latlong = re.split(r"[,\n ]", latlong)
latlong = [x for x in latlong if x]
latlong = np.round(np.array(latlong, dtype="float64"), decimals=15)
array1 = np.reshape(latlong, (21, 2), order='A')
with open('garage-data-Name.txt', 'rb') as f:
    contents = f.read()
contents = contents.decode("utf-16")
contents = contents.split('\r\n')
INI_u = ConfigParser()
INI_u.read('./base/Setting.ini')  # ini file 000000000000
ant_origin = INI_u.getint('SETTING', 'ant_origin')
ite_origin = INI_u.getint('SETTING', 'ite_origin')
e_origin = INI_u.getfloat('SETTING', 'e_origin')  # evaporation rate
alpha_origin = INI_u.getint('SETTING', 'alpha_origin')  # pheromone factor
beta_origin = INI_u.getint('SETTING', 'beta_origin')  # visibility factor
Solu = INI_u.getint('SETTING', 'method')  # Select Method
Method = ('Decision Criteria', 'Ant-ACS', 'Ant & 2-Opt')
cycle_E = INI_u.getint('SETTING', 'cycle_enable')  # Enable cycle
Ccycle = INI_u.getint('SETTING', 'cycle')  # Number of cycle

time_string = time.strftime('%I:%M:%S %p')
big_letters = list(map(chr, range(ord('A'), ord('Z') + 1)))
loop_combo = 4
d = np.ones((loop_combo, loop_combo))
g = str(contents)
g = g.split(",")
Number_city = (4, 5, 6, 7, 8, 9, 10)
position = np.array(np.zeros((1, 30), dtype=int))
cbop_x = np.array((0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), dtype=object)

def Matrix():
    global d, xlsx_file, xxx, wb_obj, sheetAll, sheet1
    xlsx_file = Path(filename)
    xxx = openpyxl.workbook
    wb_obj = openpyxl.load_workbook(xlsx_file)
    wb_obj.active
    sheetAll = wb_obj.sheetnames  # Read the active sheet
    sheet1 = wb_obj[sheetAll[0]]
    d = np.ones((loop_combo, loop_combo))
    for i in range(loop_combo):
        for j in range(loop_combo):
            if i == j:
                d[i, j] = 0
            if i != j:
                Value_1 = float(sheet1[str(big_letters[int(position[0, j])]) + str(position[0, i] + 1)].value)
                d[i, j] = float(Value_1)
    Cycle_check()

def openfile(selc):
    global filename
    if selc == 0:
        filename = r'DataGarage - 08.00 - 10.20.xlsx'
    if selc == 1:
        filename = r'DataGarage - 11.00 - 13.20.xlsx'
    if selc == 2:
        filename = r'DataGarage - 15.20 - 17.00.xlsx'
```

รูปที่ ก.1 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 1 library และการดึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if selc == 3:
        filename = r'DataGarage - 17.30 - 19.20.xlsx'

def DecitionTree():
    global d, loop_combo, DTroute, pathDT, AntMedthod_return, Result_A
    AntMedthod_return = 1
    pathDT = 0
    DTroute = np.ones((1, loop_combo + 1), dtype=int)
    LS2 = []
    LS4 = []
    LS3 = 0
    for i in range(loop_combo - 1):
        LS1 = list(d[LS3, :])
        for j1 in range(len(LS2)):
            LS1[DTroute[0, j1] - 1] = 0
        LS2.append(LS3)
        for j3 in range(len(LS1)):
            if int(LS1[j3]) != 0:
                LS4.append(LS1[j3])
        DisTance1 = min(LS4)
        DTroute[0, i + 1] = LS1.index(DisTance1) + 1
        LS3 = LS1.index(DisTance1)
        LS4 = []
    for i in range(loop_combo):
        pathDT = pathDT + d[DTroute[0, i] - 1, DTroute[0, i + 1] - 1]
    if cycle_E == 1:
        Result_A = " "
        aAns2 = " "
        aAns3 = " "
        for i in range(loop_combo + 1):
            aAns1 = g[int(DTroute[0, i]) - 1].split(" ", 1)
            aAns2 = str(aAns2) + "(" + str(i + 1) + ")" + " " + str(aAns1[0])
            if i != loop_combo:
                aAns2 = str(aAns2) + ", "
        Result_A = str(str(aAns2) + str(" , ") + str(pathDT) + str(' km'))
        file_text.write(Result_A + '\n')
    if cycle_E == 0 or loop_cycles == Ccycle - 1:
        Dpm()
        TextEditAnswer()
        winmap()

def AntMedthod():
    global best_route, path1, visibility, store01, dist_min_loc, AntMedthod_return
    global dist, path0, store02, dist1, aAns2, Result_A
    iteration = ite_origin
    m = ant_origin
    n = loop_combo  # Number of citys
    e = e_origin  # evaporation rate
    alpha = alpha_origin  # pheromone factor
    beta = beta_origin  # visibility factor
    store01 = np.zeros((m, n + 1))  # store path changed
    dist_cost = np.zeros((m, 1))  # intializing total_distance_of_tour with zero
    np.seterr(divide='ignore', invalid='ignore')
    visibility = 1 / d
    visibility = np.nan_to_num(visibility, copy=True, posinf=0)
    neta = np.power(visibility, beta)
    pheromone = 1 * (np.ones((n, n)) - np.identity(n))
    min_rute = np.zeros((1, n + 1))
    for ite in range(iteration):
        pheromone = np.power(pheromone, alpha)
        Att = np.multiply(neta, pheromone)
        rute = np.ones((m, n + 1))
        rute[:, 0] = 1  # initial starting and ending positon of every ants '1' i.e city '1'
        for i in range(m):  # Ants
            sct = 0  # start city
            gone = np.ones(n)
            gone[0] = 0
            for j in range(n - 1):  # find path each ant
                q = np.count_nonzero(gone == 1)
                cum_prob = np.zeros(q)  # intializing cummulative probability array to zeros
                explor = np.zeros(q)  # intializing exploration
                q1 = np.nonzero(gone == 1)
                q1 = q1[0]
                s_Att = 0
                for k in range(q):
                    s_Att = s_Att + Att[sct, q1[k]]  # sum
                    explor[k] = Att[sct, q1[k]]
                explor = np.divide(explor, s_Att)
```

รูปที่ ก.2 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 2 อัลกอริทึมสำหรับประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                explor = np.divide(explor, s_Att)
                cum_prob = np.cumsum(explor)
                r = np.random.random_sample()  # random no in [0,1)
                city = np.nonzero(cum_prob > r)[0] + 1
                city = city[0]
                city = q1[city - 1] + 1
                sct = city - 1
                rute[i, j + 1] = int(city)  # adding city to route
                gone[city - 1] = 0
            if Solu == 2:
                fix_route = np.array(rute[i, :], dtype=int)
                store02 = np.zeros((1, n + 1))  # path for roll
                z = 0
                z1 = 0
                for l1 in range(n - 1):  # 2-opt
                    if l1 == 0:
                        mem_route = np.array(rute[i, :], dtype=int)
                        pathM = 0
                        for k in range(n):  # find distance of  fix_route
                            pathM = pathM + d[int(mem_route[k] - 1), int(mem_route[k + 1] - 1)]
                    if l1 + 3 > n - 1:
                        z1 = z1 + 1
                        z = z1
                    if l1 + 3 <= n - 1:
                        z = z + 3
                    store02 = np.array(rute[i, :], dtype=int)
                    store02 = np.delete(store02, l1 + 1)
                    store02 = np.insert(store02, z, fix_route[l1 + 1])
                    dist1 = 0
                    for l2 in range(n):  # find distance of store02
                        dist1 = dist1 + d[int(store02[l2] - 1), int(store02[l2 + 1] - 1)]
                    if dist1 <= pathM:
                        pathM = dist1
                        mem_route[:] = store02[:]
                    if l1 + 3 <= n - 1:
                        z = z - 3
                        z = z + 1
                rute[i, :] = mem_route[:]
                dist_cost[i, 0] = pathM
        for i in range(m):  # find minimum path
            dist = 0
            for j in range(n):
                dist = dist + d[int(rute[i, j] - 1), int(rute[i, j + 1] - 1)]
            if Solu == 1:
                dist_cost[i, 0] = dist  # memory distance of each path
            if i == 0:
                min_dist = dist
                min_rute = rute[i, :]
            if i != 0 and dist < min_dist:
                min_dist = dist
                min_rute = rute[i, :]
        pheromone = (1 - e) * pheromone  # evaporation of pheromone with (1-e)
        for i in range(n):  # updating the pheromone with minimum path
            dt = 1 / min_dist
            pheromone[int(min_rute[i] - 1), int(min_rute[i + 1] - 1)] = pheromone[int(min_rute[i] - 1), int(
                min_rute[i + 1] - 1)] + dt
            pheromone[int(min_rute[i + 1] - 1), int(min_rute[i] - 1)] = pheromone[int(min_rute[i + 1] - 1), int(
                min_rute[i] - 1)] + dt
    store01[:, :] = np.array(rute[:, :], dtype=int)
    dist_min_loc = np.argmin(dist_cost)  # finding location of minimum of dist_cost
    best_route = store01[dist_min_loc, :]  # intializing current traversed as best route
    path1 = float(np.min(dist_cost))
    if cycle_E == 1:
        Result_A = " "
        aAns2 = " "
        aAns3 = " "
        for i in range(loop_combo + 1):
            aAns1 = g[int(store01[dist_min_loc, i] - 1)].split(" ", 1)
            aAns2 = str(aAns2) + "(" + str(i + 1) + ")" + " " + str(aAns1[0])
            if i != loop_combo:
                aAns2 = str(aAns2) + ", "
        Result_A = str(str(aAns2) + str(" , ") + str(path1) + str(' km'))
        file_text.write(Result_A + '\n')
    if cycle_E == 0 or loop_cycles == Ccycle - 1:
        Dpm()
        TextEditAnswer()
        AntMedthod_return = 1
        winmap()
```

รูปที่ ก.3 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 3 อัลกอริทึมสำหรับประมวลผล (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
235
236 def Cycle_check():
237     button_2["state"] = "disabled"
238     global loop_cycles, file_text
239     file_text = open('Result_Method.txt', 'w+')
240     if cycle_E == 1:
241         file_text.write('\t' + str(Method[Solu]) + '\n\n')
242         if Solu == 1 or Solu == 2:
243             file_text.write("Ant = " + str(ant_origin) + '\n')
244             file_text.write("Iteration = " + str(ite_origin) + '\n')
245             file_text.write("Evaporation rate = " + str(e_origin) + '\n')
246             file_text.write("Alpha = " + str(alpha_origin) + '\n')
247             file_text.write("Beta = " + str(beta_origin) + '\n\n')
248         for loop_cycles in range(Ccycle):
249             if Solu == 1 or Solu == 2:
250                 AntMedthod()
251             elif Solu == 0:
252                 DecitionTree()
253         file_text.close()
254         os.startfile('Result_Method.txt')
255     if cycle_E == 0:
256         if Solu == 1 or Solu == 2:
257             AntMedthod()
258         elif Solu == 0:
259             DecitionTree()
260     button_2["state"] = "normal"
261
262 def GetVar():
263     g[0] = str(cbo_01.get())
264     for i in range(loop_combo - 1):
265         g[i + 1] = str(cbop_x[i].get())
266     for i in range(21 - loop_combo - 1):
267         g[21 - i] = '0'
268     g1 = str(cbo_02.get())
269     for i in range(4):
270         if g1 == Time_line[i]:
271             openfile(i)
272     dimention1()
273
274 def dimention1():  # 0000000000000
275     global position
276     for j in range(int(loop_combo)):
277         for i in range(len(contents)):
278             if str(contents[i]) == str(g[j]):
279                 position[0, j] = i + 1
280     Matrix()
281
282 def loopVar():
283     global loop_combo
284     loop_combo = int(c.get())
285
286 def savesetting():
287     global ant_origin, ite_origin, e_origin, alpha_origin, beta_origin, Solu, cycle_E, Ccycle
288     ant_origin = int(g_e1.get())
289     ite_origin = int(g_e2.get())
290     e_origin = float(g_e3.get())  # evaporation rate
291     alpha_origin = int(g_e4.get())  # pheromone factor
292     beta_origin = int(g_e5.get())  # visibility factor
293     Solu = g_e6.current()
294     cycle_E = int(g_cbV1.get())
295     Ccycle = int(g_e8.get())
296     INI_u.set('SETTING', 'ant_origin', str(ant_origin))
297     INI_u.set('SETTING', 'ite_origin', str(ite_origin))
298     INI_u.set('SETTING', 'e_origin', str(e_origin))
299     INI_u.set('SETTING', 'alpha_origin', str(alpha_origin))
300     INI_u.set('SETTING', 'beta_origin', str(beta_origin))
301     INI_u.set('SETTING', 'method', str(Solu))
302     INI_u.set('SETTING', 'cycle_enable', str(cycle_E))
303     INI_u.set('SETTING', 'cycle', str(Ccycle))
304     with open('./base/Setting.ini', 'w') as configfile: # save to a file
305         INI_u.write(configfile)
306     gui.grab_release()
307     closenewwin(1)
308
309 def defaultsetting():
310     global ant_origin, ite_origin, e_origin, alpha_origin, beta_origin
311     global g_e1, g_e2, g_e3, g_e4, g_e5, INI_c, g_e8
312     INI_c = ConfigParser()
```

รูปที่ ก.4 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 4 การรับค่าจากหน้าต่างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    INI_c.read('./base/Setting.ini')
    g_e1.delete(0, END)
    g_e1.insert(END, str(INI_c.getint('DEFAULT', 'ant_origin')))
    g_e2.delete(0, END)
    g_e2.insert(END, str(INI_c.getint('DEFAULT', 'ite_origin')))
    g_e3.delete(0, END)
    g_e3.insert(END, str(INI_c.getfloat('DEFAULT', 'e_origin')))
    g_e4.delete(0, END)
    g_e4.insert(END, str(INI_c.getint('DEFAULT', 'alpha_origin')))
    g_e5.delete(0, END)
    g_e5.insert(END, str(INI_c.getint('DEFAULT', 'beta_origin')))
    g_e6.current(INI_c.getint('DEFAULT', 'method'))
    g_e8['state'] = 'normal'
    g_e8.delete(0, END)
    g_e8.insert(END, str(INI_c.getint('DEFAULT', 'cycle')))
    state_C()

myappid = 'mycompany.myproduct.subproduct.version' # arbitrary string
ctypes.windll.shell32.SetCurrentProcessExplicitAppUserModelID(myappid)
V_hide = tk.Tk()
root = tk.Toplevel(V_hide)
V_hide.attributes("-alpha",0.0)
window_height = 600
window_width = 800
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
x_cordinate = int((screen_width / 2) - (window_width / 2))
y_cordinate = int((screen_height / 2) - (window_height / 2))
root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
V_hide.iconbitmap(r".\ico\ANT.ico")
root.resizable(width=False, height=False)
root.update_idletasks()
root.overrideredirect(True)
root.configure(background='#e8dff1')

def move_window(event):
    root.geometry("+" + str(int(event.x)) + "+" + str(int(event.y)))
    root.geometry('+{0}+{1}'.format(event.x_root, event.y_root))
    return
def Close1(event):
    global close_button
    close_button['bg'] = 'red'
def Close2(event):
    global close_button
    close_button['bg'] = '#f27cee'
def Hidden1(event):
    global close_button
    hide_button['bg'] = 'red'
def Hidden2(event):
    global close_button
    hide_button['bg'] = '#f27ccf'
def hide_screen():
    root.withdraw()
    V_hide.iconify()
def showscreen(event):
    root.deiconify()
    V_hide.deiconify()
def screen_appear(event):
    root.overrideredirect(1)
def closeMain():
    root.destroy
    V_hide.destroy()
def onRootIconify(event):
    root.withdraw()

back_ground = "#d18aed"
title_bar = Frame(root, bg=back_ground, relief='raised', bd=0, highlightcolor=back_ground, highlightthickness=0)
canvas = Canvas(title_bar, bg=back_ground, bd=0, relief='ridge', highlightthickness=0, width=50, height=30)
canvas.pack(side=LEFT, fill="both")
image1 = PIL.Image.open('./ico/ANT.png')
image1 = image1.resize((30, 30), PIL.Image.ANTIALIAS)
image1 = ImageTk.PhotoImage(image1)
canvas.create_image(7, 0, image=image1, anchor=NW)
close_button = Button(title_bar, text='x', command=closeMain, bg='#f27cee', padx=5, pady=2,
                      activebackground="red",bd=0, font="bold", fg='white',
                      activeforeground="white", highlightthickness=0)
hide_button = Button(title_bar, text='-', command=hide_screen, bg='#f27ccf', padx=5, pady=2,
                     activebackground="red",bd=0, font="bold", fg='white',
```

รูปที่ ก.5 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 5 การสร้างหน้าต่างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    activeforeground="white", highlightthickness=0)
title_window = "INSURANCE CLAIM MANAGEMENT SYSTEM"
title_name = Label(title_bar, text=title_window, bg=back_ground, compound=tk.CENTER, fg="#7f34a1",
                   font=("28 Days Later", 14, 'bold'))
title_bar.pack(side=TOP, fill=X)
title_name.pack(side=LEFT, fill=X)
close_button.pack(side=RIGHT)
hide_button.pack(side=RIGHT)
x_axis = None
y_axis = None

def restate_cbo():
    loopVar()
    button_2["state"] = "normal"
    a = 0
    for i in range(loop_combo - 1):
        cbop_x[a]["state"] = "readonly"
        root.update()
        a = a + 1

    for j in range(10 - loop_combo):
        cbop_x[a]["state"] = "disabled"
        root.update()
        a = a + 1

def combo_test(count):
    Lc2 = tk.Label(root, text='Next Points', bg='#e8dff1', font=("Courier", 12, 'bold'), fg="#6a2b86")
    Lc2.place(x=490, y=190)
    global cbop_x
    for i in range(count):
        cbop_x[i] = ttk.Combobox(root, state="readonly", width=40, values=contents)
        cbop_x[i].current(i + 1)
        cbop_x[i].place(x=490, y=215 + 25 * i)
        cbop_x[i]["state"] = "disabled"
        root.update()

def Numberonly(S):
    if S.isdigit():
        return True
    root.bell()
    return False

def donothing():
    pass

def winmapAns():
    global tr2_2, tr2_1, DTroute, pathDT, Solu, tr2_3, Method
    aAns2 = " "
    aAns3 = " "
    for i in range(loop_combo + 1):
        if Solu == 1 or Solu == 2:
            aAns1 = g[int(store01[dist_min_loc, i] - 1)].split(" ", 1)
        if Solu == 0:
            aAns1 = g[int(DTroute[0, i]) - 1].split(" ", 1)
        aAns2 = str(aAns2) + "(" + str(i + 1) + ")" + " " + str(aAns1[0])
        if i != loop_combo:
            aAns2 = str(aAns2) + ", "
    tr2_2.delete('1.0', END)
    tr2_2.insert(INSERT, str(aAns2))
    tr2_2.configure(font=("Angsana New", 16, 'bold'), fg="#c70039")
    tr2_1.delete('1.0', END)
    tr2_1.configure(font=("Angsana New", 16, 'bold'), fg="#900c3f")
    if Solu == 1 or Solu == 2:
        tr2_1.insert(INSERT, "      " + str("{:.2f}".format(round(path1, 3)) + " " + 'km'))
    if Solu == 0:
        tr2_1.insert(INSERT, "      " + str("{:.2f}".format(round(pathDT, 3)) + " " + 'km'))
    tr2_3.delete('1.0', END)
    tr2_3.configure(font=("Angsana New", 16, 'bold'), fg="#900c3f")
    tr2_3.insert(INSERT, "      " + str(Method[Solu]))

    global pathplot
    pathplot = np.zeros((2 * loop_combo, 2))
    xa2 = 0
    for xa1 in range(loop_combo):
        if Solu == 1 or Solu == 2:
            kk1 = int(store01[dist_min_loc, xa1]) - 1
            kk2 = int(store01[dist_min_loc, xa1 + 1]) - 1
            kk3 = int(position[0, kk1]) - 1
```

รูปที่ ก.6 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 6 การสร้างหน้าต่างโปรแกรมและแสดงผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            kk4 = int(position[0, kk2]) - 1
        if Solu == 0:
            kk1 = int(DTroute[0, xa1]) - 1
            kk2 = int(DTroute[0, xa1 + 1]) - 1
            kk3 = int(position[0, kk1]) - 1
            kk4 = int(position[0, kk2]) - 1
        xa2 += 1
        if xa1 == (loop_combo):
            xa2 = 1
        ax.text(array1[kk3, 1], array1[kk3, 0], "  " + str(int(xa2)), fontsize=8,
                fontstyle='italic', color="blue")
        ax.arrow(array1[kk3, 1], array1[kk3, 0], (array1[kk4, 1] - array1[kk3, 1]),
                 (array1[kk4, 0] - array1[kk3, 0]),head_width=0.005, width=0.0001,
                 color='red', length_includes_head=True)
        ax.scatter(array1[kk3, 1], array1[kk3, 0], c="black", s=20)

def winmap():  # map after process
    global imp, figureM1, root2, ax, bar, toolbar
    root2 = tk.Tk()
    root2.title('Map Plots')  # ▯▯▯▯▯▯▯▯▯▯
    root2.resizable(width=False, height=False)
    root2.geometry('800x550')
    root2.configure(background='white')
    root2.update_idletasks()
    global tr2_2, tr2_1, Fr2_1, Fr2_2, Fr2_3, tr2_3
    Fr2_1 = tk.Frame(root2, bg='snow', relief='groove', bd=2, highlightcolor="snow",
                     highlightthickness=1, width=800,height=450)
    Fr2_1.pack()
    img = plt.imread("Map Ues jpg02.jpg")
    figureM1 = plt.Figure(figsize=[5.5, 3], dpi=150)
    figureM1.subplots_adjust(top=0.92, right=0.95, left=0.13)
    ax = figureM1.add_subplot()
    bar = FigureCanvasTkAgg(figureM1, Fr2_1)
    toolbar = NavigationToolbar2Tk(bar, Fr2_1)
    toolbar.place(x=0, y=0)
    bar.get_tk_widget().place(x=0, y=0)
    ax.imshow(img, extent=[100.3449690000, 100.9911030000, 13.5927330000, 13.9215370000])
    ax.set_xlabel('Longitude', fontsize=16, font="angsana new")
    ax.set_ylabel('Latitude', fontsize=16, font="angsana new")
    ax.set_title('Map', font="angsana new", fontsize=16)
    Fr2_2 = tk.Frame(root2, bg='snow', relief='sunken', bd=1, highlightcolor="snow",
                     highlightthickness=1, width=800,height=50)
    Fr2_2.pack()
    Lb2_2 = tk.Label(Fr2_2, text="Best Path:", font=("Times New Roman", 15), bg="snow")
    Lb2_2.place(x=0, y=10)
    tr2_2 = tk.Text(Fr2_2, wrap="none", font=("Times New Roman", 15), highlightthickness=1, bd=1)
    tr2_2.place(x=90, y=5)
    tr2_2.place_configure(width=700, height=40)
    Fr2_3 = tk.Frame(root2, bg='snow', relief='sunken', bd=1, highlightcolor="snow",
                     highlightthickness=1, width=800,height=50)
    Fr2_3.pack()
    Lb2_1 = tk.Label(Fr2_3, text="Total Distance:", font=("Times New Roman", 15), bg="snow")
    Lb2_1.place(x=50, y=10)
    tr2_1 = tk.Text(Fr2_3, wrap="none", font=("Times New Roman", 15), highlightthickness=1, bd=1)
    tr2_1.place(x=180, y=5)
    tr2_1.place_configure(width=120, height=40)
    Lb2_3 = tk.Label(Fr2_3, text="Method:", font=("Times New Roman", 15), bg="snow")
    Lb2_3.place(x=350, y=10)
    tr2_3 = tk.Text(Fr2_3, wrap="none", font=("Times New Roman", 15), highlightthickness=1, bd=1)
    tr2_3.place(x=430, y=5)
    tr2_3.place_configure(width=200, height=40)
    if AntMedthod_return == 1:
        winmapAns()
    if AntMedthod_return == 0 or AntMedthod_return == None:
        pass

def newwin1():  # Setting Window
    close_button["state"] = "disabled"
    hide_button["state"] = "disabled"
    button_sc["state"] = "disabled"
    button_2["state"] = "disabled"
    button_3["state"] = "disabled"
    button_4["state"] = "disabled"
    button_X["state"] = "disabled"
    global g_l1, g_l2, g_l3, g_l4, g_l5, g_e1, g_e2, g_e3, g_e4, g_e5, gui, g_e6, Method, g_e7, g_l8, g_e8, g_cbV1
    gui = tk.Tk()
    W_H = 130
    W_W = 400
```

รูปที่ ก.7 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 7 หน้าต่างแสดงผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    screen_w = gui.winfo_screenwidth()
    screen_h = gui.winfo_screenheight()
    x_c = int((screen_w / 2) - (W_W / 2))
    y_c = int((screen_h / 2) - (W_H / 2))
    gui.geometry("{}x{}+{}+{}".format(W_W, W_H, x_c, y_c))
    gui.overrideredirect(1)
    gui.resizable(width=False, height=False)
    gui.configure(background='pink')
    gui.update_idletasks()
    gui.focus_set()
    gui.grab_set()
    vcmd = (gui.register(Numberonly), '%S')
    g_l1 = Label(gui, text="Ant Population", background='pink')
    g_l1.place(x=15, y=10)
    g_e1 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e1.insert(END, str(ant_origin))
    g_e1.place(x=110, y=10)
    g_l2 = Label(gui, text="Iteration ", background='pink')
    g_l2.place(x=15, y=40)
    g_e2 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e2.insert(END, str(ite_origin))
    g_e2.place(x=110, y=40)
    g_l3 = Label(gui, text="Evaporation Rate", background='pink')
    g_l3.place(x=15, y=70)
    g_e3 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e3.insert(END, str(e_origin))
    g_e3.place(x=110, y=70)
    g_l4 = Label(gui, text="Alpha (α)", background='pink')
    g_l4.place(x=185, y=10)
    g_e4 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e4.insert(END, str(alpha_origin))
    g_e4.place(x=280, y=10)
    g_l5 = Label(gui, text="Beta (β)", background='pink')
    g_l5.place(x=185, y=40)
    g_e5 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e5.insert(END, str(beta_origin))
    g_e5.place(x=280, y=40)
    g_l6 = Label(gui, text="Method Process", background='pink')
    g_l6.place(x=185, y=70)
    g_e6 = ttk.Combobox(gui, state="readonly", width=12, values=Method)
    g_e6.current(Solu)
    g_e6.place(x=280, y=70)
    g_cbV1 = IntVar(gui, cycle_E)
    g_e7 = tk.Checkbutton(gui, text="Enable Cycle", variable=g_cbV1,
                          command=state_C,bg="pink",activebackground="pink")
    g_e7.place(x=15, y=100)
    g_l8 = Label(gui, text="Cycles", background='pink')
    g_l8.place(x=175, y=100)
    g_e8 = tk.Entry(gui, width=10, validatecommand=vcmd)
    g_e8.insert(END, str(Ccycle))
    g_e8.place(x=110, y=100)
    state_C()
    button_g1 = tk.Button(gui, text="Save", command=savesetting)
    button_g1.place(x=350, y=100)  # 00000000000000
    button_g2 = tk.Button(gui, text="Default", command=defaultsetting)
    button_g2.place(x=300, y=100)  # 00000000000000

def state_C():  # for cycle in setting window
    if g_cbV1.get() == 0:
        g_e8['state'] = 'disabled'
    if g_cbV1.get() == 1:
        g_e8['state'] = 'normal'

def closenewwin(f):
    if f == 1:
        gui.destroy()
        button_sc["state"] = NORMAL
        close_button["state"] = NORMAL
        hide_button["state"] = NORMAL
        button_2["state"] = NORMAL
        button_3["state"] = NORMAL
        button_4["state"] = NORMAL
        button_X["state"] = NORMAL
    if f == 0:
        donothing()

def newwin2():  # Display Matrix After run Antmedthod
    global gui_1
```

รูปที่ ก.8 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 8 การสร้างหน้าต่างตั้งค่าตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gui_1 = tk.Tk()
    gui_1.title('Matrix')  # ๐๐๐๐๐๐๐๐๐
    gui_1.geometry('900x400')
    gui_1.configure(background='gray')
    gui_1.update_idletasks()
    gui_1.focus_set()
    Lb2 = tk.Label(gui_1, text='Matrix of Distance Between Points', bg='grey', font=("Courier", 12))
    Lb2.place(x=10, y=5)
    Lb1 = tk.Label(gui_1, text=str(d), font=("Courier", 18))
    Lb1.place(x=10, y=40)

def TextEditAnswer():  # RESULT ๐๐๐๐๐๐๐๐๐๐๐๐๐๐๐๐
    for widget in myframe.winfo_children():
        widget.destroy()
    Table_in_Frame()
    DpmAf()
    for i in range(loop_combo + 1):
        if Solu == 1 or Solu == 2:
            aAns1 = g[int(store01[dist_min_loc, i] - 1)].split(" ", 1)
        elif Solu == 0:
            aAns1 = g[int(DTroute[0, i]) - 1].split(" ", 1)
        aAns2 = aAns1[0]
        aAns3 = aAns1[1]
        x1 = tk.Entry(myframe, width=5, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'),
                      justify='center',readonlybackground="snow", relief='groove')
        x1.insert(INSERT, str(i + 1))
        x1['state'] = 'readonly'
        x1.grid(row=i + 1, column=0)
        x2 = tk.Entry(myframe, width=10, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'),
                      justify='center',readonlybackground="snow", relief='groove')
        x2.insert(INSERT, str(aAns2))
        x2['state'] = 'readonly'
        x2.grid(row=i + 1, column=1)
        x3 = tk.Entry(myframe, width=42, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'),
                      justify='center',readonlybackground="snow", relief='groove')
        x3.insert(INSERT, str(aAns3))
        x3['state'] = 'readonly'
        x3.grid(row=i + 1, column=2)

def TextAnswer():  # RESULT ๐๐๐๐๐๐๐๐๐๐
    global text, textContainer, textVsb, textHsb, TextLead1
    global wrapper1, mycanvas, yscrollbar, xscrollbar, myframe
    TextLead1 = tk.Label(root, text="RESULT", fg="#6a2b86", font=("Courier", 18, 'bold'), bg="#e8dff1")
    TextLead1.place(x=10, y=40)
    wrapper1 = LabelFrame(root, relief="sunken")
    mycanvas = Canvas(wrapper1, bg="snow", highlightthickness=0, width=480, height=360)
    mycanvas.grid(row=0, column=0, sticky='wesn')
    xscrollbar = ttk.Scrollbar(wrapper1, orient="horizontal", command=mycanvas.xview)
    xscrollbar.grid(row=1, column=0, sticky='we')
    yscrollbar = ttk.Scrollbar(wrapper1, orient="vertical", command=mycanvas.yview)
    yscrollbar.grid(row=0, column=1, sticky='ns')
    myframe = Frame(mycanvas, borderwidth=0, bg="snow", width=480, height=360)
    mycanvas.create_window((0, 0), window=myframe, anchor="nw", height=480)
    wrapper1.place(x=10, y=70)
    mycanvas.configure(yscrollcommand=yscrollbar.set, xscrollcommand=xscrollbar.set)
    mycanvas.bind('<Configure>', lambda e: mycanvas.configure(scrollregion=mycanvas.bbox('all')))
    Table_in_Frame()
    root.update()

def Table_in_Frame():  # RESULT ๐๐๐๐๐๐๐๐๐๐๐๐
    global mycanvas
    x1 = tk.Entry(myframe, width=5, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'), justify='center',
                  readonlybackground="#d18aed", relief='groove')
    x1.insert(INSERT, "NO.")
    x1['state'] = 'readonly'
    x1.grid(row=0, column=0)
    x2 = tk.Entry(myframe, width=10, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'), justify='center',
                  readonlybackground="#d18aed", relief='groove')
    x2.insert(INSERT, "City Code")
    x2['state'] = 'readonly'
    x2.grid(row=0, column=1)
    x3 = tk.Entry(myframe, width=42, fg="#6a2b86", font=('TH SarabunPSK', 14, 'bold'), justify='center',
                  readonlybackground="#d18aed", relief='groove')
    x3.insert(INSERT, "Garage Name")
    x3['state'] = 'readonly'
    x3.grid(row=0, column=2)
    mycanvas.configure(scrollregion=mycanvas.bbox('all'))

```

รูปที่ ก.9 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 9 การแสดงผลลัพธ์แบบตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
703 def Dpm():  # Path & Distance Box
704     Lb3 = tk.Label(root, text='Total Distance', bg='#e8dff1', font=("Courier", 12, 'bold'), fg="#908c3f")
705     Lb3.place(x=10, y=530)
706     Lb4 = tk.Label(root, text='Best Path', bg='#e8dff1', font=("Courier", 12, 'bold'), fg="#c70839")
707     Lb4.place(x=10, y=460)
708     global textContainer_1, text_1, textHsb_1
709     textContainer_1 = tk.Frame(root, borderwidth=1, relief="sunken")
710     text_1 = tk.Text(textContainer_1, width=50, height=50, wrap="none", font=("Times New Roman", 15))
711     textHsb_1 = tk.Scrollbar(textContainer_1, orient="horizontal", command=text_1.xview)
712     text_1.configure(xscrollcommand=textHsb_1.set)
713     text_1.place_configure(width=420, height=400)
714     textHsb_1.place_configure(y=30, width=420, height=10)
715     textContainer_1.place_configure(width=420, height=40)
716     textContainer_1.place(x=10, y=480)  # Best Path
717     global textContainer_2, text_2, textHsb_2
718     textContainer_2 = tk.Frame(root, borderwidth=1, relief="sunken")
719     text_2 = tk.Text(textContainer_2, width=50, height=50, wrap="none", font=("Times New Roman", 15))
720     textHsb_2 = tk.Scrollbar(textContainer_2, orient="horizontal", command=text_2.xview)
721     text_2.configure(xscrollcommand=textHsb_2.set)
722     text_2.place_configure(width=200, height=400)
723     textHsb_2.place_configure(y=30, width=200, height=10)
724     textContainer_2.place_configure(width=200, height=40)
725     textContainer_2.place(x=10, y=550)  # Total Distance
726     root.update()
727
728 def DpmAf():
729     global text_1, text_2, aAns1
730     aAns2 = " "
731     aAns1 = []
732     for i in range(loop_combo + 1):
733         if Solu == 1 or Solu == 2:
734             aAns1 = g[int(store01[dist_min_loc, i] - 1)].split(" ", 1)
735         elif Solu == 0:
736             aAns1 = g[int(DTroute[0, i]) - 1].split(" ", 1)
737         aAns2 = str(aAns2) + str(aAns1[0])
738         if i != loop_combo:
739             aAns2 = str(aAns2) + ", "
740     text_1.delete('1.0', END)
741     text_1.insert(INSERT, str(aAns2))
742     text_1.configure(font=("Courier", 12, 'bold'), fg="#c70839")
743     text_2.delete('1.0', END)
744     text_2.configure(font=("Courier", 12, 'bold'), fg="#908c3f")
745     if Solu == 1 or Solu == 2:
746         text_2.insert(INSERT, str("{:.2f}".format(round(path1, 3)) + " " + 'km'))
747     elif Solu == 0:
748         text_2.insert(INSERT, str("{:.2f}".format(round(pathDT, 3)) + " " + 'km'))
749
750 time_1 = tk.Label(root, compound=tk.CENTER, bg="#e8dff1")
751 time_1.place(x=600, y=580)
752 tx_2 = tk.Label(root, text="Ver 0.202E", compound=tk.CENTER, bg="#e8dff1",
753                 fg="#7f34a1", font=("Courier", 10, 'bold'))
754 tx_2.place(x=710, y=580)
755 tx_3 = tk.Label(root, text='Time Interval:', bg='#e8dff1', font=("Courier", 12, 'bold'), fg="#6a2b86")
756 tx_3.place(x=490, y=110)
757 Time_line = ('08:00 AM', '11:00 AM', '03:20 PM', '05:30 PM')
758 cbo_02 = ttk.Combobox(root, state="readonly", width=10, values=Time_line)
759 cbo_02.current(0)
760 cbo_02.place(x=640, y=110)
761 Lc1 = tk.Label(root, text='Starting Point', bg='#e8dff1', font=("Courier", 12, 'bold'), fg="#6a2b86")
762 Lc1.place(x=490, y=135)
763 cbo_01 = ttk.Combobox(root, state="readonly", width=40, values=contents)
764 cbo_01.current(0)
765 cbo_01.place(x=490, y=160)
766 c = ttk.Combobox(root, state="readonly", width=10, values=Number_city)
767 c.place(x=490, y=75)
768 combo_test(9)  # ๏๏๏๏๏๏๏๏๏๏๏๏๏๏๏
769 TextAnswer()  # ๏๏๏๏๏๏๏๏๏
770 Dpm()
771
772 def Bx_S(event):
773     global W_bx, BxSS
774     W_bx = Toplevel(root)
775     W_bx.wm_overrideredirect(1)
776     W_bx.wm_geometry('+{0}+{1}'.format(event.x_root, event.y_root))
777     Lxs = tk.Label(W_bx, text="Add The number of Points", justify=LEFT, background="#ffffe0",
778                    relief=SOLID, borderwidth=1,font=("tahoma", "8", "normal"))
779     Lxs.pack()
780 def Bx_H(event):
```

รูปที่ ก.10 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 10 การแสดงผลลัพธ์แบบตาราง (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
781     W_bx.destroy()
782     W_bx.update()
783
784 im_B_x = PIL.Image.open('./ico/plus.png')
785 im_B_x = im_B_x.resize((18, 18), PIL.Image.ANTIALIAS)
786 im_B_x = ImageTk.PhotoImage(im_B_x)
787 button_X = tk.Button(root, text="  Add  ", command=restate_cbo, image=im_B_x, bg="snow",
788                      compound="left", width=80,font=("Bell MT", 8, 'bold'))  # 00000000000000000
789 button_X.place(x=580, y=71)
790 button_X.bind("<Enter>",Bx_S)
791 button_X.bind("<Leave>",Bx_H)
792 im_B_2 = PIL.Image.open('./ico/process.png')
793 im_B_2 = im_B_2.resize((30, 30), PIL.Image.ANTIALIAS)
794 im_B_2 = ImageTk.PhotoImage(im_B_2)
795 button_2 = tk.Button(root, bg="snow", text=" Process ", command=GetVar, image=im_B_2,
796                      compound="left", width=80,font=("Bell MT", 10, 'bold'))
797 button_2.place(x=490, y=450)  # 000000000000000
798 button_2["state"] = "disabled"
799 im_B_3 = PIL.Image.open('./ico/Lmatrix.png')
800 im_B_3 = im_B_3.resize((30, 30), PIL.Image.ANTIALIAS)
801 im_B_3 = ImageTk.PhotoImage(im_B_3)
802 button_3 = tk.Button(root, text="  Matrix  ", command=newwin2, image=im_B_3, compound="left", width=80,
803                      font=("Bell MT", 10, 'bold'), bg="snow")
804 button_3.place(x=580, y=450)  # Matrix
805 im_B_4 = PIL.Image.open('./ico/Lmap.png')
806 im_B_4 = im_B_4.resize((30, 30), PIL.Image.ANTIALIAS)
807 im_B_4 = ImageTk.PhotoImage(im_B_4)
808 button_4 = tk.Button(root, text="  Map  ", command=winmap, image=im_B_4, compound="left", width=80,
809                      font=("Bell MT", 10, 'bold'), bg="snow")
810 button_4.place(x=490, y=490)  # Map Plot
811 AntMedthod_return = 0
812 im_B_sc = PIL.Image.open('./ico/Gear.png')
813 im_B_sc = im_B_sc.resize((30, 30), PIL.Image.ANTIALIAS)
814 im_B_sc = ImageTk.PhotoImage(im_B_sc)
815 button_sc = tk.Button(root, text="  Setting  ", command=newwin1, image=im_B_sc, compound="left", width=80,
816                       font=("Bell MT", 10, 'bold'), bg="snow")  # 000000000000 line1
817 button_sc.place(x=580, y=490)  # 000000000000 line2
818 title_bar.bind('<B1-Motion>', move_window)
819 title_name.bind('<B1-Motion>', move_window)
820 canvas.bind('<B1-Motion>', move_window)
821 title_bar.bind('<Button-3>', showscreen)
822 title_bar.bind('<Map>', screen_appear)
823 close_button.bind('<Enter>', Close1)
824 close_button.bind('<Leave>', Close2)
825 hide_button.bind('<Enter>', Hidden1)
826 hide_button.bind('<Leave>', Hidden2)
827 V_hide.bind("<Unmap>", onRootIconify)
828 V_hide.bind("<Map>", showscreen)
829
830 root.mainloop()
831
832
```

รูปที่ ก.11 โค้ดที่ใช้ในการพัฒนาระบบ ส่วนที่ 11 การสร้างปุ่มและช่องรับค่าของหน้าต่างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เปเปอร์ที่ได้รับการตีพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Proceedings of the
26th Annual Meeting in Mathematics (AMM 2022)
School of Mathematics, Institute of Science
Suranaree University of Technology, Thailand

การศึกษาและพัฒนาระบบการจัดการงานสินไหม ในกรณีความเสียหายหนักของการประกันภัยรถยนต์

เบญจมาภรณ์ ศรีอัมพร[1,‡] บุษยมาส พิมพ์พรรณชาติ และ เทิดขวัญ ช้างเผือก

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 10520

บทคัดย่อ

เนื่องด้วยส่วนงานสินไหมความเสียหายหนัก เป็นส่วนงานการพิจารณาสินไหมราคาจัดซ่อมรถยนต์ที่มีความเสียหายหนัก หรือราคาเสนอซ่อมจากอู่มากกว่า 100,000 บาท จึงได้ศึกษาและพัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนักของการประกันภัยรถยนต์ เพื่อจัดเก็บข้อมูลสินไหมความเสียหายหนักให้อยู่ในรูปแบบที่สามารถเรียกใช้งานได้อย่างมีประสิทธิภาพ และออกแบบแนวทางการช่วยตัดสินใจจากปัจจัยพื้นฐานด้านบุคคลากรให้สอดคล้องกับระดับปริมาณงาน โดยระบบการจัดการงานสินไหมความเสียหายหนักที่ถูกสร้างขึ้นบน Microsoft Excel ซึ่งนำข้อมูลจากการเก็บตัวอย่างอู่ซ่อมรถยนต์ในพื้นที่กรุงเทพมหานครที่เป็นส่วนงานการพิจารณาสินไหมของการประกันภัยรถยนต์ โดยจัดเจ้าหน้าที่ออกอนุมัติราคาให้ที่ศูนย์ฯ ผู้วิจัยจึงได้สร้างระบบดังกล่าว สำหรับการแก้ปัญหาการจัดเส้นทางการเดินรถเพื่อไปยังสถานบริการซ่อมรถยนต์ต่าง ๆ โดยสุ่มข้อมูลมาประมวลผลวิเคราะห์เส้นทางที่เหมาะสมในการเดินทางของเจ้าหน้าที่ จากผลการทดสอบพบว่าหน่วยงานสามารถมองเห็นภาพการบริหารงานจัดการสินไหมที่เหมาะสมมากขึ้น และสถานะงานสินไหมได้อย่างถูกต้อง ตัวอย่างเช่นการตัดสินใจเพิ่มคน หรือวางแผนการเร่งพิจารณาสินไหม

คำสำคัญ: สินไหมกรณีความเสียหายหนัก, ความเสียหายหนัก, สินไหมทดแทน, การประกันภัยรถยนต์

1 บทนำ

การประกันภัยรถยนต์มีบทบาทมากขึ้นในปัจจุบัน จากรายงานภาวะธุรกิจประกันภัยไทย ประจำไตรมาส 1 ปี 2564 พบว่าประกันรถยนต์ภาคสมัครใจปรับตัวเพิ่มขึ้นมากกว่าช่วงเดียวกันของปีก่อน 2.68% [1] แสดงให้เห็นว่าการทำ

[†]ผู้นำเสนอ [‡]ผู้แต่งหลัก
อีเมล: 62605021@kmitl.ac.th (เบญจมาภรณ์ ศรีอัมพร), busayamas.pi@kmitl.ac.th, thurdkwan.ch@kmitl.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกันภัยรถยนต์มีความสำคัญมากขึ้น และเมื่อเกิดอุบัติเหตุเกี่ยวกับรถยนต์แบบมีคู่กรณี ที่เรียกว่า “เคลมสด”รถประกัน หรือรถคู่กรณีมีความเสียหายมาก จนไม่สามารถขับเคลื่อนได้ เมื่อบริษัทประกันภัยได้รับแจ้งเหตุดังกล่าว บริษัทฯ จะจัดส่งพนักงานออกตรวจสอบ ณ ที่เกิดเหตุ โดยผู้ขับขี่รถประกันหรือรถคู่กรณีจะได้รับใบหลักฐานในการติดต่อค่าสินไหมจากพนักงานตรวจสอบอุบัติเหตุ หลังจากนั้นหากผู้เสียหายต้องการซ่อมรถกับอู่ หรือสถานบริการในเครือของบริษัทประกันภัย บริษัทจะดำเนินการประเมินราคาและส่งรถเข้าสถานบริการเพื่อดำเนินการซ่อมได้เลย แต่หากผู้เสียหายต้องการซ่อมห้าง คือผู้เอาประกันหรือเจ้าของรถต้องการซ่อมรถที่อู่ หรือสถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัย ผู้เอาประกันต้องนำใบเสนอราคาค่าซ่อมที่ให้อู่ หรือสถานบริการที่ไม่ใช่ในเครือของบริษัทประกันภัยประเมินราคามาติดต่อที่บริษัทประกันภัย เพื่อให้บริษัทฯ จัดส่งพนักงานคุมราคาออกอนุมัติราคาให้ที่สถานบริการนั้นๆ [2] โดยทั้งหมดที่กล่าวมานั้นเป็นหนึ่งในขั้นตอนการแจ้งเคลม ทั้งนี้บริษัทประกันภัยได้จัดเก็บข้อมูลผู้เอาประกัน ผู้ขับขี่ รายละเอียดการเกิดอุบัติเหตุต่างๆ รวมถึงบันทึกรายการค่าสินไหม ผ่านใบรายการความเสียหาย จากขั้นตอนข้างต้น ผู้วิจัยจึงเล็งเห็นว่าข้อมูลดังกล่าวมีความสำคัญที่สามารถนำไปวิเคราะห์ในด้านอื่น ๆเพิ่มเติมได้ ซึ่งมองการพิจารณาสินไหมเป็น 2 ส่วน คือ ส่วนงานพิจารณาสินไหมความเสียหายที่ต่ำกว่า 100,000 บาท และส่วนงานพิจารณาสินไหมความเสียหายเกินกว่า 100,000 บาท ส่วนงานพิจารณาสินไหมความเสียหายเกินกว่า 100,000 บาท เป็นสินไหมกรณีความเสียหายหนักเนื่องจากบริษัทฯ ต้องจัดส่งพนักงานคุมราคาออกอนุมัติราคาที่สถานบริการนั้นๆการจัดส่งพนักงานออกนอกสถานที่นั้นมีค่าใช้จ่ายในการดำเนินงาน ผู้วิจัยจึงนำมาสร้างระบบการจัดการงานสินไหมในกรณีความเสียหายหนักของการประกันภัยรถยนต์ โดยระบบจะจัดเก็บข้อมูลที่จำเป็น เช่น สถานบริการซ่อมรถยนต์ ชื่อพนักงานคุมราคา ระยะเวลาที่ใช้ ราคาเสนอ ราคายุติ วันที่ และอื่น ๆ อีกทั้งระบบจะช่วยประมวลผลการจัดเส้นทางการเดินรถโดยใช้พื้นฐานจาก Dijkstra’s Algorithm เพื่อไปยังสถานบริการซ่อมรถยนต์ต่าง ๆ ตามที่กำหนด รวมถึงจัดเก็บข้อมูลที่อาจมีการระบุไว้ในใบรายการความเสียหาย ที่อาจไม่ได้จัดเก็บข้อมูลดังกล่าวแบบเป็นระบบ เพื่อให้ข้อมูลดังกล่าวอยู่ในรูปแบบที่สามารถเรียกใช้งานได้อย่างมีประสิทธิภาพ

นิยามศัพท์

เคลมสด	หมายถึง การเคลมที่เกิดขึ้นทันทีหลังเกิดอุบัติเหตุรถชนกัน โดยยังมีคู่กรณีอยู่ด้วย (เมื่อเกิดอุบัติเหตุระหว่างรถยนต์กับรถยนต์)
การประกันภัย	หมายถึง การบริหารความเสี่ยงภัยวิธีหนึ่งซึ่งโอนความเสี่ยงภัยจากผู้เอาประกันภัยไปสู่บริษัทผู้รับประกันภัยโดยทำสัญญาความคุ้มครองในกรมธรรม์ประกันภัย และผู้เอาประกันภัยต้องจ่ายค่าเบี้ยประกันภัยให้กับบริษัทผู้รับประกันภัย เมื่อเกิดความเสียหายผู้รับประกันภัยจะชดใช้ค่าสินไหมทดแทนตามที่ได้ระบุความคุ้มครองไว้ในกรมธรรม์ประกันภัยตามสัญญาที่ได้ตกลงกันไว้ [3]
ผู้เอาประกันภัย	หมายถึง บุคคลธรรมดาหรือนิติบุคคลก็ได้ มีหน้าที่ส่งเบี้ยประกันภัยให้แก่ผู้รับประกันภัยจนครบกำหนดตามที่ตกลงกันไว้ในสัญญา [3]
สินไหม	หมายถึง การเรียกร้องค่าสินไหมทดแทน,ค่าสินไหมทดแทน (CLAIM) (1) การเรียกร้องของผู้เอาประกันภัยที่จะให้มีการชดใช้ค่าเสียหายตามสัญญาประกันภัย หรือสัญญาค้ำประกัน (2) จำนวนเงินค่าเสียหายที่แท้จริงหรือโดยประมาณ [4]
ซ่อมห้าง	หมายถึง การนำรถเข้าซ่อมกับศูนย์บริการรถยนต์ของยี่ห้อนั้นๆ โดยจะได้รับการบริการจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่างผู้เชี่ยวชาญที่ได้รับการรับรองมาตรฐานจากทางศูนย์บริการ รวมถึงชิ้นส่วนอะไหล่ที่มีมาตรฐานของยี่ห้อรถยนต์นั้นๆ โดยตรง อีกทั้งมีระบบการทำงานที่เป็นมาตรฐานและมีประสิทธิภาพ รวมถึงการบริการที่ดี แต่แลกมาด้วยค่าใช้จ่ายที่สูง [5]

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 กราฟ (Graph)

กราฟประกอบด้วย 2 เซตได้แก่ เซตจุดยอด (Vertex set: V) และเซตเส้นขอบ (Edge set: E) โดยแสดงเป็น G(V,E). ซึ่งจุดยอดสามารถเรียกอีกอย่างว่า โหนด (Node) และเส้นขอบสามารถเรียกอีกอย่างว่า ส่วนโค้ง (Arcs) หรือ แขนง (Branches) โดยทั้งสองเซตนั้นต้องเชื่อมต่อกันเป็นโครงข่าย

2.1.2 Dijkstra's Algorithm

ขั้นตอนวิธีสำหรับหาวิถีสั้นที่สุดสำหรับกราฟที่มีการถ่วงน้ำหนักของเส้นเชื่อมที่ไม่เป็นลบ โดยหาระยะทางสั้นที่สุดจากจุดยอดหนึ่งไปยังจุดใด ๆ ในกราฟโดยหาเส้นทางที่สั้นที่สุดไปทีละจุด จนครบตามที่ต้องการ [6]

2.1.3 การใช้ PivotTable

PivotTable คือเครื่องมือในโปรแกรม Microsoft Excel ใช้สำหรับการสรุปข้อมูลจำนวนมาก จะถูกสร้างจากข้อมูลใน Worksheet หรือ Range ที่มีข้อมูลต้องการจะใช้สรุปผล เราสามารถนำเอาหัวข้อในแต่ละคอลัมน์มาใส่ใน Pivot Table เพื่อใช้สรุปผลโดย จะมีด้วยกัน 4 ส่วนที่เราจะต้องลากเอาข้อมูลมาวางเพื่อสรุปผล ส่วนประกอบของ Pivot Table ได้แก่ 1. Row Label = ใช้สรุปข้อมูลที่หัวของตารางทางด้านซ้าย

2. Column Label = ใช้สรุปข้อมูลที่หัวของตารางทางด้านบน

3. ∑ Values = สรุปผลของข้อมูลที่ต้องการ โดยจะต้องกำหนดว่าจะใช้สูตรใด SUM, AVERAGE, MAX, MIN, COUNT เป็นต้น

4. Filter = ใช้กำหนดว่าจะใช้คอลัมน์ใดเพื่อใช้กรองข้อมูล โดยการแสดงผลจะเป็น Drop down list เพื่อให้เลือกว่าเราจะกรองข้อมูลใด [7]

2.1.4 การใช้ Data Validation

การใช้ Data Validation คือความสามารถในการกำหนดว่าช่อง Cell นั้นสามารถระบุข้อมูลหรือค่าอะไรได้บ้างและสามารถกำหนดข้อมูลที่ระบุใน Cell เป็นแบบ dropdown list ได้

1. เลือกพื้นที่ของเซลล์ที่ต้องการกำหนดเกณฑ์ในการป้อนข้อมูล แล้วเลือกคำสั่ง Data --> Data Validation
2. คลิกเลือกแท็บ Setting ของกรอบ Data Validation กำหนดเงื่อนไขจากตัวเลือกต่าง ๆ [8]

2.1.5 การใช้ VBA

Visual Basic for Applications (VBA) คือการใช้ภาษา Visual Basic ในการเขียนโค้ดควบคุมโปรแกรมประยุกต์อื่นๆ ซึ่งการทำงานส่วนใหญ่ของ Excel VBA จะเหมือนกับการทำงานทั่วไปของ Excel แต่บางครั้งถือเป็นขั้นตอนย่อยๆ จำนวนมาก จึงเขียนชุดคำสั่ง VBA ที่ทำให้ Excel ทำงานอัตโนมัติ รวดเร็ว และง่ายขึ้น [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานวิจัยที่เกี่ยวข้อง

ปัญหาการจัดเส้นทางสำหรับยานพาหนะ(Vehicle Routing Problem: VRP) เป็นปัญหาที่มีลักษณะคล้ายกับปัญหาการเดินทางของพนักงานขาย (Traveling salesman problem: TSP) ซึ่งเกี่ยวกับการจัดเส้นทางการขนส่งสินค้า จากศูนย์กระจายสินค้าไปยังลูกค้าที่กระจายตามจุดต่าง ๆ ภายใต้เงื่อนไขหรือข้อจำกัดในการขนส่ง เช่น ความจุของรถบรรทุก กรอบเวลาในการส่งสินค้า เป็นต้น โดยมีวัตถุประสงค์เพื่อหาเส้นทางการเดินรถที่สั้นที่สุด หรือมีต้นทุนการขนส่งที่น้อยที่สุด [10]

(นิรุชา อาภามงคลรัตน์และ เปรมรัตน์ พูลสวัสดิ์,2560) พัฒนาระบบวางแผนการเดินทางท่องเที่ยวในจังหวัดสมุทรปราการโดยใช้ Dijkstra's algorithm ขึ้นเพื่ออำนวยความสะดวก และเพิ่มประสิทธิภาพในการวางแผนการเดินทางด้วยรถโดยสารสาธารณะโดยนำDijkstra's algorithm มาช่วยในการคำนวณหาวิธีในการเดินทาง โดยให้ผู้ใช้กำหนดพิกัดตำแหน่งของจุดต้นทางและจุดปลายทาง [11]

3 ขั้นตอนการดำเนินงาน

ตารางที่ 1 การเปรียบเทียบอัลกอริทึมในการค้นหาเส้นทาง

อัลกอริทึม	ลักษณะงานที่ใช้	เวลาในการทำงาน
Kruskal Algorithm	การเดินทางที่ต้องผ่านทุกจุด	O(|E| lg |V|)
Prim Algorithm	การเดินทางที่ต้องผ่านทุกจุด	O(|E|+|V| lg |V|)
Bellman – Ford Algorithm	การค้นหาเส้นทางเดินรถ มีค่าน้ำหนักเป็นลบเข้ามาเกี่ยวข้อง	O(|V||E|)
Dijkstra's Algorithm	การค้นหาเส้นทางเดินรถ ไม่มีค่าน้ำหนักเป็นลบเข้ามาเกี่ยวข้อง	O(|V|)

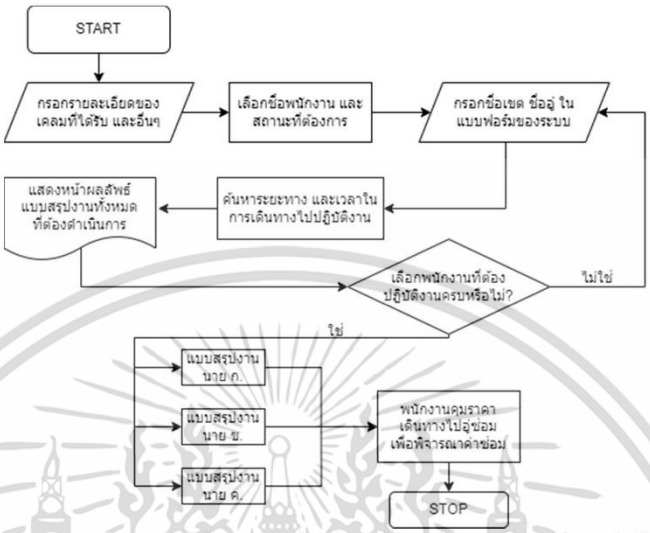
งานวิจัยนี้เป็นการพัฒนาระบบในการจัดเส้นทางเพื่อการเดินทางไปปฏิบัติงานโดยใช้โปรแกรม Microsoft excel และ VBA function ซึ่งจากการเปรียบเทียบอัลกอริทึมดังตารางที่ 1 ในการค้นหาเส้นทางแบบต่าง ๆ พบว่า Kruskal Algorithm ใช้เวลาในการทำงาน O(|E| lg |V|), Prim Algorithm ใช้เวลาในการทำงาน O(|E|+|V| lg |V|), Bellman – Ford Algorithm ใช้เวลาในการทำงาน O(|V||E|) และ Dijkstra's Algorithm ใช้เวลาในการทำงาน O(|V|)
โดย |V| คือ จำนวนจุดยอด, |E| คือ จำนวนเส้นขอบของกราฟ, สัญลักษณ์ V หมายถึง |V| และสัญลักษณ์ E หมายถึง |E| ตัวอย่างเช่น "อัลกอริทึมทำงานในเวลา O(V E)" หมายความว่าอัลกอริทึมทำงานในเวลา O(|V||E|) ซึ่งจะทำให้สูตรเวลาของการทำงาน (running-time) อ่านง่ายขึ้นโดยไม่ทำให้เกิดความเสี่ยงที่จะเกิดความคลุมเครือของสูตร [12] ดังนั้นจึงเลือกใช้วิธี Dijkstra's Algorithm ในการค้นหาเส้นทางที่เหมาะสม เนื่องจากวิธีนี้ใช้เวลาในการทำงานน้อยที่สุด โดยมีวิธีการดำเนินงานวิจัยดังนี้

3.1 ศึกษาและออกแบบระบบจัดการงานสินไหมความเสียหายหนัก

จากการสอบถามและเก็บข้อมูลเบื้องต้นทำให้ทราบถึงปัญหาที่เกิดขึ้นว่า สินไหมกรณีความเสียหายหนักไม่ได้มีการเก็บรวบรวมข้อมูลในส่วนที่จำเป็นเช่น วันที่ได้รับงาน กระทั่งวันที่ปฏิบัติงานแล้วเสร็จ จึงทำให้การติดตามงานหรือวัดผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดำเนินงานเป็นไปได้ยาก ผู้วิจัยจึงได้รวบรวมข้อมูลที่จะใช้ในการเก็บ จัดแยกประเภท สถานะ และออกแบบรูปแบบการเก็บฐานข้อมูล เมื่อกรอกข้อมูลข้างต้นข้อมูลจะถูกนำเข้ากระบวนการของระบบตามกระบวนการดังรูปที่ 1



รูปที่ 1 กระบวนการงานระบบการจัดการงานสินไหมในกรณีความเสียหายหนักของการประกันภัยรถยนต์

3.2 สร้างสมการวัตถุประสงค์ และเงื่อนไขที่ใช้ในระบบ

เนื่องจากระบบจัดการงานสินไหมความเสียหายหนัก จำเป็นต้องทราบถึงเวลาทั้งหมดโดยประมาณที่พนักงานคุมราคาใช้ในการปฏิบัติงาน ซึ่งรวมทั้งเวลาเดินทาง และเวลาขณะปฏิบัติงานอยู่ ณ อู่ซ่อมรถยนต์ ผู้วิจัยจึงได้สร้างขั้นตอนการหาเวลาทั้งหมดที่กล่าวมาโดยมีฟังก์ชันวัตถุประสงค์ (Objective Function) ดังสมการที่ 1

$$F(T) = T_S + T_O \tag{1}$$

เมื่อ $F(T)$ คือ เวลาทั้งหมดโดยประมาณในการปฏิบัติงาน , T_S คือ เวลาเดินทาง(คำนวณจากระบบ) และ T_O คือ เวลาปฏิบัติงานทั้งหมดตามจำนวนงานที่ได้รับ โดยเวลาเดินทางจากระบบ คำนวณได้ดังสมการที่ 2

$$T_S = (\frac{D_S(\text{km})}{80(\text{km/hr})}) \times 60(\text{min}) \tag{2}$$

ซึ่งใช้อัตราความเร็วที่ 80 (km/hr.) ตามมาตรา ๕ และมาตรา ๖๗ วรรคหนึ่ง แห่งพระราชบัญญัติจราจรทางบก พ.ศ. ๒๕๒๒ รัฐมนตรีว่าการกระทรวงมหาดไทยออกกฎกระทรวงไว้ดังนี้ ข้อ ๑ ในกรณีปกติ ให้กำหนดความเร็วสำหรับรถดังต่อไปนี้ (๓) สำหรับรถยนต์อื่นนอกจากรถที่ระบุไว้ใน (๑) หรือ (๒) หรือรถจักรยานยนต์ ให้ขับในเขตกรุงเทพมหานคร เขตเมืองพัทยา หรือเขตเทศบาล ไม่เกินชั่วโมงละ ๘๐ กิโลเมตร [13]และเวลาปฏิบัติงานทั้งหมดตามจำนวนงานที่ได้รับ โดยผู้วิจัยกำหนดให้มีเวลาในการปฏิบัติงานละ 60 นาที คำนวณได้จากสมการที่ 3

$$T_O = b \times 60(\text{min}) \tag{3}$$

และขั้นตอนการหาผลรวมของระยะทางที่ได้สามารถคำนวณได้จากอัลกอริทึม 3.3.2

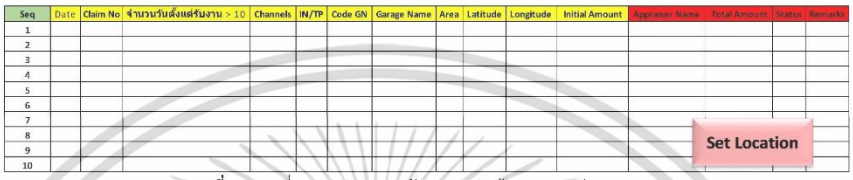
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการออกแบบระบบจัดการงานสินไหมความเสียหายหนัก

3.3.1 สร้างแบบฟอร์มสำหรับระบบงานสินไหมความเสียหายหนัก

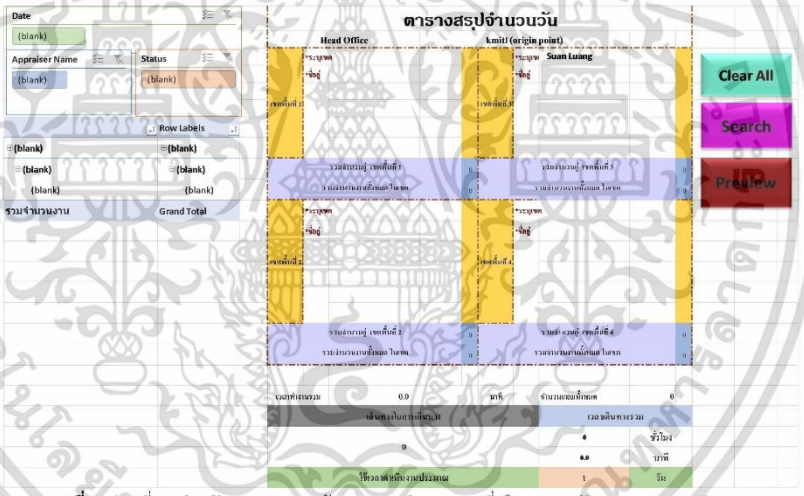
แบบฟอร์มจะถูกแบ่งออกเป็น 3 ส่วนดังนี้

ส่วนที่ 1: สำหรับกรอกข้อมูลที่เกี่ยวข้องกับงานสินไหมความเสียหายหนัก เช่น วันที่รับงาน รหัสเคลม ช่องทางที่รับงานมา รหัสอู่ ชื่ออู่ และพนักงานคุมราคาที่รับผิดชอบงาน ซึ่งส่วนนี้จะมีการใช้ Macro ที่เขียนคำสั่งใน VBA เพื่อดึงข้อมูลมากรอกโดยอัตโนมัติ (ข้อมูลเขตของอู่ และข้อมูลพิกัด Latitude, Longitude) และใช้ Data Validation เพื่อให้ง่ายต่อการกรอกข้อมูล โดยส่วนที่ 1 แสดงในรูปที่ 2



รูปที่ 2 ส่วนที่ 1:สำหรับกรอกข้อมูลงานสินไหมความเสียหายหนัก

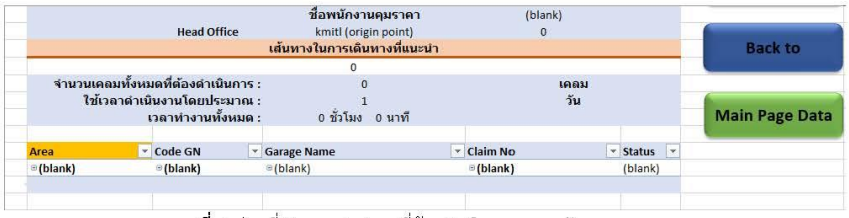
ส่วนที่ 2: สำหรับประมวลผลเส้นทางตามจำนวนงานที่เลือกของพนักงานคุมราคาแต่ละคน ซึ่งการเลือกงานนั้นจะใช้ PivotTable เป็นเครื่องมือกรองข้อมูลจากส่วนที่ 1 จากนั้นนำข้อมูลที่เลือกไปประมวลผลในอัลกอริทึมเพื่อหาเส้นทางและคำนวณเวลาตามสมการวัตถุประสงค์ โดยส่วนที่ 2 แสดงดังรูปที่ 3



รูปที่ 3 ส่วนที่ 2: สำหรับประมวลผลเส้นทางตามจำนวนงานที่เลือกของพนักงานคุมราคาแต่ละคน

ส่วนที่ 3: เป็นแบบสรุปงานที่ต้องดำเนินการของพนักงานคุมราคาที่เลือกในส่วนที่ 2 โดยส่วนนี้ใช้ PivotTable ดึงข้อมูลในส่วนที่ 2 มาแสดง ซึ่งส่วนนี้จะส่งให้กับพนักงานคุมราคา เพื่อให้ดำเนินการตามแบบสรุป ดังแสดงในรูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 ส่วนที่ 3: แบบสรุปงานที่ต้องดำเนินการของพนักงานคุมราคา

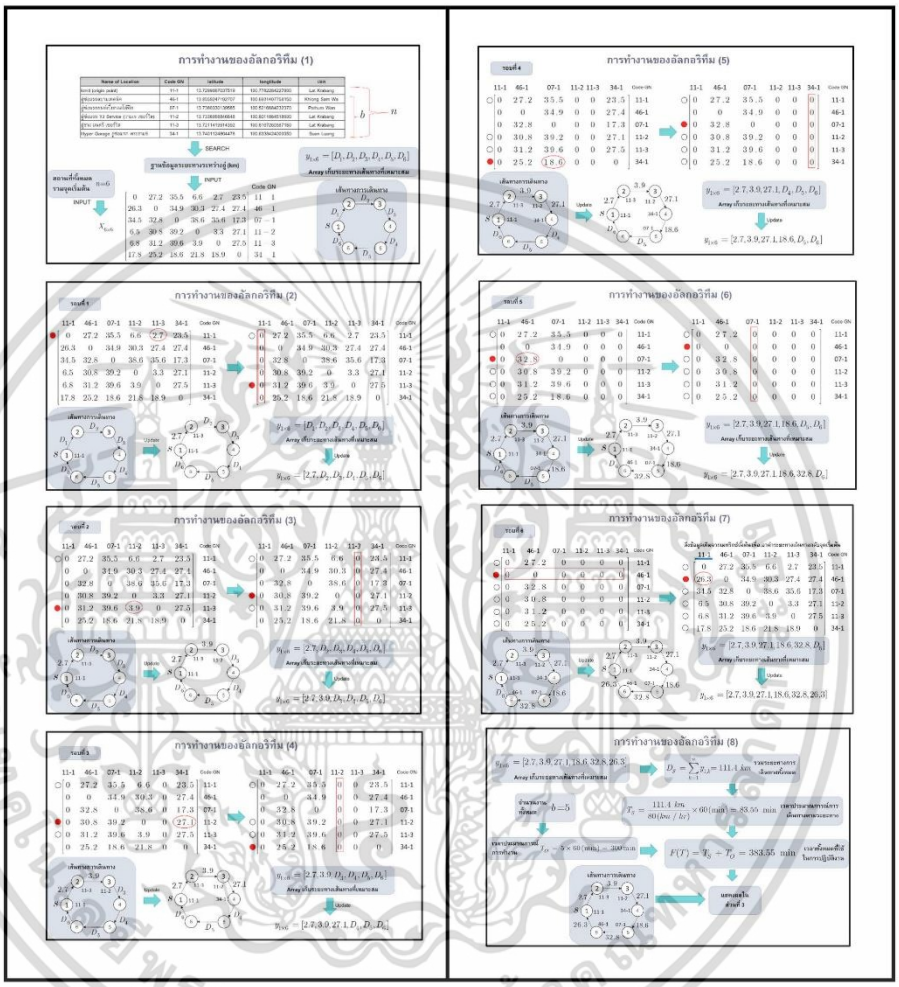
3.3.2 พัฒนาอัลกอริทึมสำหรับหาเส้นทางที่เหมาะสม

อัลกอริทึมสำหรับหาเส้นทางที่เหมาะสม เขียนอยู่บนพื้นฐานของ Dijkstra's Algorithm โดยใช้ VBA ในการเขียนอัลกอริทึม โดยแสดงขั้นตอนแบบย่อดังนี้

1. $X_{n\times n} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n} \\ \vdots & \ddots & \vdots \\ d_{n,1} & \cdots & d_{n,n} \end{bmatrix}$ //เมทริกซ์เก็บค่าระยะทาง (เส้นเชื่อม: Edge) ขนาด $(n \times n)$
2. $y_{1\times k} = [D_1, D_2, \ldots\ldots, D_n]$ //Array สำหรับเก็บค่าระยะทางที่สั้นที่สุดจากจุดหนึ่ง (Node) ไปยังจุดใดๆ
3. $c = 1$ //ตัวแปรสำหรับกำหนดจุดเริ่มต้น
4. for (i = 1 to b) //แถว
5. for (j = 1 to b) //หลัก
6. if ($X_{c,j}$ น้อยกว่า $X_{c,j+1}$ และ $X_{c,j} \neq 0$) //โดย $X_{c,j} = 0$ เมื่ออยู่อยู่ปัจจุบัน และเป็นอู่ที่เดินทางไปแล้ว
7. D_i = $X_{c,j+1}$ //เก็บค่าระยะทางสั้นที่สุด
8. $c = j$ //ตัวเลขจุด(อู่)ถัดไปที่มีระยะทางสั้นที่สุด
9. End if
10. Next j
11. for (k = 1 to n) //ให้หลักในเมทริกซ์ของจุดที่เดินทางไปแล้วเป็น 0 ทั้งหมด
12. $X_{k,c}$ = 0
13. Next k
14. Next i
15. D_n = ระยะทางจากจุดสุดท้าย กลับไปยังจุดเริ่ม
16. $D_S = \sum_{k=1}^{n} y_{1,k}$ //รวมระยะทางทั้งหมดของเส้นทางที่ได้
17. $T_S = (\frac{D_S(\text{km})}{80(\text{km/hr})}) \times 60(\text{min})$ //คำนวณเวลาในการเดินทางทั้งหมด
18. $T_O = b \times 60(\text{min})$ //คำนวณเวลาในการปฏิบัติงานตามจำนวนงาน
19. $F(T) = T_S + T_O$ //เวลาทั้งหมดที่ใช้ในการปฏิบัติงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดให้ $x_{c,j}$ คือ ค่าที่น้อยที่สุดและไม่เป็นศูนย์ของสมาชิกในแต่ละแถวของเมทริกซ์ $X_{n\times n}$, D_S คือ ระยะทางรวม , $y_{1\times k}$ คือ เมทริกซ์แถวสำหรับเก็บค่าระยะทางน้อยที่สุดของจุดถัดไปโดยที่ไม่ซ้ำจุดเดิม, b คือ จำนวนงานสินไหม, $X_{n\times n}$ คือ เมทริกซ์สำหรับเก็บค่าระยะทางจากจุดหนึ่งไปยังจุดหนึ่ง โดยที่ $n=b+1$ นั่นคือจำนวนมิติเกิดจากการนำจำนวนงานสินไหมรวมกับจุดเริ่มต้น และกระบวนการทำงานของอัลกอริทึมจะแสดงได้ดังรูปที่ 5



รูปที่ 5 การทำงานของอัลกอริทึม

คือเมื่อเลือกจำนวนงานสินไหมที่ต้องการดำเนินการแล้ว จำนวนสถานที่ทั้งหมดที่เลือกรวมถึงจุดเริ่มต้นแทนด้วย n จากนั้นระบบจะสร้างเมทริกซ์เก็บค่าระยะทางที่ตรงกับข้อมูลที่เลือก เพื่อเตรียมข้อมูลนี้เข้าสู่อัลกอริทึม เมื่อเข้าสู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึม เมทริกซ์แถวที่1ซึ่งเป็นจุดเริ่มต้นจะเข้าสู่for loop เพื่อหาระยะทางที่มีค่าน้อยสุดและไม่มีค่าเป็นศูนย์ ซึ่งในรอบที่1 จะเลือกไปอู่ที่รหัส 11-3 ถัดมาที่อยู่ปัจจุบันจะถูกเปลี่ยนเป็นสีแดงและรหัสที่เดินทางไปแล้วจะแสดงเป็นจุดสีขาวพร้อมทั้งเปลี่ยนให้มีค่าเป็นศูนย์ทั้งคอลัมน์ จากนั้นเส้นทางที่เลือกจะถูกบันทึกและค่าระยะทางที่เลือกจะถูกนำไปเก็บในอาร์เรย์ที่ได้สร้างไว้ โดยจะทำตามกระบวนการเหมือนดังรอบที่ 1 ไปเรื่อยๆ จนเมื่อเลือกอู่จนครบทุกอู่แล้วจะไม่สามารถเดินทางไปยังอู่อื่นได้อีก เนื่องจากทั้งแถวในเมทริกซ์มีค่าเป็นศูนย์ ดังนั้นระบบจะดึงข้อมูลจากเมทริกซ์ตั้งต้นมาเพื่อเอาค่าระยะทางสำหรับเดินทางกลับไปยังจุดเริ่มต้น และสุดท้ายเมื่อได้เส้นทางที่เหมาะสมระบบจะนำค่าระยะทางที่เก็บในอาร์เรย์มาประมวลผลหาระยะทางทั้งหมด รวมถึงนำไปหาระยะเวลาในการเดินทางโดยประมาณ จากนั้นทำการประมาณเวลาในการทำงานจากจำนวนงานทั้งหมด ซึ่งหลังการประมวลผลนี้ระบบจะนำค่าที่ได้และเส้นทางการเดินทางที่แนะนำไปแสดงในส่วนที่ 3

4 การทดสอบระบบและผลการดำเนินงาน

4.1 การทดสอบระบบ

เมื่อระบบการจัดการงานสินไหมความเสียหายหนักถูกพัฒนาบนโปรแกรม Microsoft Excel โดยอาศัยข้อมูลจากการเก็บตัวอย่าง อู่ซ่อมรถยนต์ในพื้นที่ จังหวัดกรุงเทพมหานครจำนวน 20 อู่ ตามรูปที่ 6 และ รูปที่ 7 จากนั้นทำการกรอกข้อมูลใน ส่วนที่ 1 ตามรูปที่ 8 เมื่อเสร็จแล้วให้ไปยังส่วนที่ 2 เพื่อเลือกจำนวนงานของพนักงานคุมราคา จะได้ดังรูปที่ 9 และเมื่อทำการประมวลผลเพื่อหาเส้นทางแล้วให้ไปยังส่วนที่ 3 จะสรุปงานที่ต้องดำเนินการตามรูปที่ 10

	kmitl	ปูเปิ้ลยนต์	ช่อมรถ	ปูเปอร์	ปูเปา ช่าง Th	Hyper	ปูพัฒนา	ปูสมศักดิ์	ปูเพชร	ปูเทอร์	ปูเทอร์	ปูสมชาย	ปูเทอร์	ปูเปอร์	ปูเทอร์	ปูเทพ	ปูเปา
kmitl (origin point)	0.00	34.50	31.40	6.80	6.50	4.50	17.80	11.10	7.90	12.90	21.70	15.50	24.20	8.00	18.60	28.30	22.50	20.50	26.30	22.40	11.10
ปั๊มแก๊สแบน ปั๊มกลอไลฟ์	35.90	0.00	24.00	39.60	39.20	35.70	18.60	34.70	33.00	38.60	43.10	24.80	19.70	32.90	35.60	18.50	16.10	16.00	34.90	37.50	29.80
ปั๊มแก๊สแบน อัลฟาร์ด ลาดปลาเค้า 44	36.90	18.30	0.00	40.90	40.50	37.00	19.70	24.70	34.30	25.10	22.80	26.30	30.20	34.20	17.50	12.10	18.10	25.10	17.10	12.70	26.70
ปั๊มแก๊ส Yi Service (วายเจ เซอร์วิส)	2.70	35.60	32.50	0.00	3.20	6.30	18.90	11.40	9.00	15.30	21.20	16.60	25.30	9.00	18.90	29.40	23.60	21.60	27.40	23.40	12.70
ปั๊มช่าง มนตรี เซอร์วิส	6.60	38.60	35.40	3.30	0.00	5.90	21.80	13.70	12.90	15.50	24.40	19.50	28.20	12.30	21.20	32.30	26.50	24.80	30.30	26.40	16.00
ปั๊ TNN อกาศระบ (T N N Garage Co.,ltd.)"	4.60	35.10	31.90	6.30	5.90	0.00	18.30	8.20	11.30	10.00	18.80	16.10	24.75	8.60	15.70	22.70	23.10	21.10	26.80	22.90	14.40
Hyper Garage ปั๊มช่อมรถ พระรามอ	23.50	17.30	19.70	27.50	27.10	23.60	0.00	22.60	20.90	26.50	35.10	11.10	13.00	20.80	23.50	7.10	2.80	10.30	27.40	25.40	17.20
ปั๊พัฒนาก้าจ้ยนต์ก์ เตาะพันธ์ก	9.90	34.60	31.40	12.50	12.20	8.00	17.80	0.00	7.40	8.50	18.40	15.50	22.00	6.30	12.30	19.20	22.50	20.60	26.30	22.40	9.40
ปั๊มสมศักดิ์ การช่าง	9.10	32.70	29.10	8.10	7.60	7.20	15.40	8.00	0.00	11.90	17.80	10.70	21.80	3.20	15.50	25.90	20.20	15.80	23.90	20.00	6.40
ปั๊มแก๊สแบน - แบนช่อมแก๊สอ่อไลฟ์เซอร์วิส	11.70	36.40	33.20	14.40	14.00	9.80	19.60	2.00	9.20	0.00	13.10	17.30	23.80	8.10	9.50	21.00	24.30	22.40	17.70	24.20	11.20
ปั๊พัฒนาก้าจ้ยนต์ก์ เตาะพันธ์ก	17.50	43.20	24.90	20.10	19.80	15.60	26.40	10.90	18.20	11.20	0.00	24.00	32.80	17.00	8.90	25.30	31.70	29.90	16.50	15.00	20.10
ปั๊มแก๊สแบ ปั๊มพัฒนช์	14.30	24.30	24.30	20.50	20.10	16.60	7.40	15.60	11.70	19.50	23.90	0.00	8.00	13.80	16.40	12.30	8.60	5.70	22.20	18.30	6.80
ปั๊มแก๊สร้อยล้อยางรีการครีนเดินทร์ธร์	26.20	19.50	10.50	30.30	26.80	26.40	9.10	25.40	23.80	29.40	33.80	8.00	0.00	18.50	26.20	22.00	11.10	15.10	32.10	28.10	17.30
ปั๊สุวรรณกปั๊มช่อมรีส	6.50	32.70	29.50	12.50	10.90	8.10	15.90	7.50	3.90	11.40	17.30	7.70	13.70	0.00	15.00	26.40	20.60	12.70	24.40	20.50	3.30
ปั๊มน่อย ปั๊เปอ	17.40	38.20	21.00	20.00	19.60	15.40	21.50	7.60	14.80	7.90	7.70	19.00	27.80	13.70	0.00	17.70	26.20	24.00	13.50	13.80	17.30
ปั๊มแก๊กปั๊ aodautoshop	27.60	17.80	11.60	31.60	31.20	27.70	10.40	18.70	25.00	24.60	27.70	16.90	20.50	24.80	22.30	0.00	8.80	15.90	20.00	17.50	21.80
ปั๊นาการาจ ปั๊มแก๊รถยนต์(Ninja Garage Pattanakarn)	21.90	16.70	17.10	26.00	25.60	22.10	2.90	21.10	19.40	25.00	33.00	9.60	11.40	18.30	21.90	8.80	0.00	8.70	25.20	23.90	16.20
ปั๊มช่อมรถ Honda HKP SPEED	23.20	14.60	24.30	27.70	25.80	25.30	6.60	22.30	20.60	26.20	30.70	6.40	4.90	20.50	23.20	12.30	8.60	0.00	29.00	25.10	10.90
ปั๊มแก๊สยางแท้ปั๊มมี	27.20	32.90	13.60	31.20	30.80	27.30	25.20	16.70	24.60	17.00	14.90	22.70	31.60	24.50	9.40	17.80	23.80	27.80	0.00	8.00	21.10
ปั๊นพรัตนวิศ ปั๊มช่อมรถปั๊มโต	23.30	28.30	9.10	27.30	26.90	23.40	21.30	22.40	20.70	17.30	14.90	18.90	27.70	20.60	9.40	13.30	19.30	23.90	6.30	0.00	17.10
ปั๊ช่างพร้อม ปั๊มช่อมรถยนต์	8.00	29.30	26.60	14.90	12.50	10.60	12.60	9.00	5.50	12.60	18.80	6.90	12.80	4.80	16.50	23.10	17.30	11.90	21.50	17.60	0.00

รูปที่ 6 ข้อมูลระยะทางจากอู่หนึ่งไปยังอู่ต่าง ๆ

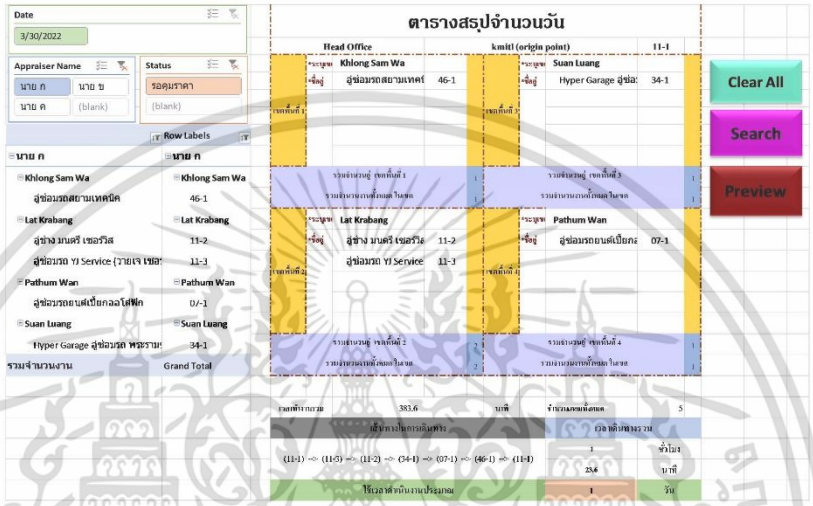
Name of Location	Code	latitude	longtitude	เขต
kmitl (origin point)	11-1	13.729087037519	100.778228427000	Lat Krabang
ปั๊มแก๊สแบน ปั๊มกลอไลฟ์	07-1	13.738031701139	100.527166842327	Pathum Wan
ปั๊มแก๊สแบน อัลฟาร์ด ลาดปลาเค้า 44	38-1	13.844130038220	100.600625469276	Bang Phlat
ปั๊มแก๊ส Yi Service (วายเจ เซอร์วิส)	11-2	13.736058846648	100.801186518500	Lat Krabang
ปั๊มช่าง มนตรี เซอร์วิส	11-3	13.721141991439	100.816726038718	Lat Krabang
ปั๊ TNN อกาศระบ (T N N Garage Co.,ltd.)"	11-4	13.737816862784	100.775430069631	Lat Krabang
Hyper Garage ปั๊มช่อมรถ พระรามอ	38-1	13.747112490447	100.633842400035	Suan Luang
ปั๊พัฒนาก้าจ้ยนต์ก์ เตาะพันธ์ก	11-5	13.708805901106	100.746799593621	Lat Krabang
ปั๊มสมศักดิ์ การช่าง	11-6	13.721430132161	100.766475181757	Lat Krabang
ปั๊มแก๊สแบน - แบนช่อมแก๊สอ่อไลฟ์เซอร์วิส	11-7	13.754331860731	100.757061490978	Lat Krabang
ปั๊พัฒนาก้าจ้ยนต์ก์ เตาะพันธ์ก	10-1	13.819537583968	100.777540800776	Min Buri
ปั๊มแก๊กปั๊ aodautoshop	32-1	13.716280079331	100.669653105906	Prawet
ปั๊มแก๊สร้อยล้อยางรีการครีนเดินทร์ธร์	32-2	13.674276925730	100.647294490842	Prawet
ปั๊สุวรรณกปั๊มช่อมรีส	11-8	13.721521633515	100.738054229668	Lat Krabang
ปั๊มน่อย ปั๊เปอ	10-2	13.802052500094	100.718769350988	Min Buri
ปั๊มช่อมรถ Honda HKP SPEED	45-1	13.708957266729	100.512570237912	Wang Thonglang
ปั๊นาการาจ ปั๊มแก๊รถยนต์(Ninja Garage Pattanakarn)	34-2	13.741467683476	100.611473823854	Suan Luang
ปั๊มช่อมรถ Honda HKP SPEED	34-3	13.708973103258	100.631020747982	Suan Luang
ปั๊มแก๊สยางแท้ปั๊มมี	46-1	13.869024710270	100.691140775815	Khlong Sam Wa
ปั๊นพรัตนวิศ ปั๊มช่อมรถปั๊มโต	43-1	13.836666523901	100.665431428079	Khan Na Yao
ปั๊ช่างพร้อม ปั๊มช่อมรถยนต์	11-9	13.724208912439	100.718339715914	Lat Krabang

รูปที่ 7 ข้อมูลที่อยู่ของอู่ซ่อมรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Seq	Date	Claim No	จำนวนวันตั้งแต่ซ่อมรถ > 30	Channels	IN/TP	Code GN	Garage Name	Area	Latitude	Longitude	Initial Amount	Approve Name	Total Amount	Status	Remarks
1	30/03/2022	00-AV1-2018-1000013508	0	Counter	INSURER	07-1	อู่ซ่อมรถแผงโปรม ลอโพ้ศิก	Pathum Wan	13.738018913585	100.527668232379	120,000.00	นาย ก	วัน	รอดูเคราะห์	
2	30/03/2022	00-AV1-2018-1000006249	0	Counter	TP	11-2	ช่าง มนตรี เซอร์วิส	Lat Krabang	13.7231328951982	100.8562784381286	12,400.00	นาย ข	วัน	รอดูเคราะห์	
3	30/03/2022	45-AV1-2018-0700000620	0	Counter	TP	34-1	Hyper Garage อู่ซ่อมรถ พระราม9	Suan Luang	13.7491124804478	100.6738424000350	250,000.00	นาย ก	วัน	รอดูเคราะห์	
4	30/03/2022	00-AV1-2018-0000113169	0	บคอมพ์	INSURER	32-1	อู่ซ่อมรถเจ้อุ่งสุขสด	Prawet	13.7323800.988310	100.6696203355.5660	178,000.00	นาย ค	วัน	รอดูเคราะห์	
5	30/03/2022	00-AV1-2018-1500015573	0	บคอมพ์	INSURER	46-1	อู่ซ่อมรถสยามเทคนิค	Khlong Sam Wa	13.8550147197747	100.6651457758150	298,500.00	นาย ก	วัน	รอดูเคราะห์	
6	30/03/2022	00-AV1-2018-1700015591	0	Counter	INSURER	11-5	อู่ช่างหรอย ซ่อมรถยนต์	1st Krabang	13.7243481524757	100.7183757275148	164,000.00	นาย ข	วัน	รอดูเคราะห์	
7	30/03/2022	00-AV1-2018-0300005914	0	ห้างราคาไม่ทราบชื่อ	TP	34-2	นิงการจาง (อู่ซ่อมรถยนต์[Ninja Garage Pattanakarn)	Suan Luang	13.7419674.989406	100.6114736238540	130,000.00	นาย ข	วัน	รอดูเคราะห์	
8	30/03/2022	00-AV1-2018-0900017490	0	Counter	TP	11-2	อู่ซ่อมรถ YJ Service (รายเจ เซอร์วิส)	1st Krabang	13.7264749856.98	100.8913867528549	221,000.00	นาย ก	วัน	รอดูเคราะห์	
9	30/03/2022	00-AV1-2018-0000000445	0	บคอมพ์	INSURER	34-3	อู่ซ่อมรถ Honda HKP SPEED	Suan Luang	13.7009731.032584	100.6710020426826	195,000.00	นาย ค	วัน	รอดูเคราะห์	
10	30/03/2022	66-AV1-2018-0000009415	0	Counter	INSURER	32-2	อู่ซ่อมรถ ช็อตรถยนต์ปริก ราชพัฒนาศึกษาพิทักษ์ไทร์	Prawet	13.6742.9.6625.7.3900	100.647.2364866426	241,000.00	นาย ข	วัน	รอดูเคราะห์	
11	30/03/2022	77-AV1-2018-0000004918	0	ห้างราคาไม่ทราบชื่อ	TP	10-1	อู่พัฒนารัก อินสุลี มลายาสปีซ	Min Buri	13.8105375809681	100.7775.4.8900.7769	145,000.00	นาย ข	วัน	รอดูเคราะห์	
12															
13															
14															
15															
16														Set Location	
17															
18															

รูปที่ 8 ข้อมูลงานสินไหมความเสียหายหนัก



รูปที่ 9 ประมวลผลเส้นทางตามจำนวนงานที่เลือก ของแต่ละพนักงานคุมราคา

Distance Matrix (km)

	1	2	3	4	5	6	
	kmitl (origin poin	อู่ซ่อมรถสยามเทค	อู่ช่าง มนตรี เซอร์	อู่ซ่อมรถ YJ Servic	Hyper Garage อู่ซ	อู่ซ่อมรถยนต์เบี้ยก	Code GN
1 kmitl (origin point)	0	27.2	6.6	2.7	23.5	35.5	11-1
2 อู่ซ่อมรถสยามเทคนิค	26.3	0	30.3	27.4	27.4	34.9	46-1
3 อู่ช่าง มนตรี เซอร์วิส	6.5	30.8	0	3.3	27.1	39.2	11-2
4 อู่ซ่อมรถ YJ Service (รายเจ เซอร์วิส)	6.8	31.2	3.9	0	27.5	39.6	11-3
5 Hyper Garage อู่ซ่อมรถ พระราม9	17.8	25.2	21.8	18.9	0	18.6	34-1
6 อู่ซ่อมรถยนต์เบี้ยกลอโต้พัก	34.5	32.8	38.6	35.6	17.3	0	07-1

เส้นทางที่ได้	(11-1) --> (11-3) --> (11-2) --> (34-1) --> (07-1) --> (46-1) --> (11-1)
ระยะทางรวม (km)	111.4

รูปที่ 10 ผลลัพธ์ที่ได้จากอัลกอริทึม

4.2 อภิปรายผลการดำเนินงาน

จากกลุ่มตัวอย่างที่นำมาทดสอบผ่านระบบการจัดการงานสินไหมความเสียหายหนัก โดยนำมาทดสอบจำนวน 6 โหนด (มีอู่จำนวน 5 อู่ และมี 1 จุดเริ่มต้น) คำนวณเวลาในการปฏิบัติงานตามจำนวนงาน (T_O) จะได้ 300 นาที จากนั้นทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลด้วยอัลกอริทึมเขียนอยู่บนพื้นฐานของ Dijkstra's Algorithm จะได้เส้นทางการเดินทาง และระยะทางทั้งหมดของเส้นทาง (D_S) ดังรูปที่ 10 จากนั้นนำระยะทางทั้งหมดของเส้นทางมาคำนวณเวลาการเดินทาง (T_S) ตามสมการที่ 2 ได้ 83.55 นาที แล้วหาเวลาทั้งหมดโดยประมาณในการปฏิบัติงาน $F(T)$ = 383.55 นาที หรือ 6 ชั่วโมง 23.55 นาที ซึ่งมีความเป็นไปได้ที่สามารถปฏิบัติงานเสร็จได้ในหนึ่งวัน

เมื่อดูถึงความเหมาะสมของผลลัพธ์ที่ระบบให้มาแล้ว สามารถจัดทำแบบสรุปงานทั้งหมดที่ต้องดำเนินการตามรูปที่ 11 ซึ่งมีการระบุรายละเอียดต่างๆ ที่พนักงานคุมราคาควรทราบได้แก่ วันที่ออกใบรายการ ชื่อพนักงานคุมราคา ผลสรุปเส้นทางในการเดินทางที่ระบบแนะนำให้พนักงานคุมราคาเดินทางไปปฏิบัติงาน จำนวนงานสินไหมทั้งหมดที่ต้องดำเนินการ รวมไปถึงการประมาณการเวลาในการดำเนินงานเพื่อเดินทางไปปฏิบัติงานดังกล่าว

แบบสรุปงานทั้งหมดที่ต้องดำเนินการ

Print | Back to | Main Page Data

	วันที่ออกใบรายการ	Friday, April 29, 2022
	ชื่อพนักงานคุมราคา	นาย ก
Head Office	kmitl (origin point)	11-1

เส้นทางในการเดินทางที่แนะนำ

(11-1) --> (11-3) --> (11-2) --> (34-1) --> (07-1) --> (46-1) --> (11-1)

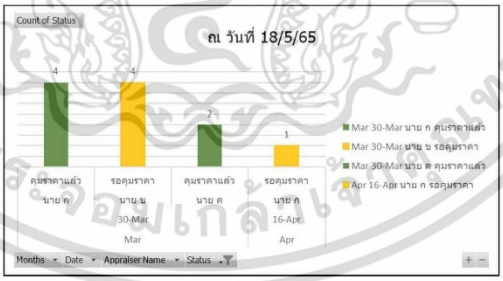
จำนวนเคลมทั้งหมดที่ต้องดำเนินการ :	5	เคลม
ใช้เวลาดำเนินงานโดยประมาณ :	1	วัน
เวลาทำงานทั้งหมด :	6 ชั่วโมง 23.55 นาที	

Area	Code GN	Garage Name	Claim No	Status
Pathum Wan	07-1	อู่ซ่อมรถยนต์เปี๊ยกออโต้ฟิก	00-AV1-2018-330001358	รอคุมราคา
Lat Krabang	11-2	อู่ซ่อมรถ YJ Service (วายเจ เซอร์วิส)	00-AV3-2018-080003780	รอคุมราคา
	11-3	อู่ช่าง มนตรี เซอร์วิส	00-AV1-2018-470000694	รอคุมราคา
Suan Luang	34-1	Hyper Garage อู่ซ่อมรถ พระราม9	45-AV1-2018-470000620	รอคุมราคา
Khlong Sam Wa	46-1	อู่ซ่อมรถสยามเทคนิค	00-AV1-2018-150001573	รอคุมราคา

รูปที่ 11 แบบสรุปงานที่ต้องดำเนินการของพนักงานคุมราคา

5 บทสรุป

จากผลการทดสอบระบบแสดงให้เห็นว่าเมื่อมีการนำระบบการจัดการงานสินไหมความเสียหายหนักไปใช้จะทำให้เห็นภาพการแบ่งงานที่เหมาะสมมากขึ้น รวมไปถึงทำให้บริษัทฯ ทราบปริมาณงานสินไหมความเสียหายหนักที่เข้ามาใหม่ในแต่ละวันงานที่พิจารณาแล้วเสร็จ และการติดตามงานคงค้างได้จากสถานะ หรือหาจากสาเหตุได้อย่างชัดเจน เช่น รอพนักงานฝ่ายราคาจัดการงานในระบบ หรือรอให้พนักงานคุมราคาออกไปปฏิบัติงาน เป็นต้น ซึ่งจะช่วยให้บริษัทฯ สามารถบริหารจัดการงานได้จากปริมาณงาน และสถานะงานสินไหมได้อย่างถูกต้อง ตัวอย่างเช่น การตัดสินใจเพิ่มคนหรือวางแผนการเร่งพิจารณาสินไหม โดยสามารถนำข้อมูลไปวิเคราะห์เพิ่มเติมได้ดังรูปที่ 12



รูปที่ 12 ตัวอย่างการนำข้อมูลไปใช้ประโยชน์ในการวิเคราะห์ผลลัพธ์ด้านอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] สำนักงานคณะกรรมการกำกับและส่งเสริมการประกอบธุรกิจประกันภัย (คปภ.). (2565). **รายงานภาวะธุรกิจประกันภัยไทย ประจำไตรมาส 1 ปี 2564**. [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://www.oic.or.th/th/consumer/education-resources/article.

[2] LMG Insurance Public Company Limited. (2017). **ขั้นตอนการเคลมประกันภัย**. [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://www.lmginsurance.co.th/th/Claims/Pages/involve-other-party.aspx#.

[3] สำนักงานคณะกรรมการกำกับและส่งเสริมการประกอบธุรกิจประกันภัย (คปภ.). (2565). **ความหมายของการประกันภัย**. [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://www.oic.or.th/th/education/insurance/about/meaning.

[4] Bank of Thailand. (2014). **คำศัพท์น่ารู้ (ประกันวินาศภัย)**. [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://www.1213.or.th/th/others/insurances/Pages/glossary.aspx.

[5] บริษัท วิริยะประกันภัย จำกัด(มหาชน). (2557). **ซ่อมห้าง vs ซ่อมอู่ แบบไหนดีกว่ากัน ?** [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://www.viriyah.com/article/detail/28-ซ่อมห้าง-vs-ซ่อมอู่-แบบไหนดีกว่ากัน.

[6] Wang, L. T., Chang, Y. W., & Cheng, K. T. T. (Eds.). (2009). **Electronic Design Automation: Synthesis, Verification, and Test.** Morgan Kaufmann.

[7] Microsoft Support. (2022). **Look up values with VLOOKUP, INDEX, or MATCH.** [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://support.microsoft.com/en-us/office/look-up-values-with-vlookup-index-or-match-68297403-7c3c-4150-9e3c-4d348188976b.

[8] Microsoft Support. (2022). **Create a drop-down list.** [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. https://support.microsoft.com/en-us/office/create-a-drop-down-list-7693307a-59ef-400a-b769-c5402dce407b.

[9] Walkenbach, J. (2019). **Excel VBA Programming For Dummies**.

[10] Laporte, G. (1992). **The Vehicle Routing Problem: An overview of exact and approximate algorithms.** European Journal of Operational Research North-Holland. 59: 345-358.

[11] นิรุชา อาภามงคลรัตน์ และ เปรมรัตน์ พูลสวัสดิ์. (2560). **ระบบวางแผนการเดินทางท่องเที่ยวในจังหวัดสมุทรปราการโดยใช้ Dijkstra's algorithm**. ASTC2017: The 5th Academic Science and Technology Conference 2017. หน้า 1476-1483.

[12] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). **Introduction to Algorithms, Second Edition.** The MIT Press.

[13] สำนักงานคณะกรรมการกฤษฎีกา. (2560). **กฎกระทรวง ฉบับที่ ๖ (พ.ศ. ๒๕๒๒) ออกตามความในพระราชบัญญัติจราจรทางบก พ.ศ. ๒๕๒๒.** [Online]. สืบค้นเมื่อวันที่ 26 เมษายน 2565. http://web.krisdika.go.th/data/law/law2/%A803/%A803-2b-2522-006.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ	นางสาว เบญจมาภรณ์ ศรีอัมพร
วันเกิด	4 กรกฎาคม 2539
ที่อยู่ปัจจุบัน	144 หมู่ 6 ตำบลค่ายบกหวาน อำเภอเมือง จังหวัดหนองคาย 43100
ประวัติการศึกษา	(2562) วิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ประยุกต์ เกรดเฉลี่ย 2.98 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (2565) วิทยาศาสตรมหาบัณฑิต สาขาคณิตศาสตร์ประยุกต์ เกรดเฉลี่ย 3.82 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ทุนการศึกษาที่ได้รับ	ทุนอุดหนุนการศึกษา ประเภททุนอุดหนุนยกเว้นค่าธรรมเนียมแบบจ่ายให้ 50% คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปีการศึกษา 2562
ผลงานทางวิชาการ	เบญจมาภรณ์ ศรีอัมพร. บุษยมาส พิมพ์พรรณชาติ และ เทิดขวัญ ช้างเผือก. (2022). **การศึกษาและพัฒนาระบบการจัดการงานสินไหมในกรณีความเสียหายหนักของการประกันภัยรถยนต์.** *Proceedings of the 26th Annual Meeting in Mathematics (AMM 2022).* น. 376-387. สืบค้นจาก https://amm2022.sut.ac.th/Documents/AMM2022_draft.pdf.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้